ORNL/TM--9643

DE86 007080

Engineering Physics and Mathematics Division

VARIANCE REDUCTION METHODS APPLIED TO DEEP-PENETRATION MONTE CARLO PROBLEMS

S. N. Cramer
J. S. Tang*

Manuscript Completed: November 1985
Date of Issue: January 1986

This Work Sponsored by
Defense Nuclear Agency
Under
Interagency Agreement No. 40-65-65

*Computing and Telecommunications Division

Prepared by the
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831
operated by
Martin Marietta Energy Systems, Inc.
for the
U.S. DEPARTMENT OF ENERGY
under Contract No. DE-AC05-840R21400

TABLE OF CONTENTS

PREFACE

This report has been compiled from notes on variance reduction methods presented by the first author at a course entitled "Monte Carlo Methods in Nuclear Reactor Analysis" at the European Community Research Establishment at Ispra, Italy, in October 1984. Other ORNL reports which also consist of lecture notes for this course are:

"Applications Guide to the MORSE Monte Carlo Code," ORNL/TM-9355

"Applications Guide to the RSIC-Distributed Version of the MCNP Code," ORNL/TM-9641.

"Applications Guide to the RSIC-Distributed Version of the KENO-V Code," ORNL/TM-9642.

The lecture notes from the entire course (a total of 17 lectures by seven different authors) will eventually be published by Harwood Academic Publishers, Paris and New York. For further information, contact the course coordinator: Dr. H. Rief, Ispra Establishment, 21020 Ispra (Varese), Italy.

The authors wish to acknowledge the cooperation and assistance of Dr. J. S. Hendricks and his colleagues in the MCNP development group at Los Alamos National Laboratory and Dr. J. C. Nimal and his colleagues in the TRIPOLI group at the Centre d'Etude Nucléaires de Saclay in preparing this report. Appreciation is also extended to Dr. T. J. Hoffman, formerly of Oak Ridge National Laboratory, for his assistance and advice in the general area of Monte Carlo variance reduction methods. A special acknowledgement is due Ms. Katie Ingersoll who typed the drafts and to Ms. Stephanie Raby who typeset the manuscript.

ABSTRACT

A review of standard variance reduction methods for deep-penetration Monte Carlo calculations is presented. Comparisons and contrasts are made with methods for non-penetration and reactor core problems. Difficulties and limitations of the Monte Carlo method for deep-penetration calculations are discussed in terms of transport theory, statistical uncertainty and computing technology. Each aspect of a Monte Carlo code calculation is detailed, including the natural and biased forms of (1) the source description, (2) the transport process, (3) the collision process, and (4) the estimation process. General aspects of cross-section data use and geometry specification are also discussed. Adjoint calculations are examined in the context of both complete calculations and approximate calculations for use as importance functions for forward calculations. The idea of importance and the realization of the importance function are covered in both general and mathematical terms. Various methods of adjoint importance generation and its implementation are covered, including the simultaneous generation of both forward and adjoint fluxes in one calculation. A review of the current literature on mathematical aspects of variance reduction and statistical uncertainty is given. Three widely used Monte Carlo codes — MCNP, MORSE, and TRIPOLI — are compared and contrasted in connection with many of the specific items discussed throughout the presentation.

I. INTRODUCTION

All deep-penetration Monte Carlo calculations require variance reduction methods. Before beginning with a detailed approach to these methods, several general comments concerning deep-penetration calculations by Monte Carlo, the associated variance reduction, and the similarities and differences of these with regard to non-deep-penetration problems will be addressed. The experienced practitioner of Monte Carlo methods will easily find exceptions to any of these generalities, but it is felt that these comments will aid the novice in understanding some of the basic ideas and nomenclature. Also, from a practical point of view, the discussions and developments to follow will be oriented toward use of some specific Monte Carlo computer codes.

"Deep penetration" is a somewhat nebulous term usually associated with radiation shielding problems in contrast to reactor core or non-penetration problems. However, there are no distinct correlations between "penetration, deep penetration, and very deep penetration" and distance of penetration through matter in terms of "a few mean free paths, several mean free paths, many, or many many mean free paths." In Monte Carlo reactor calculations very precise results are usually wanted with statistical uncertainties on the order of a few *tenths* of a percent or less, even for differential quantities. And though the population of simulated particles* is relatively easy to maintain in these calculations due to particle generation from fission, many ingenious variance reduction techniques have been developed in order to achieve the desired results. These reactor core techniques often only reduce the statistical error about the result, which may be too high as often as too low, without producing a large percentage change in the result. In contrast, deep-penetration Monte Carlo has traditionally involved integral quantity calculations where results differing from experimental values by as much as a factor of ten and statistical uncertainties of a few *tens* of percent might be acceptable. Poorly biased† calculations will usually give results lower than the correct results by more than the range of the statistical error. Often this error is deceivingly small. Fortunately, the few situations where results are much larger than the truth also give very large statistical uncertainty. Deep-penetration variance reduction methods must first be concerned with producing results whose statistical error brackets the correct answer, and then as for reactor calculations, proceed to squeeze this error about the true answer until some acceptable convergence is achieved. Sometimes the primary goal is not met, and improper or poorly applied biasing methods only reduce the uncertainty around an erroneous result.

Variance reduction methods in deep penetration are directed toward sustaining the particle population as it moves from the source region to the detector (or results) region. For example, consider a shield of thickness D cm ($\Sigma_T D$ = mean free paths, η) through which

*These words — particle, history, simulation — singly or in combination, all have the same meaning with reference to Monte Carlo methods. Sometimes the particle type — neutron, gamma ray, or photon — is used.

†The application of variance reduction is usually identified by some combination or form of the following words — biasing, sampling, weighting, importance. Although there are distinctions, these words and phrases are often used interchangeably throughout the literature.

it is desired to transport 1000 particle histories at a given energy in a Monte Carlo calculation. Table 1 gives the approximate number of source particles required in an analog (no biasing) calculation. Since 10^5 source particles would be considered a large Monte Carlo calculation, the desired biasing effort is to keep the number of particles approximately constant but reduce the mathematically simulated value (weight) of each particle as it moves through the shield. This reduced weight then gives the correct contribution to the desired results.

Table 1. Required Source Particles for 1000 Escapes

η	Source Particles
5	1.5×10^5
10	2.2×10^7
20	4.9×10^{11}
30	1.1×10^{16}
50	5.2×10^{24}

Deep-penetration calculations can be greatly complicated by the presence of radiation streaming through pipes, ducts, or other paths through a shield. In fact, the classic cases of excessive radiation from reactor shields have been from streaming through areas unaccounted for in the shield design, not through penetration of solid material. Consider the shield and penetration shown in Fig. 1. If it is known that radiation streaming only through the void is significant, then the biasing effort in a Monte Carlo calculation would involve keeping the particle population in close proximity to the void region, rather than directing it straight through the shield. As the distances A, B, and C are changed relative to each other, then combinations of shield penetration and streaming through distance A or C alone might become important. In these situations the variance reductions techniques could become extremely complicated and multiple calculations might be necessary, since the biasing for streaming and penetration would be quite different. In general, Monte Carlo methods cannot be used to determine the radiation streaming paths in a geometrically complex shield, such as can be done with a two-dimensional discrete ordinates flux plot, and some *a priori* knowledge of these paths must be available before beginning a calculation and introducing variance reduction procedures.

Another complication introduced into deep-penetration calculations and biasing procedures is the generation of secondary gamma rays (photons) from neutron interactions in the shield. It is true that there exist in nature nearly pure gamma-ray sources, and gamma only calculations are relatively easy. However, gamma-ray production always results from neutron interactions with matter, even though an experimenter or an analyst may choose to ignore them. In fact, situations may exist where it is the secondary gamma rays produced in the shield, rather than the original neutrons, that create the most serious radiation hazard outside the shield. For these problems, biasing the neutron calculation for the correct gamma-ray production could be more complicated than for neutron penetration alone. These same comments would apply to fission neutron production, if applicable. Other types of calculations requiring special biasing attention are sky-shine problems and a variety of small integral experiments where large angle scattering becomes important.

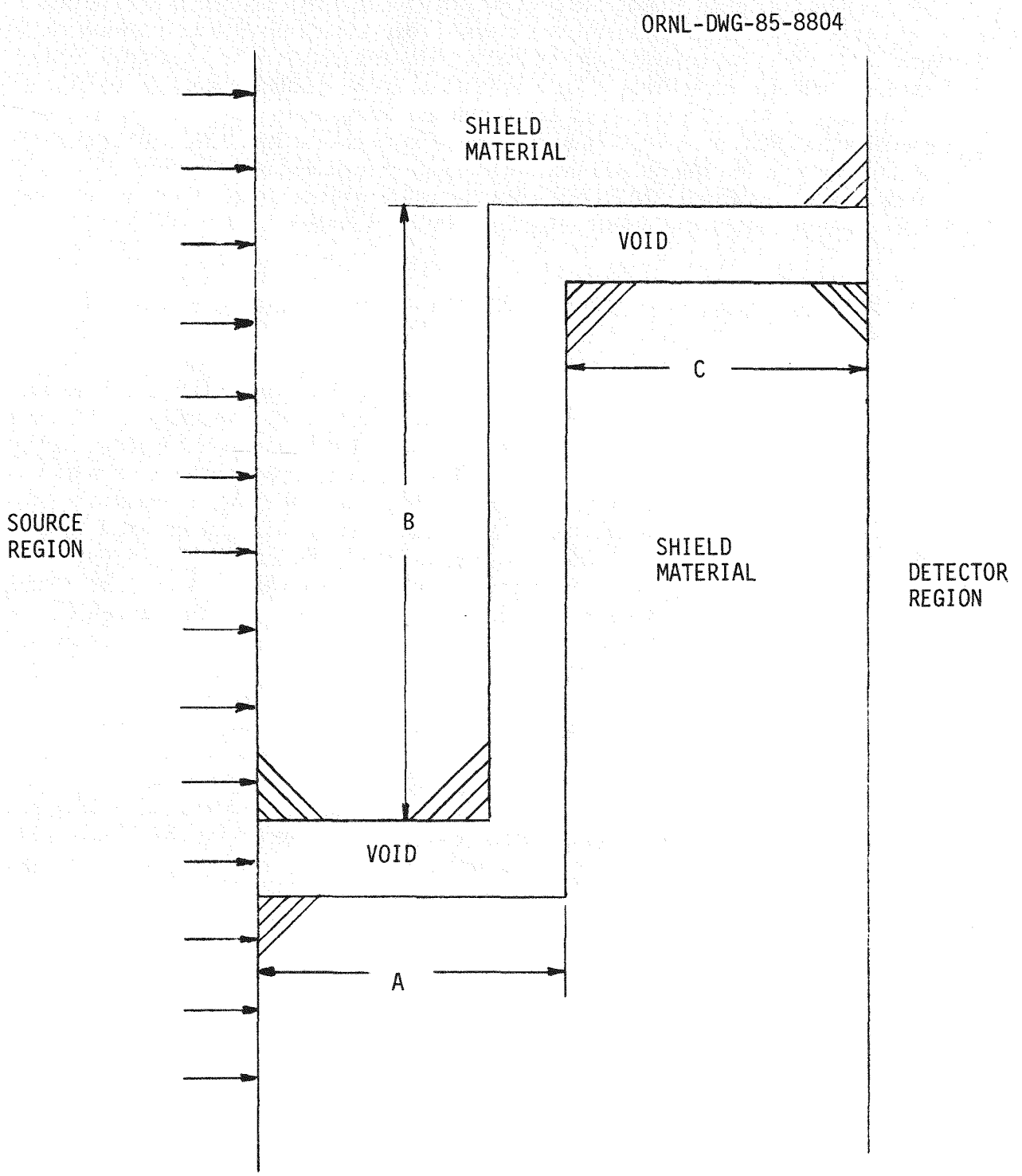

Fig. 1. Radiation shield and streaming path.

The computer codes which have been written to treat these non-reactor core problems are referred to as "general Monte Carlo codes." They are sometimes descendants of codes originally written for criticality and reactor applications and have been extended to treat general fixed source problems. Three such codes to be discussed here are TRIPOLI,[1] MORSE,[2] and MCNP.[3] TRIPOLI is a continuous energy code with which the only user interaction allowed is a rather complicated set of input instructions. MORSE is a multigroup code with less complicated input but requiring some special user written programming for each calculation. MCNP is a continuous energy code with a lengthy set of input instructions permitting, but not requiring, user written routines.

II. BASIC CONCEPTS OF DEEP-PENETRATION VARIANCE REDUCTION

Deep-penetration calculations utilize most of the variance reduction methods for non-penetration problems, such as survival biasing, splitting, and Russian roulette. To these are added any of several methods to be discussed here. The basic problem is shown in Fig. 2: a radiation source on one side of a shield, a detector region on the other side, and enough geometrical complexity to require a Monte Carlo solution. Occasionally, the Monte Carlo method is used primarily for other reasons such as energy detail or time dependence. The calculational effort is directed toward biasing the particles to those regions of phase space (position, energy, direction) which contribute most to the desired results. The position and energy variables are usually given the most attention, i.e., transport particles to the detector position which have energies to which the detector response is sensitive. The direction and time variables generally receive less attention but can be important especially in the source region. Before a calculation is begun, all phase space is divided into regions of importance, each region being assigned a relative value based on its "importance" to the results. This division can be very crude (one region for all phase space) or so complicated as to make the problem setup and running time prohibitive. For the same physical problem (source, geometry, etc.) these regions and importances will in general be different for different desired results in the detector regions. Biasing and particle weight adjustment are done according to the importance region the particle currently occupies.

Consider a calculation involving only splitting and Russian roulette in space and energy. Specification of the importance regions involves setting the splitting and Russian roulette input parameters (lately referred to in the literature as a "weight window" if the weight is a parameter) so that as particle histories move toward the detector and have the desired energies, they are split into more low weight particles. Likewise, as they transport and collide into less important space and energy regions, most are killed by Russian roulette but a few survive as higher weight particles. The weight adjustments preserve the "fair game," producing final unbiased* results at the detector. And since the calculational procedure in the code is the same regardless of the particle weight, the majority of the computation time is spent on "important" particles.

*A "biased" answer is a wrong answer. Correct biasing procedures produce "unbiased" results.

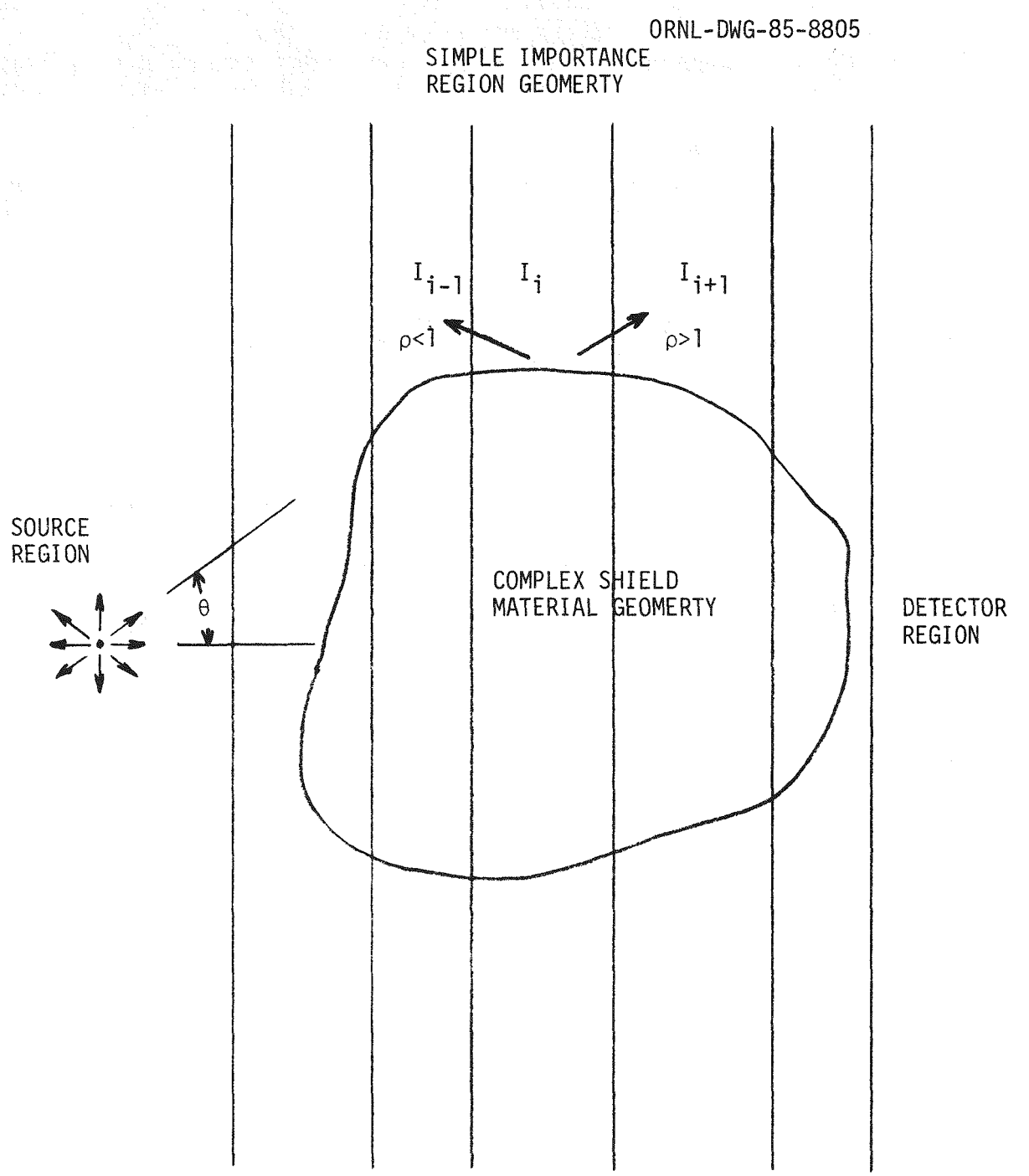

Fig. 2. Deep-penetration biasing geometry.

Care must be taken not to attempt to do this biasing too quickly by having large discontinuities in importance values over adjacent regions of phase space. Too much Russian roulette may result in information important to the answer being lost, and lost information can never be recovered. Too much splitting may result in important information never being acquired, as particles contributing to the results may be descended from only a few source particles. Too much biasing is often worse than too little and gives not only incorrect results and large variances, but can even tax the capabilities of the code and computing machine.

Of primary concern in biasing is not only getting particles to those regions of phase space important to the results, but doing so with a particle population whose dispersion in weight values is not too great. It is the particle weight which ultimately creates the desired results, and if a few high weight particles dominate many low weight particles, the end result can be disastrous. The fewer the number of significant contributions to the answer is, the higher the variance will be; and the time spent tracking and biasing the low weight particles is wasted.

All biasing procedures introduce weight changes — usually weight decreases for particles biased into regions of more importance than for an analog calculation, and weight increases for particles biased into regions of less importance than for the analog case. It is known that some biasing procedures actually cause an increase in the variance, σ^2, in a theoretical sense, over the analog procedure, e.g., Russian roulette. But since it is known from experience that this biasing is necessary to produce reasonable results in a reasonable amount of computing time τ, it is the product $\sigma^2\tau$ (or its inverse, the figure of merit) that is usually regarded as the proper parameter for determining the ultimate efficiency of a biasing technique. The output of a Monte Carlo code, in addition to giving the desired results, will also provide the user with a variety of information which will enable him to determine how useful were his chosen biasing methods in connection with his phase space importance input values. This information is available in tabular form and, for geometrically complex problems, in graphical or plotted form. Proper interpretation of this information is often more difficult than any other user interaction with the code, but it is essential for altering importance input values and options for improved results on any future calculations.

Compared to determining importance input values, all other input and setup effort necessary for a deep-penetration Monte Carlo calculation is straightforward, although these other tasks are often tedious and time consuming for the programmer. There are three principal means for setting importance values — (1) empirically, (2) using adjoint calculation results, and (3) code generated adaptive, or learning, techniques. Empirical means are by far the most used and may initially involve some guess work. But for any reasonable chance of success this must be "well-educated guessing," which can be broken into several categories — intuition; physical insight; trial and error; and, probably most important, experience with particular biasing techniques, problem types, and codes. All these empirical categories are usually used in some combination with several short calculations and subsequent biasing adjustments before a long calculation is made.

With the adjoint information, the idea of "importance" takes on a more precise mathematical meaning, i.e., the adjoint flux is the importance in phase space.[4] Since the adjoint calculation is often more difficult than the forward calculation for which importance values are wanted, this approach leads to the use of approximate adjoint solutions. These could be from (1) analytical approximations to the adjoint equation for very simple cases, (2) approximate numerical methods with exact geometry (diffusion theory), or (3) exact numerical methods with approximate geometry (discrete ordinates). It is the last item, discrete ordinate adjoints, which have had some utilization in deep-penetration Monte Carlo codes, particularly multigroup Monte Carlo codes. However, this adjoint information is often voluminous and difficult to interpret, and even though it may be a good approximation to the exact adjoint, it is usually cumbersome to implement into the Monte Carlo code input structure without some automated procedure.

Techniques for code-generated learning methods also include some initial guesswork and automatic implementation of the generated biasing parameters. Several specific applications will be discussed later in Sections IV.B and V. The parameters are created from some algorithm during the calculation just as for any other quantity. The algorithm itself may be empirical or mathematical in nature or related to a specific definition of the adjoint. These parameters may be updated and used in the same calculation or created for use in a following calculation. Comments on the variance characteristics of learning methods are included in Section VI.

III. IMPLEMENTATION OF VARIANCE REDUCTION TECHNIQUES IN DEEP-PENETRATION CALCULATIONS

As was stated previously, all deep-penetration Monte Carlo calculations require biasing, or variance reduction, techniques. Unlike reactor calculations where the calculated result may converge toward the correct answer from above or below through variance reduction methods, poorly biased deep-penetration answers are almost always low due to an inadequate sampling of (particle occurrence in) important regions of phase space. Here a certain amount of biasing is necessary before the estimated error of the result encompasses the correct answer. Thus, the various aspects of deep-penetration calculations will be examined both in terms of getting the correct answer (accuracy) and in reducing the variance of the answer (precision). This examination will be more or less in the order one would set up and run a calculation and not necessarily in the order of most important variance reduction: (1) cross-section data, (2) geometry description, (3) source description, (4) transport process, (5) collision process, and (6) estimation of results.

III.A. Cross-Section Data

Because deep-penetration calculations usually involve particles at higher energies, the necessary cross-section treatment is often less complicated than that for reactor calculations if resonance regions can be omitted or simplified. A notable exception to this situation would be when gamma-ray production from low energy neutron capture is important

or in the unusual case when fissionable material is present. Of primary importance in penetration calculations is adequate representation of any (usually broad) anti-resonances, or windows, in the neutron cross sections of elements such as iron and oxygen (see Fig. 3). Streaming through these windows (at these energies) can account for a large percentage of shield penetration, and in this respect care must be taken even when using continuous energy codes. Use of true point data precludes any problem here, but many continuous energy codes provide, in the standard code package, shielding libraries of "pseudo-point" cross sections (also called "discrete" data) averaged over a few hundred narrow energy intervals. These are not true multigroup data, but are point data constant over a given interval as illustrated for iron data in Fig. 4. Use of these "discrete" data can substantially reduce computer memory requirements from that for point data, and their use has been shown to be adequate for integral results[5] but may require adjustments for differential results.[6] The multigroup MORSE code also has a few standard shielding libraries, and problems peculiar to multigroup cross sections are discussed in reference 2. Data sets designed and weighted for specific applications should not be used indiscriminately for other applications. A particular problem arises in multigroup deep-penetration calculations if only one weighting scheme is used in the shield. Just as for discrete ordinates applications, the shield should be divided into several regions, even if it is entirely one material,

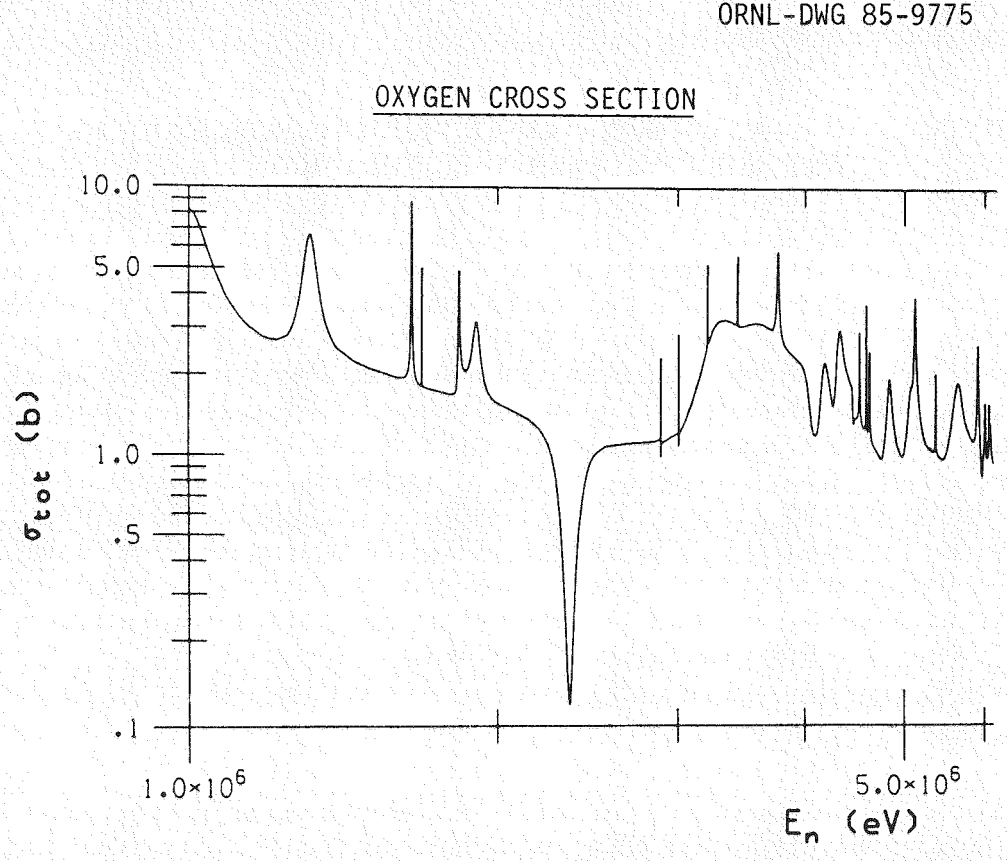

Fig. 3. Oxygen cross section.

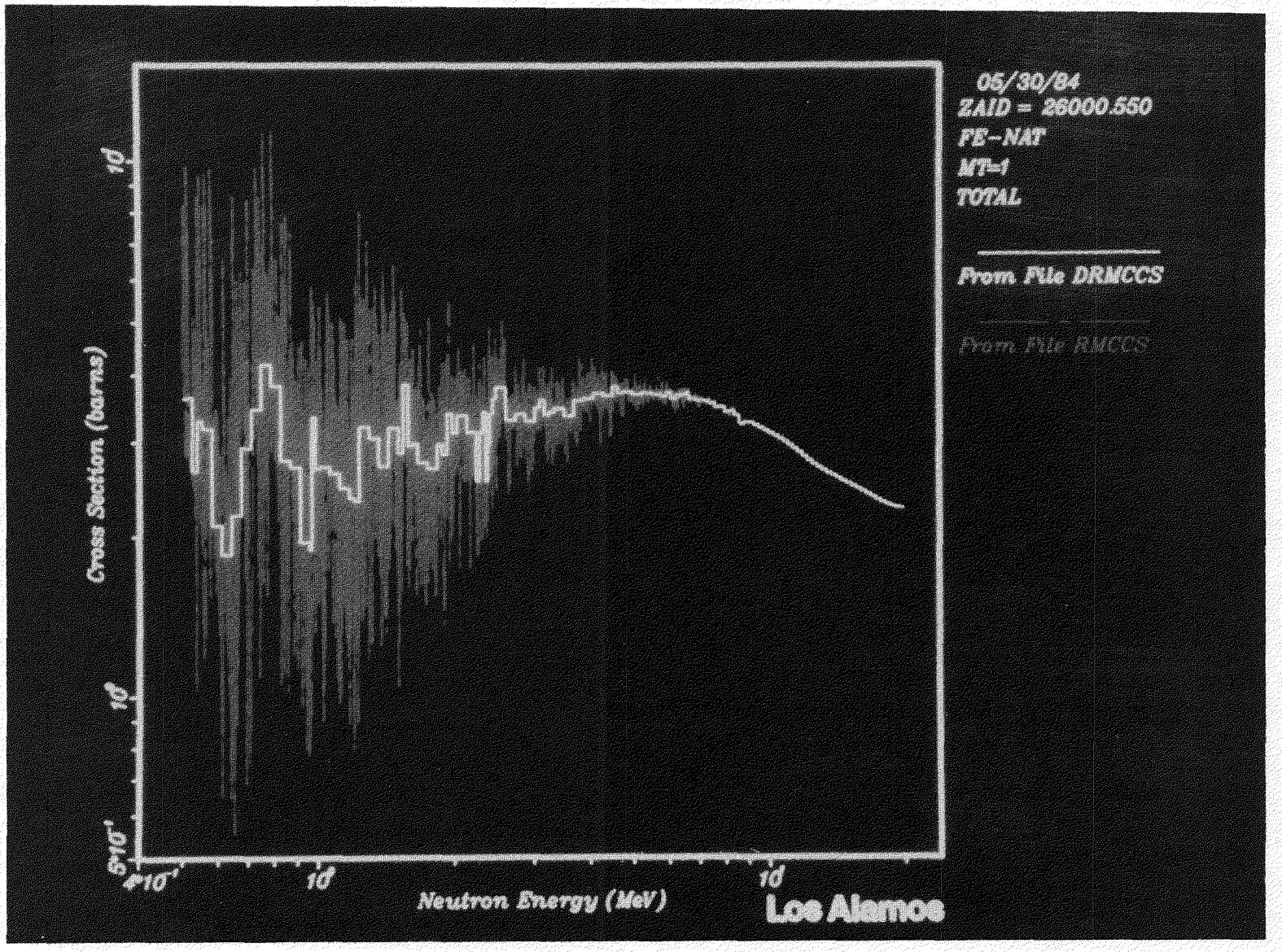

Fig. 4. Point and pseudo-point (discrete) data for the total cross section of iron.

with appropriate cross-section weighting in each region as indicated by the shape of the flux spectrum determined from the fine group collapsing procedure. Special multigroup and pseudo-point libraries provided with code packages have weighting, resonance treatment, and group boundaries selected from sensitivity and other studies, which are adequate for most deep-penetration calculations. If this is not the case, however, the programmer must turn to the original cross-section processing codes, all of which use ENDF/B, and possibly other data, for a better cross-section description. Possible codes are NJOY[7] for MCNP, THEMES[8] for TRIPOLI, or AMPX[9] for MORSE. These are large, complicated codes and the study of their theory, structure, input, and output is as difficult as for the Monte Carlo codes. However, any such cross-section processing codes might also provide data for the deterministic transport codes at a given installation, and expertise in their use is usually available.

As with the items to follow, one should select cross-section data with regard to its importance to the desired results. Any simplification such as upper or lower energy cutoffs will greatly improve the efficiency of the Monte Carlo calculation. Cross-section data would be considered to affect the accuracy rather than the precision of a calculation. However, cross-section sensitivity and perturbation methods applied to a calculation would be more related to precision.[10]

III.B. Geometry Description

The general comments given for the cross-section data also apply here. The geometry for a deep-penetration calculation is usually less complicated than for reactor studies, and one should concentrate on the detail where it is important to the desired results. Even for relatively simple geometries, a large fraction of time of any Monte Carlo calculation is spent in the geometry routines, and certain variance reducing techniques such as truncated exponential biasing and next-event estimation require much geometry tracking. As in the previous item, the geometry description would affect the accuracy of the result, but sensitivity and perturbation analysis of the composition and size of certain portions of the geometry would affect the precision. Cross-section data is usually accepted "as is" with little user checking at calculation time. However, the geometry data should be carefully checked prior to any calculation using any of the various graphical and plotting diagnostic aids available with the geometry packages in the codes. An example of array geometry capability currently available in general Monte Carlo codes is given in Section III.F.8.

III.C. Source Description

Deep-penetration calculation source descriptions can affect both the accuracy and precision of the result. All important (to the result) portions of source phase space must be adequately sampled for accuracy, and source biasing is one of the most effective and easily applied variance reducing techniques available.

III.C.1. Biased Source

The basic expression for the biased source distribution is

$$\hat{S}(x) = \frac{S(x)I(x)}{\int_{x_s} S(x)I(x)dx} , \tag{1}$$

where

$S(x)$ is the natural source distribution,
$I(x)$ is the importance function, and
x represents any independent phase space variables – space, energy, angle, or time – and the integral is over all the source phase space x_s.

If $I(x)$ is constant over the source for any variable, then that variable will be unbiased.

The weight correction to the unbiased source weight (usually unity) is

$$WT = \frac{S(x)}{\hat{S}(x)} = \frac{\int_{x_s} S(x)I(x)dx}{I(x)} . \tag{2}$$

This multiplicative weight adjustment is made after each biased variable selection x for each source particle. It is seen that the weight correction is proportional to the inverse of the importance function. This is true of any biasing procedure throughout the Monte Carlo process. In the absence of any importance information, it is usually the biased distribution, in this case $\hat{S}(x)$, that is determined empirically rather than $I(x)$. The inverse weight correction is then indicative of the assumed relative $I(x)$.

For selection of a specific variable, Eq. (1) must be a proper marginal and/or conditional distribution function, i.e., to select an energy from $S(\bar{r}_o,E,\bar{\Omega})$ after an $\bar{r}_o$ has been selected, one has

$$\hat{S}(E)|r_o = \frac{\int_{\Omega} S(\bar{r}_o,E,\bar{\Omega})I(\bar{r}_o,E,\bar{\Omega})d\bar{\Omega}}{\int_E \int_{\Omega} S(\bar{r}_o,E,\bar{\Omega})I(\bar{r}_o,E,\bar{\Omega})d\bar{\Omega}dE} . \tag{3}$$

If there is any inter-variable function dependence such as $E = f(\bar{\Omega})$, then $\bar{\Omega}$ is randomly selected from $S(x)$ and E is set as $f(\bar{\Omega})$. In any of the general computer codes, the Cartesian coordinates x, y, z (three random selections) and the direction cosines with respect to the three axes α, β, γ (random selections from polar and azimuthal distributions) must be determined for each source particle. If $S(x)$ is normalized to unity, i.e., $S_T = \int_{x_s} S(x)dx = 1$, then the results will be given relative to one starting source particle.

Energy and angle are the two most commonly biased source variables. Even when importance information is not available, the various empirical methods can be very helpful. Setting $I(E)$ equal to the response function $R(E)$ is sometimes helpful as an initial guess. It should be pointed out that excluding portions of phase space can be thought of as biasing with zero importance. For example, in Fig. 2 any backward angles should be excluded if those particles could never reach the detector, or any source energies below the detector response threshold should never be sampled. However, at what angle $\theta < 90°$ or what energies above the detector response cutoff to terminate the source selection process can be difficult to determine, but this can greatly affect the problem calculation time. For a shield penetration as in Fig. 1, only a very small source angular interval at the duct entrance would be important to the result even if the source is isotropic in direction.

For time dependent source problems, the time variable can be biased using Eqs. (1) and (2). Although the time variable can be used as a parameter for choosing a biasing scheme later in the calculation, time becomes a dependent, rather than independent variable after the source selection, and there is no provision for time biasing in the transport and collision process.

III.C.2. Segmented Sources

For a source system that is irregular or presents some difficulty in sampling, it is sometimes necessary to divide the source into j segments, Δx_j, and sample separately from each segment. The Δx_j may represent a portion of any phase space variable. The probability of selecting a source particle from segment i is

$$P_i = \frac{\int_{\Delta x_i} S(x)dx}{\sum_j \int_{\Delta x_j} S(x)dx} . \tag{4}$$

This selection may also be biased as

$$\hat{P}_i = \frac{I_i \; P_i}{\sum_j I_j \; P_j} , \tag{5}$$

where I_j is the importance for segment j,

$$I_j = \frac{\int_{\Delta x_j} I(x)dx}{\Delta x_j} . \tag{6}$$

After i is determined using the cumulative distribution of Eq. (5), Eq. (1) is used to select x within Δx_i. The weight correction in Eq. (2) is now multiplied by an additional factor WT_i,

$$WT_i = \frac{\sum_j P_j \; I_j}{I_i} \quad . \tag{7}$$

In actual practice the $\hat{P}_i$ (or I_i) are often determined empirically, and the $I(x)$ within any Δx_i might be set to unity if x is divided into many segments. In the segmentation of more than one phase space variable, Eq. (5) is applied independently for each variable as with Eq. (1), and Eq. (7) is then applied multiplicatively as with Eq. (2).

The biasing described here can cause large variations in the weight of the source particles, particularly if several variables are biased. As the particles propagate through phase space toward the desired results regions, care must be taken not to lose too much source information due to Russian roulette and splitting. This problem is exaggerated when there is also biasing in the transport and collision processes. All the code input parameters for particle population control must reflect the source biasing and weight adjustments. For example, if the source energy selection is biased to produce many important (low weight) high energy particles and a few unimportant (high weight) low energy particles, and then somewhere in the calculation the Russian roulette and splitting parameters are applied uniformly in energy, most of the low weight particles will be killed. A few of these survive with higher weights, and the low energy particles are split into many lower weight particles. The net effect of this situation is to reproduce the unbiased energy distribution as if no source biasing had occurred.

One method of easing this problem of weight and population control is to transfer the source weight correction to the estimation process. That is, do not adjust the source weight, Eqs. (2) and (7), in the source routine, but adjust the particle weight in the estimation routine. This procedure should reduce the burden of setting Russian roulette and splitting input parameters, and more source information will contribute to the results. However, the calculation time will usually increase as well as the variance, since large weight variations in the estimation process can produce large variances. Also some code modification would be necessary to carry this extra weight value throughout the calculation.

A better method of dealing with difficult source distributions is to divide the problem into i separate computer calculations, each with a source selected from Δx_i, Eq. (1), and giving a detector result and variance $\lambda_{\Delta x_i}(x)$. Each of these calculations is independent of the others in terms of source selection, biasing, weighting, etc. The complete result $\lambda(x)$ is, using Eq. (4),

$$\lambda(x) = \sum_i P_i \; \lambda_{\Delta x_i}(x) \quad . \tag{8}$$

In applying Eq. (8), care must be taken to cover all portions of source phase space once and only once.

An extension of the above method can be applied when a calculation is to be repeated many times with only a change in the source distribution, and it can be assumed that all source distributions are constant within a given Δx_i. It is also assumed that adjoint calculation is not feasible. As before, a separate calculation is made for each Δx_i, but now the source is chosen uniformly in the interval. The results of each calculation $\lambda_{\Delta x_i}(x)$ are retained, and Eq. (8) can be evaluated for any discretized source from Eq. (4). An example of this procedure would be for a problem with many possible source spectra, each of which can be discretized into the same ΔE_i intervals and across which each source can be assumed to be uniform. A λ is computed for each ΔE, and Eq. (8) is then evaluated for each spectrum.

III.C.3. Source Adjoint

The importance function $I(x)$ in Eq. (1) for the source is a subset of the adjoint for the entire problem phase space, which is usually difficult to determine, interpret, and apply in a code. However, for a situation as illustrated in Fig. 2, where the source and detector regions are relatively simple, the adjoint flux $\phi^*(x)$ from one- or two-dimensional discrete ordinates model calculations can be a good approximation to $I(x)$. For source energy or angle biasing, $\phi^*(x)$ can be applied in most codes without too much difficulty.

As an example of a learning technique, estimation of the true source adjoint can be made simultaneously with the forward Monte Carlo calculation and used as an improved $I(x)$ in subsequent calculations. The total response is given by either of the following integrals:

$$\lambda = \int_{x_R} \phi(x)R(x)dx = \int_{x_S} S(x)\phi^*(x)dx \quad , \tag{9}$$

where $\phi(x)$ and $R(x)$ are the forward flux and response function, and x_R and x_S represent the detector and source phase space, respectively. Rewriting Eq. (9) for a segmented source ($x_S = \sum_i \Delta x_i$),

$$\lambda = \sum_i \lambda_i = \sum_i \int_{\Delta x_i} S(x)\phi^*(x)dx \quad , \tag{10}$$

where λ_i is that portion of the response made from all particles which originated in Δx_i:

$$\lambda_i = \int_{\Delta x_i} S(x)\phi^*(x)dx \quad . \tag{11}$$

The average adjoint flux in the source segment is

$$\phi^*_{\Delta x_i} = \frac{\int_{\Delta x_i} S(x)\phi^*(x)dx}{\int_{\Delta x_i} S(x)dx} = \frac{\lambda_i}{\int_{\Delta x_i} S(x)dx} \quad . \tag{12}$$

The λ_i is evaluated by writing the forward flux at x as the sum of its parts coming from each source segment Δx_i,

$$\phi(x) = \sum_i \phi_{\Delta x_i}(x) \quad , \tag{13}$$

and substituting this summation into the left side of Eq. (9). Then for each *source* segment Δx_i,

$$\lambda_i = \int_{x_R} \phi_{\Delta x_i}(x)\ R(x)dx \quad . \tag{14}$$

In evaluating Eq. (14) in a Monte Carlo code, the source parameters of each particle history must be retained until the estimation process so that i can be determined. The phase space variable x in Eq. (14) is in x_R, but the i for λ_i is from the source segment Δx_i. This feature is not standard in most codes, and some nominal programming would be necessary to evaluate Eqs. (12) and (14). This source adjoint method is part of a more general procedure given in Section IV.B. An example of source energy biasing where the implicit adjoint function is improved by iteration during the forward calculation is mentioned briefly in Section V.

III.C.4. Other Sources

The source descriptions in this section can be implemented in any Monte Carlo general code with varying degrees of difficulty. TRIPOLI requires that all source descriptions be made via input instruction. In MORSE only the simplest sources (point, isotropic, monodirectional) can be input — all others must be programmed into a source routine. MCNP has a few standard input sources which are adequate for most problems and also provides for an optional user written routine. A more general MCNP source option is under development. As with the geometry description, it is important to thoroughly check the source description independent of the entire calculation. It is best to do this checking with a "source and estimation only" calculation in the code to see if the biasing and weight corrections reproduce (within statistical uncertainty) the natural source.

A technique useful in deep-penetration calculations is to create a source distribution from a previous calculation and couple it with the Monte Carlo, bypassing any source input or programming in the standard code. This procedure can save much computation time if the physical source is in a region which can be described in one or two dimensions. A few coupling schemes are mentioned in reference 11. The DOMINO[12] code couples the discrete ordinates code DOT to MORSE. The TRIPOLI and MCNP codes have similar methods under development. Biasing techniques are not generally standard features in coupling codes, and some user interaction would be required to include them. The general procedure is to produce a source (current) from the coupling code output, this usually being from a boundary flux quantity. Care must be taken in choosing the coupling boundary, the boundary conditions of the two calculations, and any overlap of the two geometry descriptions so that any boundary re-crossing is treated correctly.[13]

There has also been some work in Monte Carlo-Monte Carlo coupling.[14] A simple example of Monte Carlo-Monte Carlo coupling could be employed if the penetration in Fig. 1 were only a very small part of a much larger system extending on the left side of the figure. It could be assumed that the penetration has no effect on a Monte Carlo calculation of the larger system (see Fig. 5). A point estimator could be used to determine a directional dependent (with respect to the duct centerline) source at the duct entrance, and this source could be applied uniformly over the entrance for a second calculation through the penetration.

III.D. Transport Process

In the study of deep penetration of radiation by Monte Carlo, it is the transport process and variance reduction techniques associated with it that have received the most attention. As illustrated in Table 1, it is impractical from a computing viewpoint to perform analog deep-penetration calculations. As outlined in the introduction, the basic idea is to keep the biased particle population approximately constant as it moves through the important regions of the system from the source to the detector (results) regions. In doing this, the particle weight is reduced in value proportional to the particle population loss in an analog case such that the summation of the particle weight is conserved.

These concepts are easy to visualize in simple, one-dimensional situations. However, for realistic three-dimensional applications, they can become extremely difficult, and there is no general recipe for success. Again, any knowledge of the importance function can be very useful. As particles are transported into regions of low importance, their number should be reduced (with appropriate weight increase) so that computer time is not wasted on unimportant particles. If these particles subsequently return to higher importance regions, their number is increased with appropriate weight reduction. There are two commonly used techniques for performing this transport biasing — splitting with Russian roulette and the exponential transform.

ORNL-DWG 85-9769

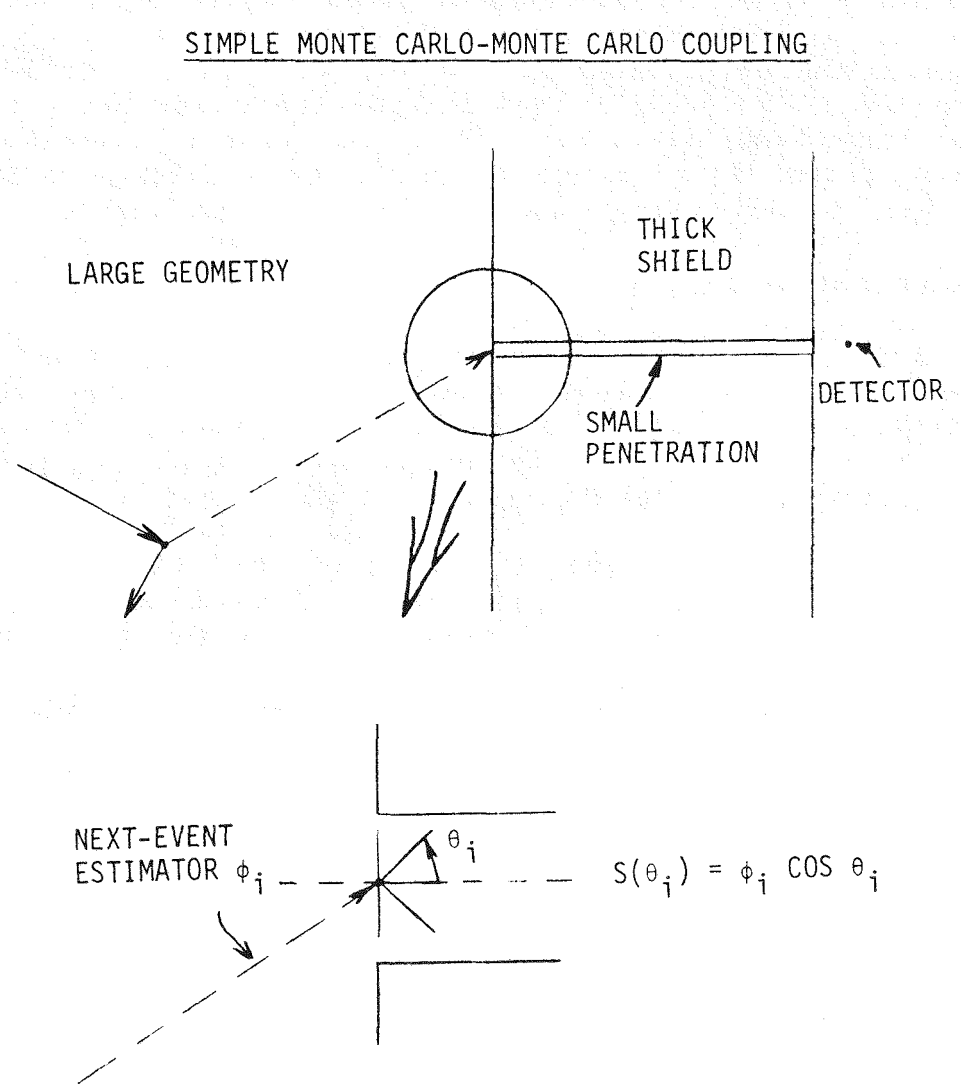

Fig. 5. Simple Monte Carlo-Monte Carlo coupling.

III.D.1. Boundary Splitting and Russian Roulette

The procedure for splitting and Russian roulette is to divide the system into volumes, each with an importance I_i. When a particle of weight WT crosses from volume i into volume $i + 1$, the ratio of importance $\rho = I_{i+1}/I_i$ determines which game is played. If $\rho < 1$, Russian roulette is implemented, and the particle survives with probability ρ with an increased weight WT/ρ. For $\rho > 1$, ρ identical particles are created each with weight of WT/ρ. (In actual practice the integer part of ρ is used to split particles and Russian roulette is played with the fractional part.) These I_i must be made relative to that for the source region since this is where the original WT is set (see Section IV.D).

For population stability in a simple case like Fig. 2, the I_i and the region dimensions could be defined such that $\rho = e^{\pm N_\eta}$, where N_η is the number of mean free paths in the region where the particle is exiting and the sign depends on the particle direction. In practice there should be enough volumes such that the particle population does not drop by more than a factor of 2 to 5 between adjustments.[5] For larger drops, too many independent particles from the original source population are lost, and the adjusted population may become highly correlated due to multiple splitting of a few particles.

This splitting and Russian roulette at region boundaries may be extended to make the procedure dependent on other variables, such as energy. This would ensure population stability in the important energy ranges of the detector response. However, any increase in the differentiality of the method greatly increases the burden of determining the I_i as a function of more than one variable. The code learning technique described in Section IV.B has been used to automatically set the I_i for splitting and Russian roulette.[15] In addition to its use in deep penetration, splitting and Russian roulette are used in reactor calculations to control the population of particles entering and returning from the reflector. Splitting and Russian roulette associated with the collision process is discussed in Section III.

III.D.2. The Exponential Transform

The other method employed in Monte Carlo codes to maintain an adequate particle population at large distances from the source is the exponential transform. (This method is also known as "path length stretching," which is a misnomer since paths are also sometimes shrunk.) The standard (time independent for convenience only) transport equation is, with the usual definition of terms,

$$\bar{\Omega}\cdot\bar{\nabla}\phi(\bar{r},E,\bar{\Omega}) + \Sigma_T(\bar{r},E)\phi(\bar{r},E,\bar{\Omega}) = S(\bar{r},E,\bar{\Omega})$$

$$+ \int_{E'}\int_{\bar{\Omega}'} \Sigma_S(\bar{r},E' \rightarrow E,\bar{\Omega}' \rightarrow \bar{\Omega})\phi(\bar{r},E',\bar{\Omega}')dE'd\bar{\Omega}' \quad . \qquad (15)$$

If it is assumed that the flux decreases exponentially with distance, then the following transform may be introduced into Eq. (15) for some arbitrary function $P(\bar{r},E,\bar{\Omega})$ such that the transformed flux ψ remains more nearly constant:

$$\psi(\bar{r},E,\bar{\Omega}) = \phi(\bar{r},E,\bar{\Omega})\ e^{P(\bar{r},E,\bar{\Omega})} \quad . \tag{16}$$

The first term on the left side of Eq. (15) is

$$\bar{\Omega}\cdot\bar{\nabla}\phi(\bar{r},E,\bar{\Omega}) = e^{-P(\bar{r},E,\bar{\Omega})}\left[\ \bar{\Omega}\cdot\bar{\nabla}\psi(\bar{r},E,\bar{\Omega}) - \psi(\bar{r},E,\bar{\Omega})\ \bar{\Omega}\cdot\bar{\nabla}P(\bar{r},E,\bar{\Omega})\ \right] \quad . \tag{17}$$

The entire transformed equation is written as

$$\begin{aligned}
&\bar{\Omega}\cdot\bar{\nabla}\psi(\bar{r},E,\bar{\Omega}) + \left[\ \Sigma_T(\bar{r},E) - \bar{\Omega}\cdot\bar{\nabla}P(\bar{r},E,\bar{\Omega})\ \right]\psi(\bar{r},E,\bar{\Omega}) \\
&= S(\bar{r},E,\bar{\Omega})\ e^{P(\bar{r},E,\bar{\Omega})} + \int_{E'}\int_{\bar{\Omega}'}\Sigma_S(\bar{r},E'\rightarrow E,\bar{\Omega}'\rightarrow\bar{\Omega}) \\
&\times\ e^{[P(\bar{r},E,\bar{\Omega})-P(\bar{r},E',\bar{\Omega}')]}\psi(\bar{r},E',\bar{\Omega}')dE'd\bar{\Omega}' \quad .
\end{aligned} \tag{18}$$

In theory, the solution of Eq. (18) and the transformation in Eq. (16) will give the solution to Eq. (15). It is seen that the total cross section for transport in Eq. (18) can be defined as

$$\Sigma^*(\bar{r},E,\bar{\Omega}) = \Sigma_T(\bar{r},E) - \bar{\Omega}\cdot\bar{\nabla}P(\bar{r},E,\bar{\Omega}) \quad . \tag{19}$$

The form of Eq. (19) indicates that the transformed cross section Σ^* can be larger or smaller than Σ_T, depending on the functional form of P. Thus, the length of the flight paths selected in the solution of Eq. (18) will be altered from that using Σ_T (i.e., stretched or shrunk).

A form of $P(\bar{r},E,\bar{\Omega})$ often used is

$$P(\bar{r},E,\bar{\Omega}) = p\Sigma_T(\bar{r},E)V \quad , \tag{20}$$

where $V = -|\bar{r}-\bar{r}_o|$ specifies a preferential direction toward $\bar{r}_o$, and p is a constant independent of $\bar{\Omega}$ but possibly dependent on both $\bar{r}$ and E.

Now the transformed cross section has the form

$$\Sigma^*(\bar{r},E,\bar{\Omega}) = \Sigma_T(\bar{r},E)[1-p\mu] \quad , \tag{21}$$

where μ is the cosine of the angle between the particle direction $\bar{\Omega}$ and the preferred direction. If this preferential direction is always toward the same fixed point, e.g., a point detector, regardless of the particle position or direction, then

$$\mu = \bar{\Omega}\cdot\bar{\nabla}V = \frac{\Omega_x(X_D-X) + \Omega_y(Y_D-Y) + \Omega_z(Z_D-Z)}{\sqrt{(X_D-X)^2 + (Y_D-Y)^2 + (Z_D-Z)^2}} , \quad (22)$$

where

X,Y,Z	are the particle coordinates at $\bar{r}$ after the last collision,
$\Omega_x,\Omega_y,\Omega_z$	are the particle direction cosines after collision ($\Omega_x^2+\Omega_y^2+\Omega_z^2 = 1$), and
X_D,Y_D,Z_D	are the coordinates of the position $\bar{r}_0$ toward which it is desired to increase the path lengths.

The position $\bar{r}_o$ could also be made a variable so that the preferred direction changes as the particles traverse a system such as the bent duct in Fig. 1. If the sign in the exponent of Eq. (16) had been negative, then this procedure would be reversed. Paths would be shrunk in the direction toward X_D,Y_D,Z_D and stretched in opposite directions.

In actual practice Eq. (18) is not solved; rather, Eq. (15) is solved using a biased transport kernel, with Σ^* as the transport cross section, to determine the next collision site at $\bar{r}+\bar{\Omega}R$:

$$T^*(\bar{r},E,\bar{\Omega}) = \Sigma^*(\bar{r},E,\bar{\Omega})e^{-\int_o^R \Sigma^*(\bar{r}',E,\bar{\Omega})dR'} . \quad (23)$$

In the usual practice for Monte Carlo codes, the variable is changed to the number of mean free paths, $\eta' = \Sigma^*(\bar{r},E,\bar{\Omega})R = \Sigma_T(\bar{r},E)(1-p\mu)R$, where R is the distance chosen as if the medium of the current collision site extended indefinitely; i.e., $\Sigma^*(\bar{r},E,\bar{\Omega})$ is constant and the normalization of Eq. (23) is unity. This is the usual convention since the codes work in mean free paths instead of cross sections and distance. Then, in compacting notation, one has

$$T^*(\eta') = T^*(R)\frac{dR}{d\eta'} , \quad (24)$$

$$T^*(\eta') = (1-p\mu)\, e^{-(1-p\mu)\eta}\, \frac{d\eta}{d\eta'} = e^{-\eta'} , \quad (25)$$

where η is the unbiased variable and $\eta' = \eta(1 - p\mu)$.

It is seen that the distribution for η' in Eq. (25) is mathematically identical to that for the unbiased kernel, $e^{-\eta}$, and the selection procedure for η' is the same as for η. However, the distance R from the unbiased kernel is $\eta/\Sigma_T(\bar{r},E)$, whereas for the biased kernel

$$R = \frac{\eta}{\Sigma^*(\bar{r},E,\bar{\Omega})} = \frac{\eta}{\Sigma_T(\bar{r},E)(1-p\mu)} \quad . \tag{26}$$

Any boundary crossings and changes in $\Sigma_T(\bar{r},E)$ are handled just as for the unbiased case.

For a selected collision site the weight correction WT due to the biased selection is T/T^*:

$$WT = \frac{\Sigma_T(\bar{r},E)e^{-\Sigma_T(r,E)R}}{\Sigma^*(\bar{r},E,\bar{\Omega})e^{-\Sigma^*(\bar{r},E,\bar{\Omega})R}} = \frac{e^{-p\mu\eta}}{(1-p\mu)} \quad . \tag{27}$$

If the transport is interrupted at an importance region boundary before a collision occurs, the correction is $WT = e^{-p\mu\eta}$, where η is measured to the boundary, and a new η is chosen with the parameter p applicable to the new region.

It is seen that, since $-1 \leqslant \mu \leqslant 1$, p must be chosen such that $|p| < 1$ so that no numerical difficulties occur in the denominator of Eqs. (26) and (27). The parameter p (a code input value) reflects the amount of biasing to be done. Small values of p alter Eqs. (26) and (27) little from the unbiased and unity weight analog case and have the effect of a $\mu \approx 0$ (90° scattering angle with respect to the preferential direction) for any p. For larger values of p the forward peaked distances can become very large and the backward distances shrunk by up to a factor of two. When the chosen η is sufficiently large and $p\mu$ is not close to unity, the weight corrections may become very small in the forward direction and very large in the backward direction. However, for small η, Eq. (27) exhibits the somewhat curious effect of $WT < 1$ for $\mu < 0$ (shrinking) and $WT > 1$ for $\mu > 0$ (stretching). If it is assumed that the biased distribution is of the standard importance function form

$$f'(\eta) = \frac{f(\eta)I(\eta)}{\int f(\eta)I(\eta)d\eta} = \frac{f(\eta)I(\eta)}{N_\eta} \quad , \tag{28}$$

where $f(\eta) = e^{-\eta}$ and the normalization integral N_η is unity, then

$$I(\eta) = (1-p\mu)\ e^{p\mu\eta} \quad . \tag{29}$$

In Section III.E.1 the possible elimination of the $(1 - p\mu)$ term in Eqs. (27) and (29) is discussed.

III.D.3. The Exponential Transform — Truncated Form

In this presentation of deep-penetration methods, the relatively simple case such as in Fig. 2 of a source, a shield to be penetrated (with possible geometry complications), and a detector region on the other side has been studied. The primary objective has been to

completely traverse the shield. However, it is often desired to study some effect inside the shield, perhaps close to the exit side. The objective here would be to bias the particles to the region of interest and then to have collisions without much leakage. One could achieve this effect with judicious application of cell importance ratios I_{i+1}/I_i for boundary splitting and Russian roulette. One can also achieve this effect by applying the exponential transform with a non-leakage restraint. The transformed kernel in Eq. (23) is now normalized from the current collision site to R, the distance along the particle direction to the edge of the shield, instead of to infinity. Then Eq. (23) has the form

$$T^*(R) = \frac{\Sigma^*(\bar{r},E,\bar{\Omega})\ e^{-\int_o^R \Sigma^*(\bar{r}',E,\bar{\Omega})dR'}}{\int_o^{\bar{R}} \Sigma^*(\bar{r}',E,\bar{\Omega})\ e^{-\int_o^{R'} \Sigma^*(\bar{r}'',E,\bar{\Omega})dR''}\ dR'} , \tag{30}$$

and, following the convention for Eqs. (24) and (25),

$$T^*(\eta') = \frac{(1-p\mu)e^{-(1-p\mu)\eta}}{1-e^{-(1-p\mu)\eta_R}} \left(\frac{d\eta}{d\eta'} \right) = \frac{e^{-\eta'}}{1-e^{-(1-p\mu)\eta_R}} , \tag{31}$$

where η_R is the number of mean free paths to escape and must be determined in general by a geometry search prior to the application of Eq. (31).

The simplistic realization of the flight distance R, using a random number $0 \leqslant RN \leqslant 1$, is

$$R = \frac{-\ell n \left[1-RN(1-e^{-(1-p\mu)\eta_R}) \right]}{\Sigma_T(\bar{r},E)(1-p\mu)} . \tag{32}$$

The collision weight correction is

$$WT = \frac{e^{-p\mu\eta} \left(1-e^{-(1-p\mu)\eta_R} \right)}{(1-p\mu)} . \tag{33}$$

If the transport is interrupted at an importance region boundary, Eq. (33) is modified by a factor $(1-p\mu)/(1-e^{-(1-p\mu)(\eta_R-\eta)})$, where η is measured to the boundary, and then a new η is chosen with a new p and η_R.

It can be seen that these equations reduce to the untransformed non-leakage form when $p = 0$, to the exponential transform when $\eta_R = \infty$, and to the natural distribution when both $p = 0$ and $\eta_R = \infty$. Different forms of the truncated exponential transform have been used in sensitivity studies in shields.[16] Also, the form presented here is a special case for which the requirement that $p\mu<1$ is not needed.[17] Equations (32) and (33) are both

indeterminate for $p\mu = 1$. A limit analysis ($p\mu \rightarrow 1$) shows that for $p\mu = 1$, $R = \bar{R} \cdot RN$; i.e., flight distances are selected uniformly on the interval $0 \leq R \leq \bar{R}$. The weight correction, Eq. (33), becomes $WT = \eta_R e^{-\eta}$.

Truncated path selection is a standard option in MORSE. Equation (32) with $p = 0$ is used in the MCNP forced-collision option where η_R applies only to the edge of a region, not to the edge of the entire system. This option guarantees that a collision will occur in the region when a particle enters, regardless of how small it is. The collision weight is adjusted by $(1 - e^{-\eta_R})$. Another particle is created, with weight adjustment $e^{-\eta_R}$, and is deterministically placed on the region boundary, simulating an uncollided flight through the region.

III.D.4. Other Forms of Transport Biasing

A more formal approach to the transformation of the transport equation can be effected by using in Eq. (16), instead of an exponential term, an importance function of the form $I(\bar{r}, E, \Omega) = I(\bar{r})^{\chi(E)} \times I(E) \times I(\Omega)$ used in the TRIPOLI code. Subsequent analysis similar to that in Eqs. (17), (18), and (19) reveals that $\Sigma^*(\bar{r},E,\Omega) = \Sigma_T(\bar{r},E) - \chi(E)\Omega \cdot \nabla I(\bar{r})I(\bar{r}) = \Sigma_T(\bar{r},E) - \chi(E)\mu k(\bar{r})$; where $k(\bar{r}) = |\nabla I(\bar{r})|/I(\bar{r})$, $\mu = \Omega \cdot \Omega_0(\bar{r})$, and $\Omega_0(\bar{r}) = \nabla I(\bar{r})/|\nabla I(\bar{r})|$. Physical analogies of these quantities can be made using the fundamental concepts of gradients and vector field theory. The Ω_0 represents a unit vector perpendicular to the tangent of $I(\bar{r})$. If $I(\bar{r})$ represents a surface of constant importance, then $\Omega_0(\bar{r})$ is everywhere normal to this surface pointing in the direction of the maximum increase of $I(\bar{r})$. The dot product, μ, represents the cosine of the angle between the outgoing particle direction Ω and the direction of the greatest increase of importance, i.e., the most preferential direction. The transform parameter $k(\bar{r})$ represents the (normalized) magnitude of the change in importance along the directions of greatest change, $\Omega_0(\bar{r})$. The form of $k(\bar{r})$ indicates that the importance varies exponentially between surfaces of constant importance along the $\Omega_0(\bar{r})$ directions. That is, $I(\bar{r}) = e^{\pm k(\bar{r})|\bar{r}|}$ is a solution of the differential equation which defines $k(\bar{r})$. Thus, this more general presentation exhibits the same characteristics as that of the exponential transform in the previous sections, and it can be seen from the two definitions of $\Sigma^*(\bar{r}, E, \Omega)$ that $\Sigma_T(\bar{r}, E)\, p(\bar{r}, E) = \chi(E)\, k(\bar{r})$.

Various general aspects of the exponential transform and its cross section are given in reference 18. In this reference it is also shown that the procedure of selecting from the unbiased distribution $e^{-\eta}$, Eq. (25), and adjusting the particle path and weight each time, Eqs. (26) and (27), is equivalent to solving the transformed equation, Eq. (18), and applying Eq. (16). The literature is lacking in examples of use of the entire transformed equation in contrast to use of the standard transport equation with biased sampling and weight corrections (see Section IV.D). Other theoretical aspects of transport equation transformation are given in references 17, 19, and 20.

The form of Eq. (21) was chosen so that with $0 < |p| < 1$, Σ^* is always positive and the path selection procedure is similar to that of the unbiased case. This is the method used in the standard versions of MORSE and MCNP. In using the Σ^* developed in this section, the path selection must be altered as Σ^* approaches zero or becomes negative. In

TRIPOLI, when $\Sigma^* < \epsilon$, a small positive input number, several options are employed, depending on the value of Σ^*, including two similar to those given in the last two paragraphs of the previous section. In each case, a portion of the particle weight is placed determinstically on the next boundary with an appropriate weight adjustment. There is another possible method for treating small Σ^* without a deterministric weight adjustment. The implementation of this method is equivalent to adding an extra transport term $\Sigma_\delta\psi$ to each side of the transformed transport equation, such as in Eq. (18) with the complete importance replacing the exponential term, so that the cross section for transport, $\Sigma_T(\bar{r},E)-\chi\mu k + \Sigma_\delta(E)$, is positive. This extra term is also placed under the scattering integral with a delta-function kernel, $\Sigma_\delta(E')\delta(E'\rightarrow E,\Omega'\rightarrow\Omega)$, so that with probability $\Sigma_\delta/(\Sigma_s+\Sigma_\delta)$ the particle is "scattered" straight ahead with no change in energy or direction. This procedure, without the exponential transform, is the "delta-" or "fictitious-scattering" model used for increasing the collision density in optically thin media or for omitting boundary crossing searches in very complicated geometries.[21]

It is seen that the form of Σ^* derived in this section has the property that surfaces of constant or equal importance can be set throughout a system without regard to the Σ_T of any materials these surfaces might traverse. The procedures for small Σ^* are described above. In addition, the adjusted particle weight will in general be different from that dictated by the importance, to the extent that the transformed kernel is not normalized to unity,[1] as in Eqs. (25) and (31). A procedure of stratified or quota sampling of collision sites is utilized in TRIPOLI, and also in the SAM-CE code,[22] to normalize the kernel. An expression similar to Eq. (32) is used except that the multiplicative term associated with *RN* [which normalizes Eq. (31)] is now the product of the particle weight and the importance function. Multiple collision sites are determined by setting $RN = RN + n-1$, where n is an integer count of the number of sites, until the bracketed argument of the log function becomes negative. The TRIPOLI code selects a new *RN* for each n: whereas, the SAM-CE code selects only one *RN* for the entire n collision sites. This multiple collison site procedure theoretically gives the same expected value, but a smaller variance, than the more conventional method of an initial splitting procedure followed by independent collision site selections from a normalized distribution.[17]

An example of the constant-importance surfaces of a realistic TRIPOLI calculation is given in Fig. 6 (the TRIPOLI literature actually deals with equal-weight surfaces, the inverse of the importance). The preferential directions perpendicular to each surface are shown. The "assumed" path of particle transport from the core to the region of interest, the heat-exchanger, is indicated; i.e., direct penetration through the shield is given little importance. A collision density plot for this calculation is shown in Fig. 7. The relative weight values assocated with the surfaces in Fig. 6 are indicative of the variation of particle weights throughout Fig. 7. Plots such as this are achieved, of course, only after several short calculations and subsequent biasing parameter adjustments.

III.D.5. Discussion of Transport Biasing Methods

In the earlier discussion of boundary splitting and Russian roulette, the only criterion for either process was the ratio of region importances (possibly as a function of energy) at a boundary. If there are any weight-altering processes in effect such as source, collision,

ORNL-DWG 85-9773

IMPORTANCE SURFACES AND DIRECTIONS

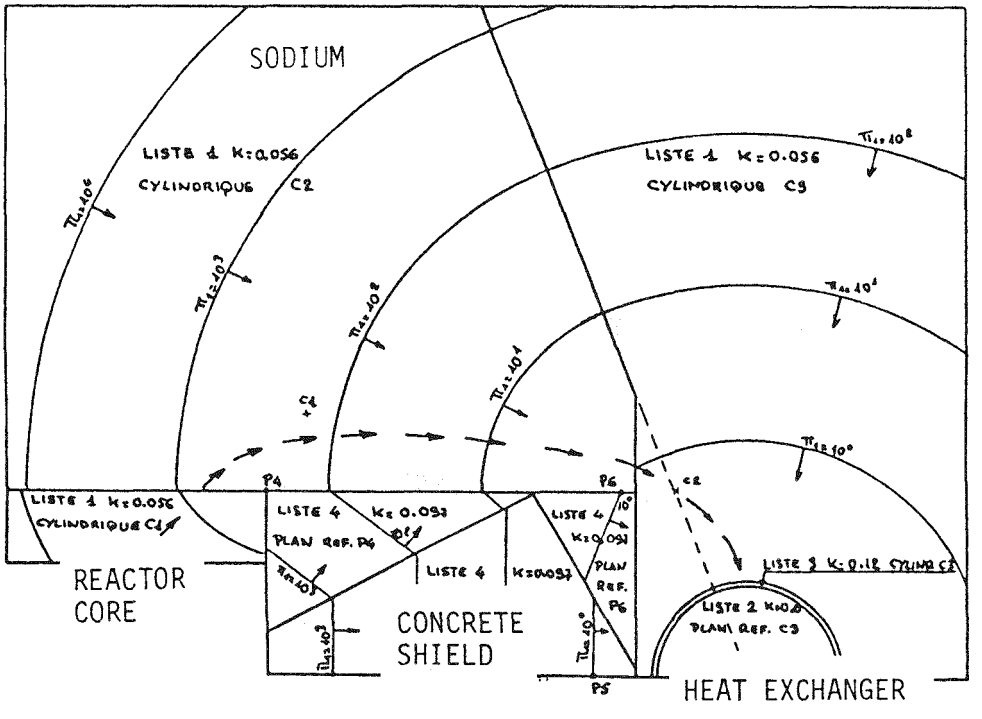

Fig. 6. Importance surfaces and directions.

ORNL-DWG 85-9782

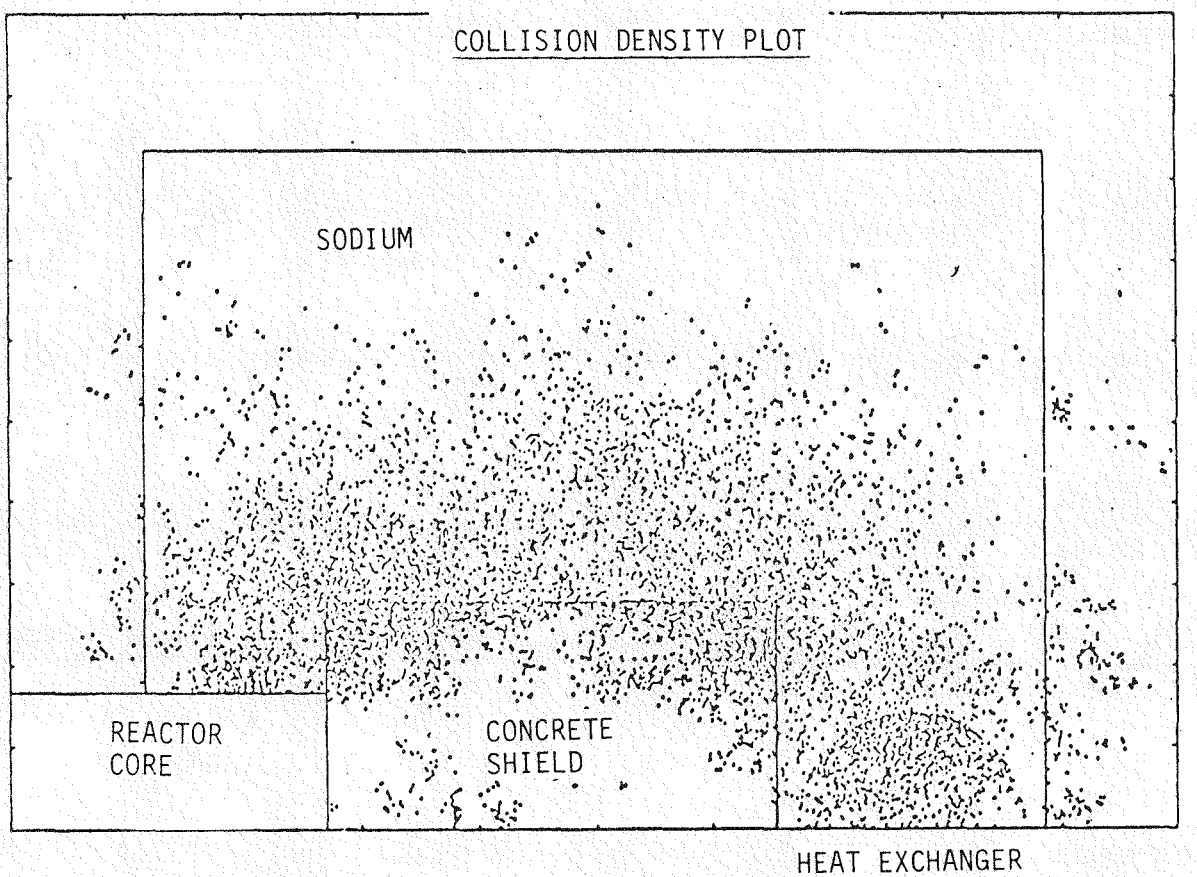

Fig. 7. Collision density plot.

or survival biasing, large dispersions in particle weight can be created throughout the system. It is common practice in any Monte Carlo calculation to use a splitting and Russian roulette scheme based on the particle weight as well as the phase space variables. (This is the "weight-window" which keeps all particle weights in a given importance region between an upper and lower bound.) Otherwise, low weight particles, insignificant to the answer, are accorded the same calculation time as high weight particles, and just a few very high weight particles can create havoc in the statistical uncertainty of the final results. These weight related games can be played at real or artificial boundaries, or in association with the collision process. Boundary splitting and Russian roulette are known to be more efficient, but not so much so that, because of other practical reasons, the collision site is often used for the weight control process.[23]

The problem of weight dispersion and its control can be greatly exaggerated by use of the exponential transform. The problems of picking the transform parameter and weight-window parameters for all importance regions is an enormous task for a complicated system. If the weight-window is set too high in the preferential direction at large distances from the source, the same situation will result as for the example given for source energy selection (Section III.C.2). Most particles which have reached these distances through path stretching with low-weight corrections (WT in Eq. (27) $<<$ 1) will be killed by Russian roulette, and a few higher weight particles will survive. Likewise, if the weight-window is too low in the unpreferred direction, the few high-weight-corrected particles will be split into many lower weight particles. The final result would resemble the unbiased particle population and weight distribution as if no exponential transformation had taken place.

Even if TRIPOLI is not used, it is instructive to examine its input and examples of deep-penetration calculations using the exponential transform. There is a separate geometry input, completely independent of the physical geometry, which traverses the entire system with a series of equal-weight surfaces. These surfaces, the transform parameter k, the preferential directions, and the weight-window are combined automatically by the code so as to minimize splitting and Russian roulette. Creating this weighting geometry is often much more difficult than setting up the physical geometry, and it is done almost exclusively by empirical methods. MORSE has most often been used by employing the exponential transform for deep-penetration problems. The standard version has splitting and Russian roulette at collision sites, but due to the user oriented structure these methods can be implemented at boundaries also.[2] The authors of the MCNP code have traditionally advocated use of boundary splitting and Russian roulette with their code, although some successful work with automatic generation of weight-windows in conjunction with the exponential transform has been recently reported.[15]

It is difficult to determine which of the two methods presented here is better, i.e., boundary splitting with Russian roulette or the exponential transform. It is sufficient to say that either method used correctly and cautiously will give satisfactory results. The boundary method introduces the least weight fluctuation of the two methods, but its improper use could introduce undesirable correlation among histories due to splitting and too much wasted computation time due to Russian roulette. It can be argued that for certain problem types the best transport procedure is to control the particle population with boundary splitting and Russian roulette only, without regard to particle weights (no weight window), and to allow analog absorption at collisions. In some applications where particle

absorption in the form of energy deposition is an effect of interest, analog absorption is necessary; e.g., for determination of pulse height distributions in detector systems or the creation of charged particles.

It is always hoped that improper biasing of any kind will be reflected in the statistical uncertainty of the results. Unfortunately, this is not always clear with the exponential transform. It is possible that a result with an apparently reasonable statistical error will change significantly in another calculation with a different random number sequence. This sequence change might have been done specifically or brought about by changing the transport parameter or some other biasing parameter. It is apparent that the weight fluctuations introduced by the transform can produce results that are not normally (Gaussian) distributed,[24] whereas this distribution is assumed in the interpretation of the calculated uncertainty. Another problem area with the transform is that the path stretching process may cause a significant portion of the particle population, which will ultimately escape the shield and contribute to the result, to skip important intermediate collisions. These omitted collisions will save computation time, but if there exist minima in the cross sections (windows) through which particles could stream, or if gamma-ray production from neutron interactions is significant, then the results may not correctly reflect these physical phenomena. The exponential transform is also the usual method employed for "quick and dirty" calculations. With a value of p sufficiently close to unity, one can always get some particle penetration for a (usually wrong) result. However, the same haste in applying boundary splitting might result in no penetration at all.

The transport process is probably the most important of all the aspects of a deep-penetration calculation (assuming there are no errors elsewhere). If the transport is treated properly, the other processes could be handled in a relatively simple manner. Whereas, if the transport is treated poorly, sophisticated treatment of the other processes is often to no avail. As a result, the literature abounds with various methods for treating the transport process. Unfortunately, these are often of only academic or such highly specialized interest that they are of no general use. In the last decade, however, there has been an increase in the effort to obtain realistic importance functions (adjoints and others) to be used in existing biasing techniques rather than in dealing with the techniques themselves. These methods are discussed in Sections IV and V, where it is pointed out that for application to the exponential transport the correct adjoint function is the "event value function" $W(P)$, rather than the "point value" $\phi^*(P)$, the adjoint flux.

Two separate studies of optimized transport with the exponential transform have recently been made. A two-dimensional multigroup Monte Carlo study of neutron leakage from a spent-fuel shipping cask indicates that there is no significant improvement in efficiency when the exponential transform is included in a calculation where the weight window for splitting and Russian roulette has been automatically set throughout the system from adjoint calculated importances.[25] In this study the transform parameter was determined in an optimal manner from a discrete ordinates adjoint event-value function. This optimized energy- and spatial-dependent parameter was found to never exceed 0.3 in a highly scattering system of steel and depleted uranium. But for gamma-ray transport in the same system, all optimized parameters exceeded 0.9. These same general effects are reported in an iron benchmark example in Ref. 15 (with a "forward-adjoint" generated weight window) where a uniform decrease in calculational efficiency resulted as the transform parameter was increased from 0.2. In a more absorbing medium (concrete), it was

found that the exponential transform had (1) only a marginal beneficial effect when applied to neutrons in a coupled neutron-secondary gamma-ray calculation, (2) some benefit (approximate factor of 2) in a neutron only calculation, and (3) an even greater benefit when applied to gamma rays.

Some conclusions from this discussion indicate that the exponential transform performs best in absorbing systems with some control of large weight-correction fluctuations, with gamma rays being afforded more benefit than neutrons. This is consistent with the premise of the transform that the particle flux is attenuated exponentially and also with semi-empirical methods such as point kernels where gamma-ray scattering can be included with a fit to an exponential function and buildup factor. From these relatively low values of the optimal neutron transport parameter, 0.2-0.3 in these examples, it is apparent that empirically set parameters are often too large. This would lead to too many particles being stretched out of the system (escaping) in the vicinity of the detectors. The resulting low collision density would then lead to low calculated results.

For comparative purposes it must be pointed out that in these optimized studies the weight windows for splitting and Russian roulette were first set before introducing the transform. Thus, a substantial amount of biasing was already in effect before attempting to make an improvement with the transform. In contrast, for empirically determined input schemes the transform is the principal biasing device, and splitting and Russian roulette are used only when necessary for control of large weight fluctuations. No procedures have been reported where the optimization of the exponential transform was done first. A further discussion of this subject is presented in Section IV.E following a presentation of adjoint methods.

III.E. Collision Process

Most Monte Carlo codes have survival biasing (non-absorption weighting) as a standard procedure or option, i.e., all collisions are scatters with the particle weight adjusted by Σ_S/Σ_T. The delta scattering and forced collision methods mentioned in previous sections can be used to increase the collision density in optically thin regions. Generally, the collision process receives little attention as compared to the other processes.

III.E.1. Biasing the Collision

An option for explicitly biasing the collision kernel for the post-collision energy and angle is available in many Monte Carlo codes, but this option is usually the least used of all variance reduction techniques available. The collision kernel is a function of pre- and post-collision energy and angle. The interdependence of the variables depends on the nuclide, particle type, collision type, and energy range. The general biased kernel[26] and weight correction are similar to that for the source spectrum, Eqs. (1) and (2), with the extension to more variables. Various methods of empirically setting the necessary importance functions have not generally been successful. In practice it is often easier to implicitly bias the energy and angle variables separately in the transport and weight-window processes and leave the collision kernel unaltered. In these other processes,

including the source, the importance of the phase space variables toward the result is more apparent than the post-collision energy-angle selection at intermediate collisions in the particle history. The use of collision biasing in conjunction with other biasing techniques increases the difficulty in setting weight-windows.

In TRIPOLI, collision biasing is introduced in a manner consistent with the overall transformed equation as outlined in Section III.D.4. However, the manual[1] warns against the possible abuse and/or inefficiency of the method. The angular importance is taken as $I(\bar{\Omega}) = \Sigma_T(\bar{r}, E)/\Sigma_T^*(\bar{r}, E, \bar{\Omega})$. The weight correction due to $I(\bar{\Omega})$ cancels with a similar term from the exponential transform; i.e., if this angular biasing were applied to Section III.D.2, specifically Eq. (29), then the combined spatial and angular correction term in Eq. (27) would only be the exponential term with a unit denominator (see Section IV.E).

The difficulty with this angular importance function procedure is that the general biased collision kernel cannot be integrated in closed form for the creation of a cumulative distribution and subsequent outgoing particle direction selection. In TRIPOLI a method has been devised which selects several outgoing directions from the unbiased kernel and then creates a discrete distribution with these directions input into the biased kernel, from which one of the outgoing directions is chosen and a weight adjustment is made. A similar but more complicated biasing procedure is available in TRIPOLI for choosing outgoing directions in the vicinity of voids and streaming paths.[27]

For isotropic scattering, the above angular biasing procedure is analytic. An application of this method has recently been utilized in the MCNP code with gamma ray scattering angular biasing.[28] Here, the straightforward but somewhat lengthly integration of the biased Klein-Nishina kernel is also analytic.

In MORSE, a multigroup code which selects the outgoing energy group first and then an angle, collision energy biasing is a standard option that has been little used. The recent work on shipping casks mentioned in Section III.D.5 employed a one-dimensional discrete ordinates adjoint as the importance function in MORSE. Here the adjoint flux ϕ^* is used directly as the post-collision energy biasing parameter, much in the same manner as for source energy biasing. It was found that optimized collision biasing, like the exponential transform, was ineffective when the weight window for splitting and Russian roulette was determined from ϕ^*. Several additional angular biasing procedures have been incorporated into MORSE, but none of these are part of the standard code.[29,30] A procedure for including general angular biasing with $I(\bar{\Omega})$ in the MORSE code is given in Section IV.E.

III.E.2. DXANG and DXTRAN Biasing

The MCNP code does not have an explicit collision biasing option, but one can be included as a "patch" (non-standard programming available for specific use). However, a conceptually simple form of angular biasing, designated DXANG,[31] has recently been made available to the MCNP code. Here, a spherical region, not necessarily part of the physical system geometry, is defined toward which it is desired to scatter particles (see Fig. 8, top). At each particle collision (and source event) a cone is defined by the solid angle $\Delta\Omega$ subtended at the collision site by the sphere. A secondary particle is created

ORNL-DWG 85-9854

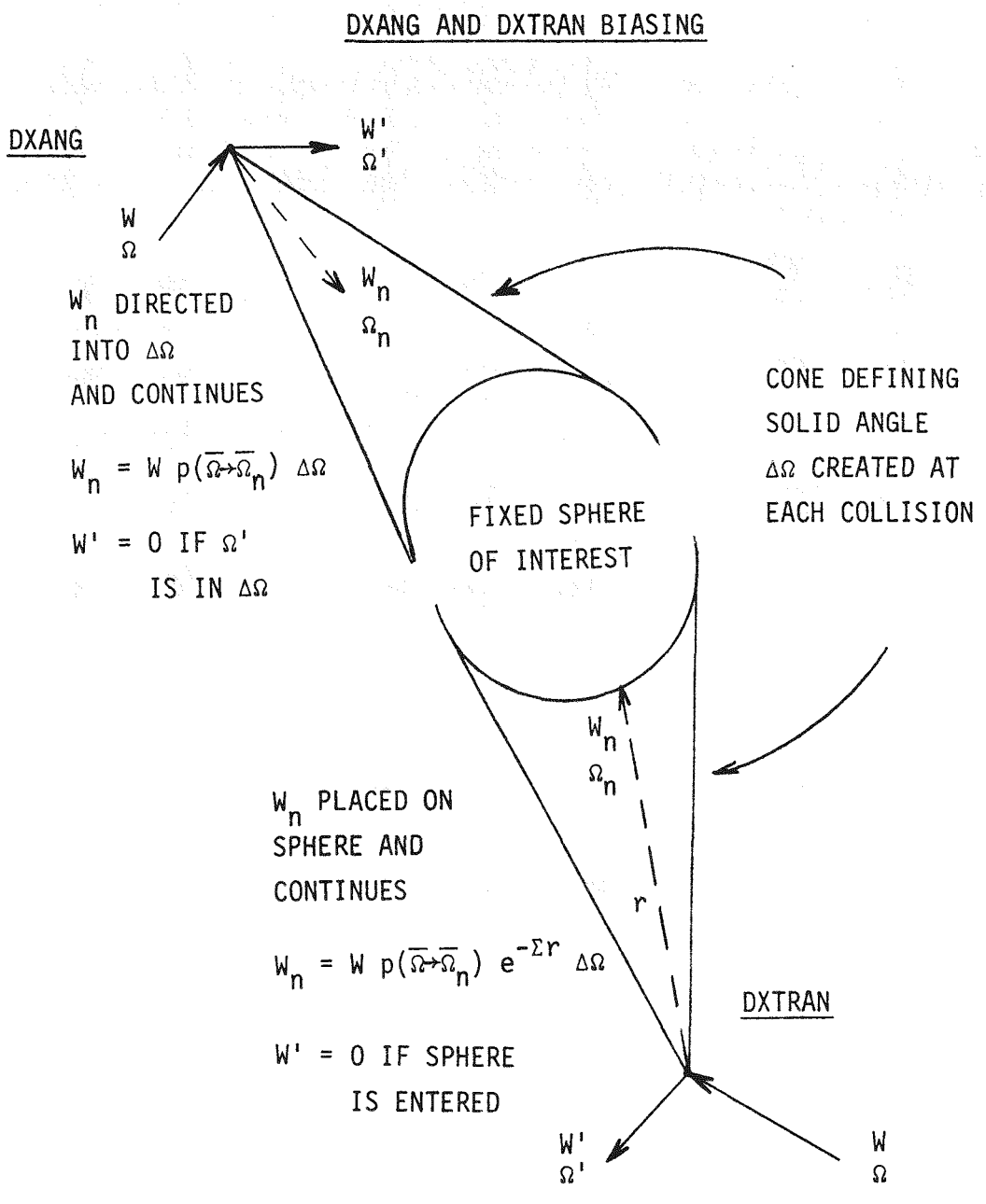

Fig. 8. DXANG and DXTRAN biasing.

with a direction $\overline{\Omega}$ chosen uniformly in the cone and with a weight adjusted from that of the primary particle before collision by multiplying by the easily-computed $\Delta\Omega$ and by the angular scattering probability $p(\Omega)$, as for a next-event point estimator. The primary particle continues normally, producing other particles at subsequent collisions, unless it actually scatters into the cone created for a secondary particle, in which case it is terminated. Each secondary particle becomes immediately another primary particle having the same transport, production, and termination characteristics as the particle which created it. In this manner, many particles are directed toward some region of interest without the difficulties associated with conventional angular biasing. The extra weight accumulation of the many reduced-weight secondary particles is offset by the occasional termination of a full-weight primary particle.

The DXANG method has been applied to a multi-leg duct problem similar to Fig. 1 (see Fig. 9) with spheres surrounding the interior corner areas and the duct exit region. Using a code-generated direction- and region-dependent weight window (Section IV.B), this method provided a factor of 25 increase in efficiency over a standard MCNP calculation of the same problem. In Fig. 9 the indicated importance regions were set empirically, as well as the numbers designating toward which sphere the new particles in a given region are directed.

Another MCNP biasing option is the DXTRAN method. Although it has been a standard code feature for several years, DXTRAN is, in concept, an extension of the more recent DXANG method. After the secondary particle has been directed into the cone, as described above, it is further transported deterministically and placed on the spherical surface with an additional (exponential attenuation) weight reduction before beginning its actual transport (see Fig. 8, bottom). The primary particle is now terminated only if it enters the sphere.

The justification of these two methods can be illustrated with a simple example using DXTRAN. In a standard utilization of next-event estimation to a boundary (see Section III.F.2), the estimation is made with the same W_n estimator shown in the bottom of Fig. 8, but this "created particle" (the estimation trajectory) is terminated at the boundary. If the random walk particle crosses the surface in question, it continues normally but makes no contribution to the estimate (it may, of course, contribute to a separate analog crossing estimator, independent of this next-event estimator). In the DXTRAN method, the created particle performs both functions — it contributes to the estimate on the boundary, if desired, and continues as the "random walk" particle in the sphere. Thus, if the original particle enters the sphere, it must be terminated to preserve the fair game.

Neither the DXANG nor DXTRAN method is applicable for primary particle collisions inside the designated sphere. In practice, the scattering into the cone may itself be biased, and probabilities of creation may be assigned to control the proliferation of secondary particles, just as for next-event estimation (Section III.F.5). In fact, DXTRAN has many of the characteristics of next-event estimation and has been found to be most useful in systems with widely separated scattering regions.

ORNL-DWG-85-17435

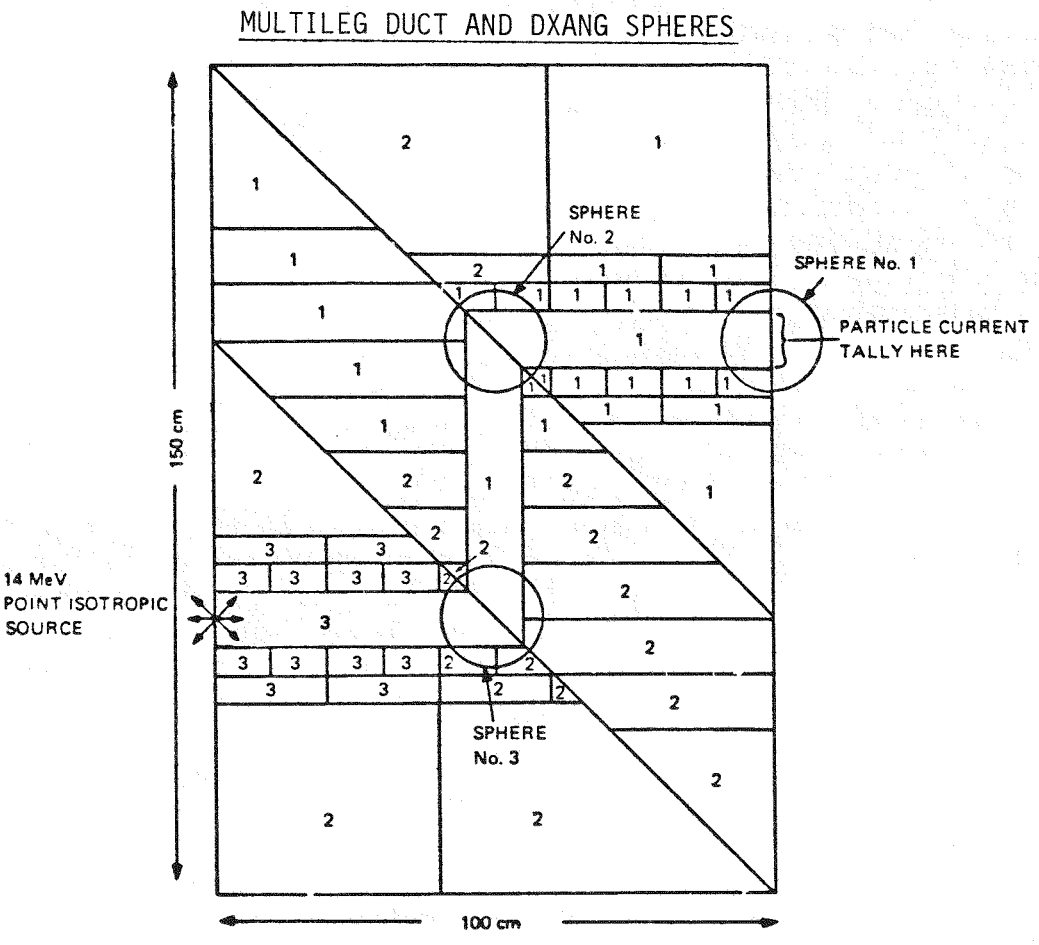

Fig. 9. Multileg duct with DXANG spheres.

III.E.3. Albedo Scattering

The albedo scattering procedure is a method for eliminating time-consuming particle tracking in spatial regions of relatively high collision density but from which no direct contribution is made to the results.[32] Particles are returned to a real collision medium from the albedo medium by reflection from an interface surface between the two media with the correct change in the energy and angular variables as if analog tracking had taken place in the albedo medium. The albedo scattering function for a given material is generally given as $\alpha(E_o,\theta_o,E,\theta,\mu)$, where E_o and E are the entrance and exit energies at the interface, θ_o and θ are the entrance and exit polar angles with respect to the outward directed normal vector at the interface, and μ is the cosine of the angle between θ_o and θ (see Fig. 10). It is assumed that the interface is of such extent that there is no spatial variation, i.e., the position on the interface does not change. The use of time dependence here is questionable, except for specular (mirror image) reflection. This is a special case where $E = E_o$, $\theta = \theta_o$, and is used only to take advantage of geometrical symmetry. Specular reflection precludes the general use of any "next-event" type of estimation process and requires special attention to the normalization of results due to symmetry in the source and detector regions.

The principal disadvantage of the general albedo method is the unavailability of reliable and inexpensive data [the $\alpha(E_o,\Omega_o,E,\Omega,\mu)$], and many approximate forms of the distribution function exist for special cases. The MORSE code (multigroup) uses albedo data generated from discrete ordinates codes;[2] an acceptable procedure since the regions of application are almost always easily approximated by one or two dimensions. However, these data are voluminous and expensive to generate, and must be regenerated for any significant changes in the geometric or material description of the system. In the absence of data, the albedo procedure may be simulated by clever application of direction dependence in the transport biasing process. The proceedings of the recent international shielding conference in Tokyo give many examples of albedo Monte Carlo use.[11,33] Many of these applications use one-dimensional (azimuthal symmetry of the emerging particle) albedos determined from inexpensive invariant imbedding calculations.

The albedo procedure is useful in simulating scattering in the reflector of a reactor and also in the walls surrounding a duct or pipe through which particles are streaming. If the duct is void (or contains only air), the calculation could be completely albedo scattering with no ordinary interactions. However, for bent or multi-leg ducts it has been found necessary to include real (analog) scattering in the protruding corners of the duct interior (see the shaded areas of Fig. 1) to correctly simulate the particle transmission. For some distributed source and detector regions, it may also be necessary to include real scattering in regions adjacent to the entrance and exit of the duct. The position and size of the shaded areas in Fig. 1 must be set empirically (usually in terms of mean free paths). The use of albedo material surrounding the passage in Fig. 1 naturally precludes any particle transport continuing straight ahead from the first leg of the duct through the shield to the exterior region; however, a special "next-event" estimator has been devised which allows transmission across albedo surfaces.[34]

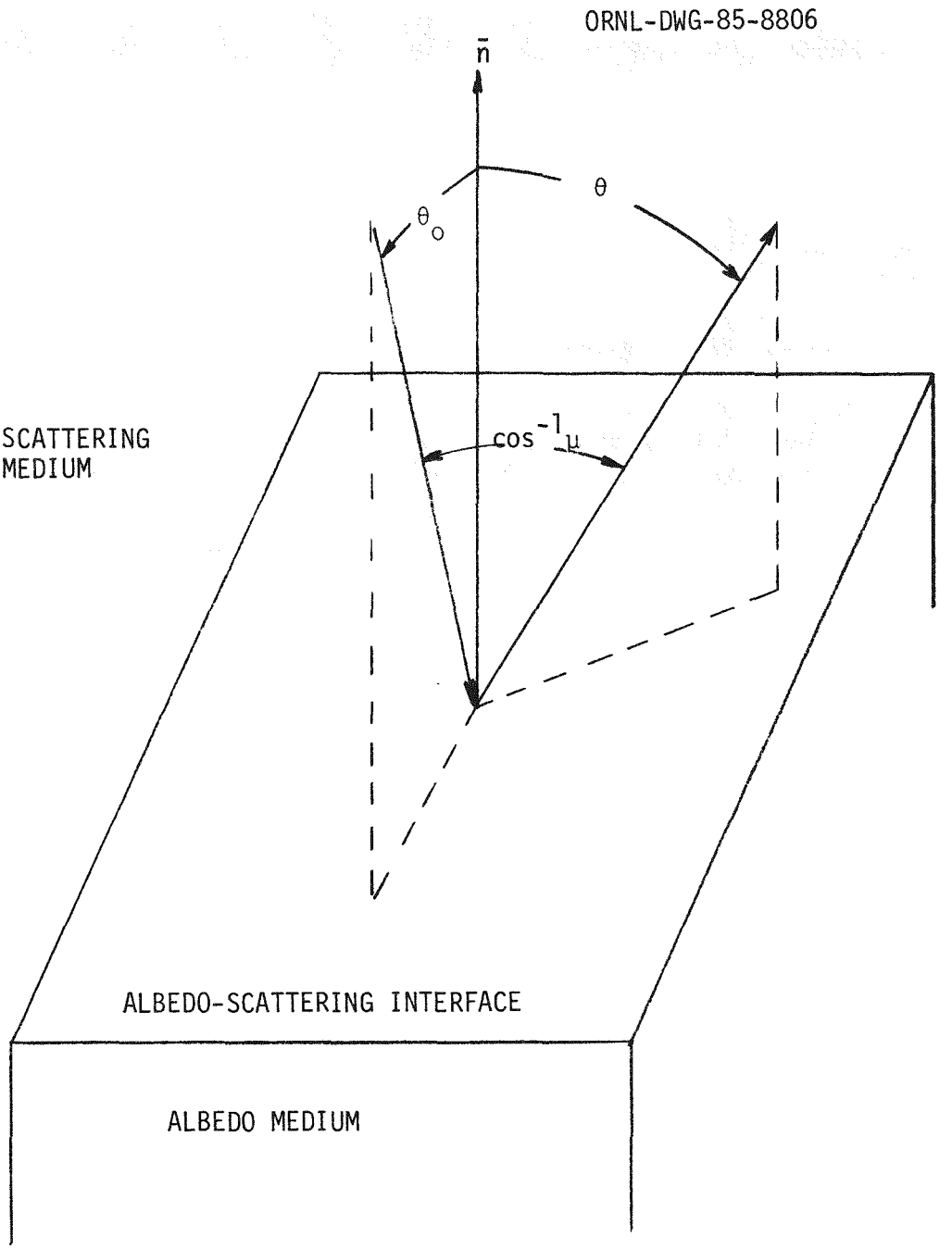

Fig. 10. Albedo scattering geometry.

III.F. The Estimation Process

If all the preceding processes associated with a deep-penetration calculation were performed correctly (or ideally), the estimation of results would be a simple task. There should be an ample number of particles in the detector region to score with the standard techniques such as collision density or path length estimation in a volume or surface crossing estimation at a boundary. Unfortunately, for many well formulated calculations this situation is often not realized, and undersampling due to inadequate and/or improper use of the system importance function leads to sparse particle population in the detector regions, producing (usually) low answers with high variance. In an attempt to overcome this problem, the method of "next-event" estimation (also known as "statistical" estimation) has long been used, and often over-used. Here, contributions to the result are made from random walk events (source and collision) which occur outside the detector region.

From each random walk scattering event at $\bar{r}$, the next-event estimation contribution to the particle number density (current) at some location $\bar{r}' = \bar{r} + R\bar{\Omega}$ at the leading edge of a convex detector region is (see Section III.F.4 for uncollided estimation from source events)

$$J = WT\, p(\bar{\Omega}) e^{-\eta} \Delta\Omega \quad , \tag{34}$$

where

WT is the random walk particle weight before collision (includes Σ_S/Σ_T for survival biasing);

$p(\bar{\Omega})$ is the probability per steradian of scattering toward the detector, numerically evaluated from the (unbiased) distribution used for random angular and energy selection in random walk scattering;

$\Delta\Omega$ is the solid angle, in steradians, subtended at the collision site by the detector region; and

$\eta = \int_0^R \Sigma_T(\bar{r},E)\,dR'$ is the number of mean free paths along the distance R.

III.F.1. Next-Event Estimation in a Large Solid Angle

When the detector region is of sufficient extent, i.e., $\Delta\Omega$ is large enough, Eq. (34) can be simplified using the direction and energy as determined from the random walk event. After each event the estimation process determines if the random walk trajectory could intersect the detector if it were extended sufficiently [$p(\bar{\Omega})\Delta\Omega = 1$ in Eq. (34)]. If not, the estimation is abandoned [$p(\bar{\Omega})\Delta\Omega = 0$], and the random walk continues (if there are multiple detector regions, each region must be tested for intersections from each event). If an intersection occurs, the particle current at the detector is $WT\, e^{-\eta}$. This estimation search could be very simple; e.g., at the top of Fig. 11 if the detector region were the entire right-most boundary of the shield, it would have to be determined only if the

trajectory vector had any component directed toward the boundary. If the detector region were only a portion of the boundary, the intersection decision could be determined analytically. For more complicated cases, the time-consuming code geometry trace procedure will be necessary, but this trace will be necessary in any case to determine η if an intersection is made.

The flux for the boundary crossing estimation is, like the analog estimator,

$$\phi = \frac{WT\ e^{-\eta}}{|\cos\ \alpha\ |} \quad , \tag{35}$$

where α is the angle between the estimation trajectory and the normal vector to the boundary. Any standard method for grazing angles ($\alpha \approx 90°$) must be applied; e.g., if $|\cos\alpha| < \epsilon$, set $|\cos\alpha| = \epsilon/2$.

If the detector region is a void volume (bottom of Fig. 11), the track length estimator is used for the flux estimate,

$$\phi = WT\ e^{-\eta}\ \ell \quad , \tag{36}$$

where η is determined for the estimation trajectory to the leading edge of the detector, and ℓ is the length of the trajectory through the void, determined either analytically, by the geometry trace, or by the average chord length in the volume, if known.

For thin non-void volumes, the flux is

$$\phi = WT\ e^{-\eta}\ \frac{\left[1-e^{-\Sigma_T(E)\ell}\right]}{\Sigma_T(E)} \quad , \tag{37}$$

where $\Sigma_T(E)$ is the cross section of the detector material, and E is the estimation energy.[17] These flux estimates must be divided by the appropriate detector volumes or surface areas to give the standard flux dimensions.

III.F.2. Next-Event Estimation in a Small Solid Angle

When the detector region is small, the probability of the trajectory-detector intersection is also small, and the preceding procedures become very inefficient. When $\Delta\Omega$ in Eq. (34) is small, the estimation trajectory must now be directed toward the detector irrespective of the random walk direction. For this situation Eq. (34) is applied where $p(\Omega)$, usually given as $p(\mu)/2\pi$, is the probability of scattering through the angle $\theta_{1,2} = \cos^{-1}\mu$ between the random walk trajectory before the event and a randomly chosen trajectory in $\Delta\Omega$ between the event site and the detector (see Fig. 11, center).

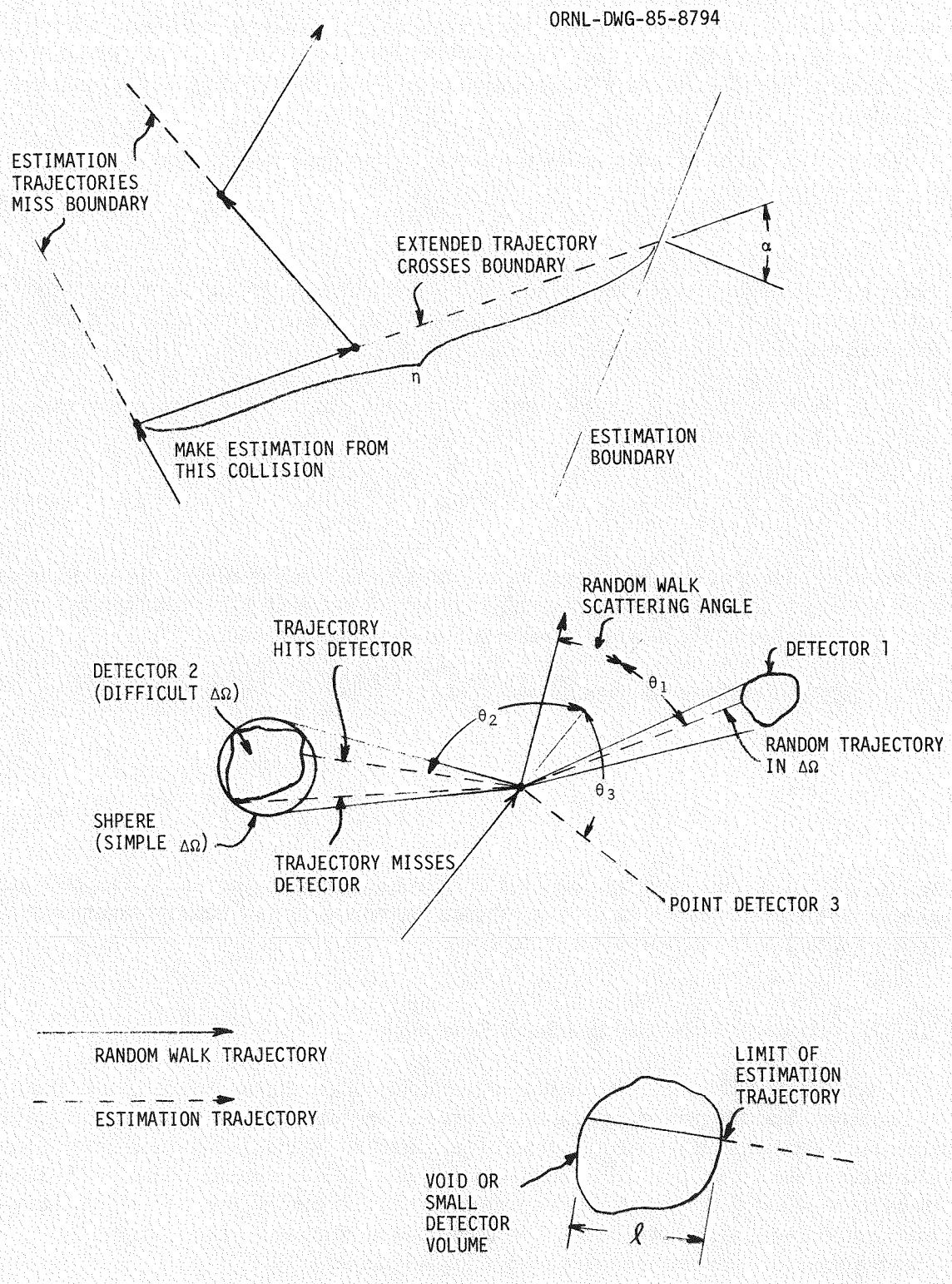

Fig. 11. Next-event estimation geometries.

For a spherical detector, $\Delta\Omega$ is determined analytically. There also exist expressions for the solid angle of a cylinder relative to a point.[35,36] A general procedure is to completely surround the detector by a sphere (the smallest possible) and determine if the randomly determined trajectory in the solid angle of the sphere passes through the detector.[22] If not, an estimation is not made. If the trajectory intersects the detector, Eq. (34) is evaluated using $\Delta\Omega$ for the sphere. The flux estimators in Eqs. (35), (36), and (37) are modified by the inclusion of the $p(\mu)\Delta\Omega/2\pi$ term. This procedure becomes inefficient if the detector region does not fill a significant portion of the sphere.

III.F.3. Next-Event Estimation to a Point

When Eq. (34) is divided by the ΔA, the projected cross-section area of the detector relative to the event site, the flux is given directly. In the limit as the detector becomes vanishingly small, the ratio $\Delta\Omega/\Delta A$ becomes $1/R^2$ and Eq. (34) becomes the "point detector" estimator:

$$\lim_{\Delta A \to 0} \frac{J}{\Delta A} = \phi = WT \frac{p(\mu)}{2\pi} \frac{e^{-\eta}}{R^2} \quad . \tag{38}$$

Few aspects of Monte Carlo radiation transport have received more attention than this estimator. In theory it converges to the correct answer but with infinite variance due to the singularity at $R = 0$. In practice the point detector estimator usually works well in non-scattering media but can give low results with deceivingly low variance in any application due to undersampling in the random walk; whereas, an analog estimator variance would more correctly indicate this undersampling. But when a collision event occurs very close to the detector point, the large contribution from this event due to $1/R^2$ can dominate the summation of the contribution from all other events from all particles in the calculation. Fortunately these large answers always have a very large variance. It is the correct combination of the many low answers and the very few high answers that gives the theoretically correct answer. But for this to happen in even a very long Monte Carlo calculation would be fortuitous. An example especially contrived to illustrate these effects is given in reference 3.

To overcome the low result problem, a sufficient number of particles must collide in some proximity to the detector. For example, in Fig. 2, if the particle population diminishes significantly before reaching the right-most regions near a point detector outside the shield, next-event estimation results might be low. In reality it is the contributions from collisions that have the least material attenuation on reaching the detector that dominate the results. Now, when the other processes are altered to increase the particle penetration, use of the point estimator might no longer be needed if analog estimators are sufficient. However, there is a large middle ground here, and point estimation has been used correctly and successfully for many years, especially for detectors in voids.

Next-event estimation to a point is a standard feature of TRIPOLI and MCNP. In MORSE a user routine must be written for each calculation, although several such routines are given in the report, sample problems, and code package. One of these gives a contribution to all possible energy groups allowed by $p(\mu)$ instead of randomly selecting the group.

A thorough coverage of the point estimator methods which have been developed to overcome the problems of collisions close to the detector will not be attempted here, but a summary of some of the commonly used approaches is as follows:

1. Use the estimator only for a detector in a void, or a sparsely dense material such as air, and place it at least a few centimeters away from a colliding medium interface. This use fits the classical shielding problem in Fig. 2 with the detector located on the far side of the shield from the source.

2. If the detector is in a colliding medium, surround it with a sphere of radius R_o and either (a) fill the sphere with void or (b) allow collisions in the sphere but do not make estimations to the point detector from it. If only a very few collisions, which would otherwise ruin the calculation, are omitted, little effect should be noticed; however, if many particles are found to enter or collide in the sphere, another estimation method should be chosen.

3. Assume the flux in the sphere is uniform and isotropic. Then the contribution from each collision in the sphere is

$$\phi = \frac{WT\left(1-e^{-\Sigma_T(r,E)R_o}\right)}{4/3\ \pi R_o^3\ \Sigma_T(r,E)} .$$

   This procedure is available in TRIPOLI and MCNP.

4. Use any of the various "once-more-collided" techniques for collisions inside or directed inside the sphere.[3] An imaginary intermediate collision point is chosen and the contribution is made to the point detector from this new location. It can be shown that this procedure reduces the $1/R^2$ singularity of the point detector to a $1/R$ singularity with a corresponding variance reduction. This method is an option in MCNP but has been little used.

5. For the special situation of rotational symmetry for the point detector, a ring-detector method has been devised.[25] Here a point detector is chosen on the ring in such a manner that sampling near the collision point is increased, and the $1/R^2$ singularity is eliminated.

The R_o for the sphere, where needed, must be set empirically; if it is too small, the particular method approaches the original point method. If R_o is too large, methods (2) and (3) become inaccurate, and method (4) can greatly increase the calculation time. Also, complications may arise if the R_o from two different point detectors overlap or if there is a change of material or some other geometric complexity inside the sphere. If R_o is set in terms of mean free paths, instead of distance, then R_o is energy dependent.

III.F.4. Next-Event Uncollided Estimation

The form of the preceding equations in this section implies a scattering event, but an uncollided contribution must also be made to each detector for each source particle for next-event estimation. The general equation, Eq. (34), applies and a value of $p(\Omega)$ is determined from the source angular distribution $S(\Omega)$ from which the initial particle direction is randomly chosen for the random walk. But the two processes, source selection and estimation, are completely independent, just as for scattering events and estimation. If $S(\Omega)$ is a biased distribution, the WT must contain the weight correction. The position, energy, time, and weight variables are usually the initial random walk values, or they may be selected independently. The form of the uncollided estimator is usually the same as that from collisions. In simple cases, this uncollided source contribution may be done analytically. The contribution to a detector from uncollided source particles in a pure deep-penetration calculation is usually negligible, but for streaming or other non-penetration calculations this contribution can easily exceed the collided contribution. If there is any particle generation present (secondary gamma rays or fission neutrons), an uncollided source calculation must be made for each generated particle. The uncollided calculation is treated with input data in TRIPOLI. In MORSE a special routine must be written for non-isotropic $[S(\Omega) \neq 1/4\pi]$ source angular distributions; in MCNP only for non-standard sources. This can become somewhat complicated for other than isotropic sources, but many examples are given in the reports and sample problems. Any source coupling procedure (see Section III.C.4) must also include an uncollided contribution if next-event estimators are used. This necessity will add complexity to any coupling scheme.

III.F.5. Next-Event Estimation Probabilities

One of the chief disadvantages of next-event estimation is that it is extremely time consuming, even in its simplest form. The particle history estimation time can easily exceed the random walk time due to the geometry trace necessary for determining the number of mean free paths, η, between the event and detector sites. This calculation time goes up proportionally for multiple detector estimations from each random walk event. It is in the regions farthest from the detector (usually closest to the source with many collisions) whose contributions require more computation time but have the smallest values. It is a common practice to assign probabilities of estimation to different spatial regions of the system (and possibly for the other variables) to determine if a contribution is to be made to a specific point detector from each event in that region.[37] If, by random choice, a contribution is made, the value is divided by the probability, conserving the "fair game." If no contribution is made, the process moves on to the remaining detectors and then on to the next collision. This process is a standard option in MCNP and TRIPOLI and is easily implemented in MORSE, and it can save an enormous amount of computation time. In the absence of any adjoint importance information, these probabilities must be set empirically. This is usually based on the event-detector distance. If radiation streaming paths or other voids exist in the system, the distance criterion should in theory be based on mean free paths, but this would require performing the geometry ray trace at each collision that the method is trying to avoid.

Another simple procedure, an option in MCNP, involves testing a partial contribution to the estimator before the time-consuming determination of the exponential attenuation term. This value, $WTp(\Omega)/R^2$, is compared to a preliminary result compiled normally. A Russian roulette game is played if the incomplete estimation contribution is already below some fraction of the preliminary result. Upon survival, the geometry ray trace is made and the estimator is completed with the attenuation term and a correspondingly increased survival weight.

III.F.6. Next-Event Estimation Variance

It is known from experience that next-event estimation, like the exponential transform, can produce results that are not normally distributed. Repetition of a poorly constructed calculation with only a change in the random number sequence may show a change in results not predicted by the computer-generated uncertainties. This method is usually the estimator of choice for "quick and dirty" deep-penetration calculations because it always gives some answer, whereas more nearly analog approaches might give nothing for poorly conceived transport processes. When combined with the exponential transform, the variance problems of the two techniques are exaggerated. Next-event estimation will produce adequate results when the other processes of the calculation are treated properly and when the problems associated with collisions close to a point detector are overcome. More formal approaches to next-event estimation and its variance characteristics are given in references 38 and 39.

III.F.7. Delta-Scatter Next-Event Estimation

One of the common uses of the delta scatter method (Section III.D.4), independent of the exponential transform, is to eliminate the need for the geometry ray trace in the random walk by effectively creating a homogeneous transport medium. It is this geometry feature that makes next-event estimation so time consuming, and methods have been developed to apply delta-scattering to the estimation process.[21] The procedure is essentially a one-dimensional delta-scatter random walk along the trajectory from the event site to the detector. However, delta scattering itself can be inefficient in terms of computation time relative to standard methods in both the random walk and estimation process for any but very complicated geometries, a situation not usually encountered in deep-penetration calculations. Collision site distances (now shorter, requiring more collisions) are determined by the maximum cross section in the system for a given energy, $\Sigma_M(E)$, and often in resonances regions, if other than the maximum value is used, a negative weight correction is necessary (negative weight can be disastrous since the results, based on the weight, decrease; but the variance, based on the square of the weight, still increases). Only with probability Σ_T/Σ_M does a real collision occur.

Use of multigroup Monte Carlo can alleviate this maximum cross-section problem somewhat. There is also a procedure for including the delta-scatter concept directly in the multigroup cross-section structure. In the standard Legendre polynomial expansion for the scattering angle cosine μ, an extra or fictitious term is added.

$$\Sigma_{g\to g'}(\bar{r},\mu) = \sum_{\ell=0}^{N} \frac{2\ell+1}{2} \left[\Sigma^{\ell}_{S_{g\to g'}}(\bar{r}) + \Sigma_{\delta_{g\to g'}}(\bar{r})\delta_{gg'} \right] P_{\ell}(\mu) \quad , \tag{39}$$

where the delta function indicates that the delta cross-section term in Eq. (39) applies only to within group ($g\to g$) scattering. The expansion for a delta function (here straight ahead scatter, $\mu = 1$) implicit in Eq. (39) is

$$\delta(\mu-1) = \sum_{\ell=0}^{N} \frac{2\ell+1}{2} P_{\ell}(\mu) \quad . \tag{40}$$

Equation (40) is very approximate for low order expansion (small N), but for large systems with multiple scattering, the errors tend to cancel out. The spatial dependence of the delta term in Eq. (39) can be adjusted so that the total cross section is constant throughout the entire system.

Multigroup Monte Carlo codes using cross sections of the form of Eq. (39) now not only need no geometry tracing procedure, but also no procedure for differentiating between collisions with real or fictitious materials. The code operates as if the entire system were completely homogeneous, and the material attenuation in the general next-event estimator of Eq. (34) is $e^{-\Sigma_M(E)R}$, i.e., regardless of a Σ_T variation between event and detector sites. The problems associated with point detectors are still present, and special attention is required for voids. Voids of any significant extent should not be modified to include delta scattering [Eq. (39) with $\Sigma^{\ell}_{S_{g\to g'}}(r) = 0$] but should be treated in an analog manner. The R in the estimator material attenuation should be only for non-void distances.

III.F.8. Coupling Methods for Estimation

If part of the system under consideration which includes the detector region can be treated in one or two dimensions or if acceptable differential results from any method exist for this region, a coupling procedure similar to that described for the source process can save considerable computation time. For the source coupling, a forward Monte Carlo source is created from some forward calculated results. But for estimation coupling the forward Monte Carlo is combined with an adjoint calculation whose source is taken from the physical detector region. Mathematically the response is (for coupling in one direction across a surface)[13]

$$\lambda = \int_{\text{area}} d\bar{r} \int_{E} dE \int_{\bar{n}\cdot\bar{\Omega}>0} d\bar{\Omega}\ \phi(\bar{r},E,\bar{\Omega})\ \phi^{*}(\bar{r},E,\bar{\Omega})(\bar{n}\cdot\bar{\Omega}) \quad . \tag{41}$$

The $\phi(\bar{r},E,\bar{\Omega})(\bar{n}\cdot\bar{\Omega})$ is the Monte Carlo current $J(\bar{r},E,\bar{\Omega})$ at the boundary, and ϕ^{*} is the adjoint flux from the other method. The integral is over the boundary surface only for particles directed into the coupling region. The current J can be either from the transport process directly (the particle weight) or from one of the two current estimation processes

from Eq. (34). As for the source procedure, care must be taken in treating the calculations in the vicinity of the coupling boundary. It is, of course, possible to use coupling methods for both the source and estimation in the same calculation.[40] Equation (41) also applies for the reversal of the forward and adjoint connotation – see Section IV.

An example of present coupling methods, and also geometry capability, will be illustrated from on-going work at ORNL in the joint U.S.-Japanese government sponsored dosimetric reevaluation of the nuclear weapon radiation environments in Hiroshima and Nagasaki. Volumes of data exist on the medical histories of the survivors, and their descendents, and of the bombings of these two cities. In order to correlate these medical records with an absolute cause-and-event relationship, a calculational scheme has been devised to determine the radiation levels inside large, reinforced concrete buildings resulting from calculated, unclassified output spectra from each of the two weapons. Only those survivors located in the interior lower floors of these large buildings received radiation injury independent of thermal, blast, structural collapse, flying debris, or other types of injury. One such building, located approximately 500 meters from ground zero in Nagasaki, is that shown in a post-blast photograph in Fig. 12. Although the truss-supported roof and an upper floor auditorium partially collapsed, the lower floors remained structually intact, and many individuals survived there, receiving various degrees of radiation-only injury.

A computer mock-up of this building, drawn with the JUNEBUG plotting package of the MARS array geometry module in MORSE,[2] is shown in Fig. 13. The picture is drawn at the approximate orientation with respect to the weapon burst 500 meters above the building. This geometry module was originally created to treat finite repeating arrays of reactor fuel pins, elements, and assemblies. The vertical and horizontal lines on the exterior of the drawing indicate the array matrix into which the building was modeled. The individual segments were constructed independently and placed in the array. Any identical repeating segments were modeled only once and placed in the array by appropriate translation and rotation input instructions. The interior of the building is a maze of thick, reinforced walls, floors, columns, beams, and girders reproducing the earthquake-resistant design of the building. In addition, localized areas of the building can be modeled in greater detail to include furniture, file cabinets, bookshelves, etc., independent of the rest of the building.

The calculational model involves starting adjoint particle histories (see Section IV) from points in these localized areas and following them, or estimation trajectories, through the building to a 200-m-long $\times$ 140-m-wide $\times$ 80-m-high coupling surface surrounding the 70-m $\times$ 17-m $\times$ 14-m building. At this surface, the MORSE spatial, energy, and angular adjoint current, $\phi^*(\vec{n}\cdot\Omega)$ in Eq. (41), is coupled with a DOT two-dimensional discrete ordinates forward flux determined from a free-field, air-over-ground calculation resulting from the weapon source. At this coupling surface it was assumed that the presence of the building did not perturb the free-field flux. Depending on the starting location in the building, various degrees of adjoint source angular biasing were used in order to direct more particles out the doors and windows rather than through the structural material. Preliminary results, from the weapon neutron output only, give dose levels at the

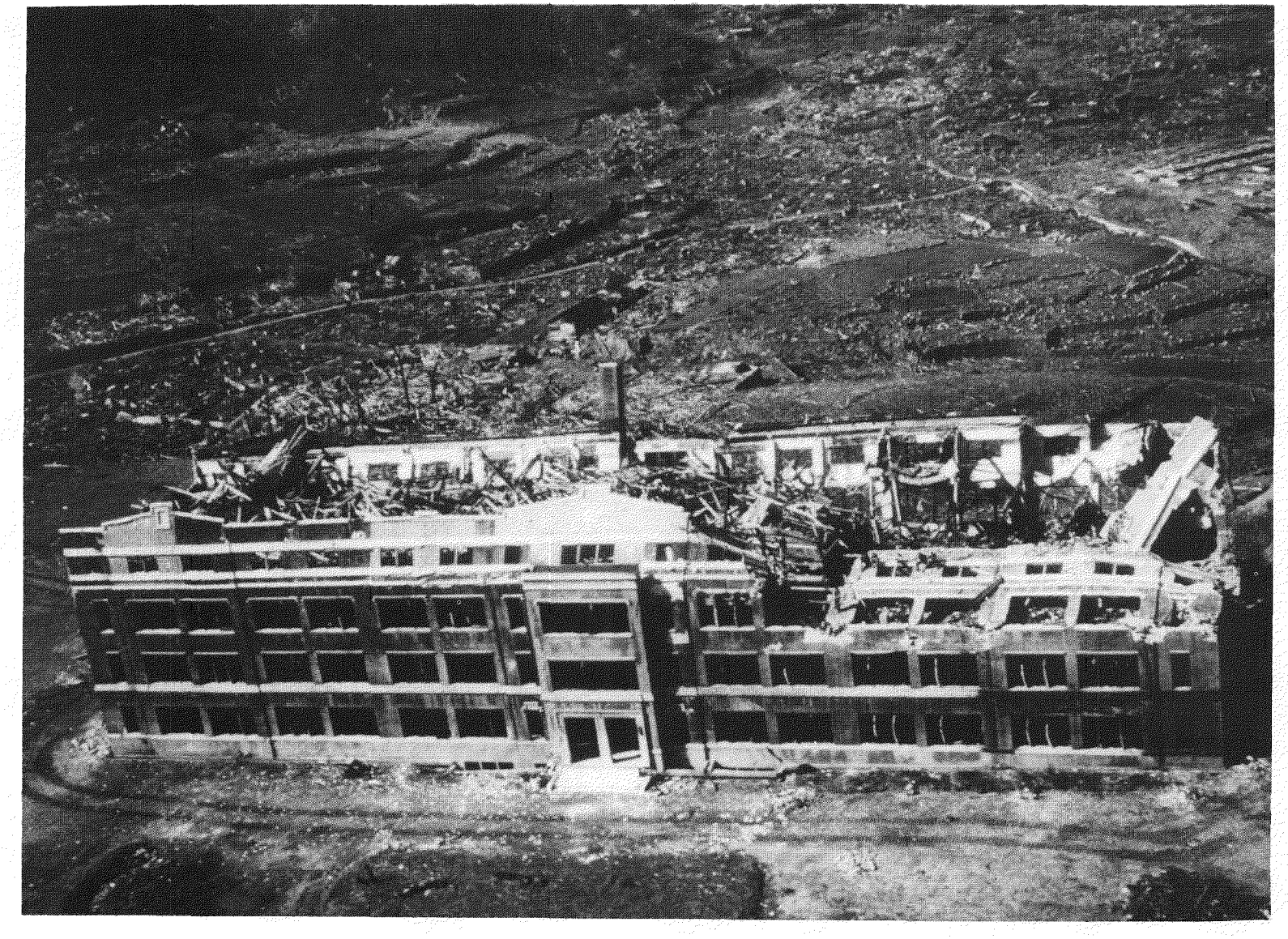

Fig. 12. Reinforced concrete building in Nagasaki (Chinzei School, used as a factory machine shop).

ORNL-DWG 85-9780

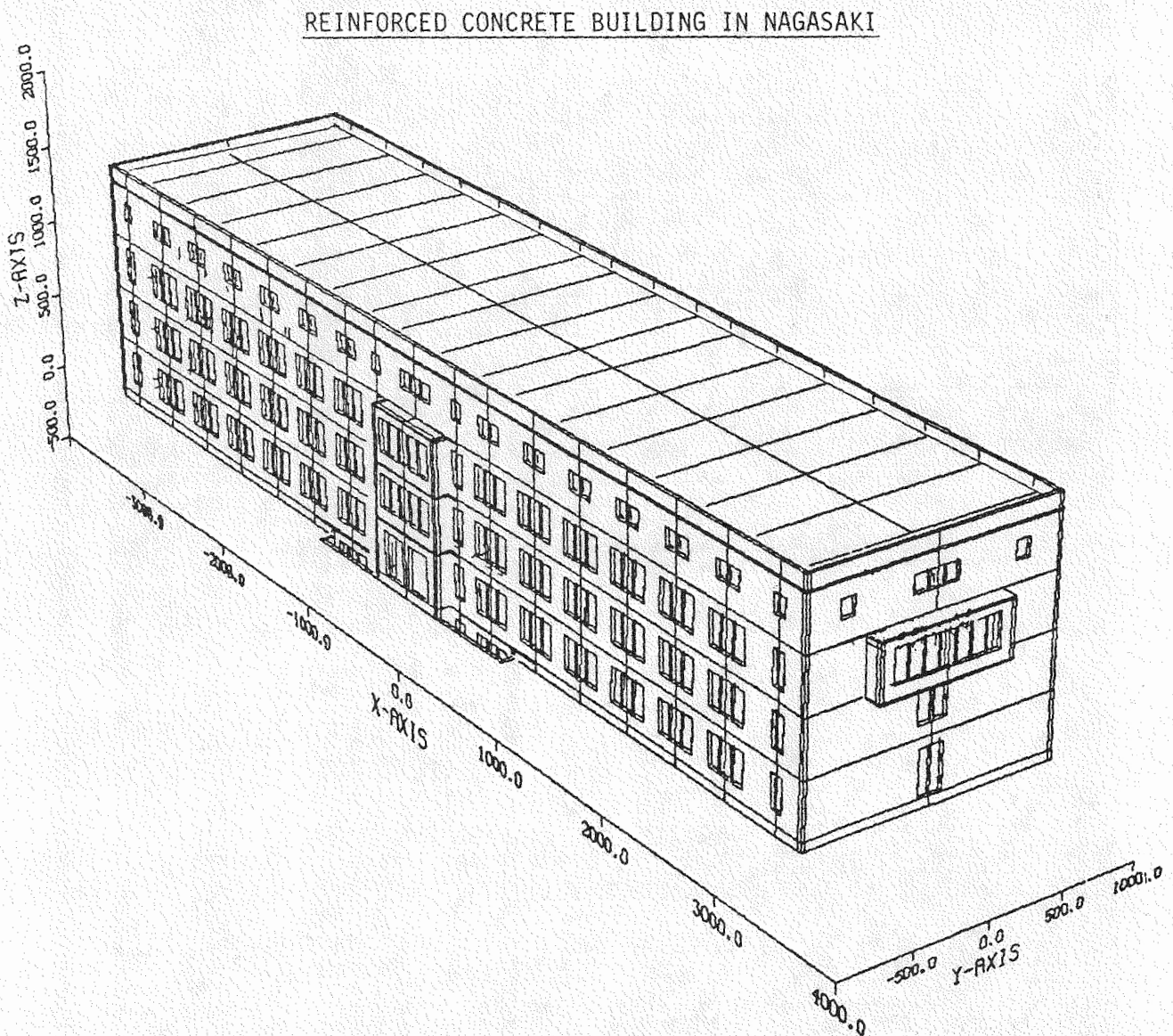

Fig. 13. Computer mock-up of a reinforced concrete building in Nagasaki.

building center on the first floor of approximately 20 rads-neutron ($\pm$8%) and 100 rads-secondary gamma rays ($\pm$12%), the latter being more than 95% from gamma ray creation in the atmosphere, only a few percent creation in the building, and a negligible amount from neutron interactions in the ground. These doses represent reduction factors of 16 and 28, respectively, from those at the same location with no building present. The adjoint leakage files can be retained (see next section) for later coupling due to any changes in the free-field calculation or weapon source spectrum.

III.F.9. Use of a History File

The use of a history file (still sometimes referred to as a collision tape) is neither a variance reduction technique nor a time-saving feature in terms of computing time. But it can be very beneficial in the overall formulation of a deep-penetration calculation. The three codes considered here, MORSE, TRIPOLI, and MCNP (non-standard), all have this capability. Since the random walk and estimation processes are completely independent, it is not necessary for them to be treated in tandem. Instead, during the random walk all pertinent information from each particle history necessary for a desired estimation is recorded on the file, but no estimation is made. Afterward, the history file is read and the estimations are made as if the entire process was just one step. In this manner each process can be modeled and perfected independently of the other. Cross-section and geometry data may also be necessary on the history file. The history file can be analyzed as often as desired for various types of estimators. For next-event estimation all necessary collision and source data must be recorded. In the preceding subsection, if J is from analog boundary crossings, only the particle information at the coupling boundary would be necessary, and it could be combined in independent coupling calculations using ϕ^* from different detector region calculations. Use of a history file also allows recovery and analysis of random walk information due to any abnormal termination of a calculation.

IV. ADJOINT CALCULATIONS

The use of adjoint calculation results is an important aspect of variance reduction for deep-penetration Monte Carlo calculations. The correct interpretation of sometimes even very approximate adjoint information is beneficial when used in the various biasing processes of a forward calculation. Before addressing the realization of this adjoint information, attention will be turned briefly to the topic of complete adjoint calculations.

IV.A. Complete Adjoint Calculations

A formal presentation of adjoint methods is given elsewhere.[4] In theory any Monte Carlo calculation can be solved equally well in the forward or adjoint mode. In all foregoing and following discussions and equations, the role of the forward and adjoint terms can be reversed with no loss in generality. The adjoint flux ϕ^* becomes the quantity of

interest, and forward calculated results determined from other (possibly Monte Carlo) methods can be used for the importance function $I(x)$. Optimal forms of the $I(x)$ for multigroup adjoint calculation biasing have been proposed.[41] The physical (forward) source $S(x)$ and response function $R(x)$ become the adjoint response and source, $R^*(x)$ and $S^*(x)$, respectively. The desired result λ is given by

$$\lambda = \int \phi(x)R(x)dx = \int \phi^*(x)S(x)dx \quad . \tag{42}$$

In practice the forward mode is usually the calculational method of choice. But there are several general problem types, dictated by various physical considerations, for which the calculation can be performed more efficiently in the adjoint mode.[42] A common situation arises when multiple calculations are to be made with different physical source descriptions, and the degree of difficulty in using the adjoint mode is not prohibitive. Then the right side of Eq. (42) can be evaluated as often as needed using $\phi^*(x)$ from only one adjoint solution. This is, of course, the counterpart of the situation for one forward calculation and multiple (physical) response functions.

The solution is often more amenable to the adjoint mode (see Fig. 14) when a deep-penetration calculation has some combination of the following characteristics:

1. a large extended physical source volume or surface;
2. a small concentrated physical detector region;
3. a large region of simple geometry and/or particle transport (one-dimensional or void, for example) adjacent to the source region;
4. a smaller region of complex geometry and/or particle transport adjacent to the detector region.

A large physical source, from which it may be difficult to adequately sample forward source particles, will be a simple adjoint-particle volume or surface detector region. Likewise, a small physical detector region (a point, for example) and all the difficulties it presents for forward estimation processes becomes a simple adjoint source. If the majority of the geometric complexity is concentrated around the adjoint source (a shielded detector), the adjoint source particles provide ample collision and transport sampling. Subsequent tracks through the simpler geometry to the large adjoint detector require no complicated biasing. In contrast, a forward calculation would require strong biasing in the forward source and simple geometric regions to track particles to the regions in and around the forward detector. Although the conditions for problem applicability for the adjoint mode are usually given in spatial terms, similar consideration should be given to any of the other phase space variables if a difficult situation arises in the forward mode.

General continuous energy adjoint codes exist, but they are not widely used, the gamma-ray option being more easily calculated than that for neutrons.[43,44,45] The difficulties associated with the creation and interpretation of adjoint cross sections and subsequent collision mechanics are virtually eliminated in multigroup adjoint Monte Carlo. Here the

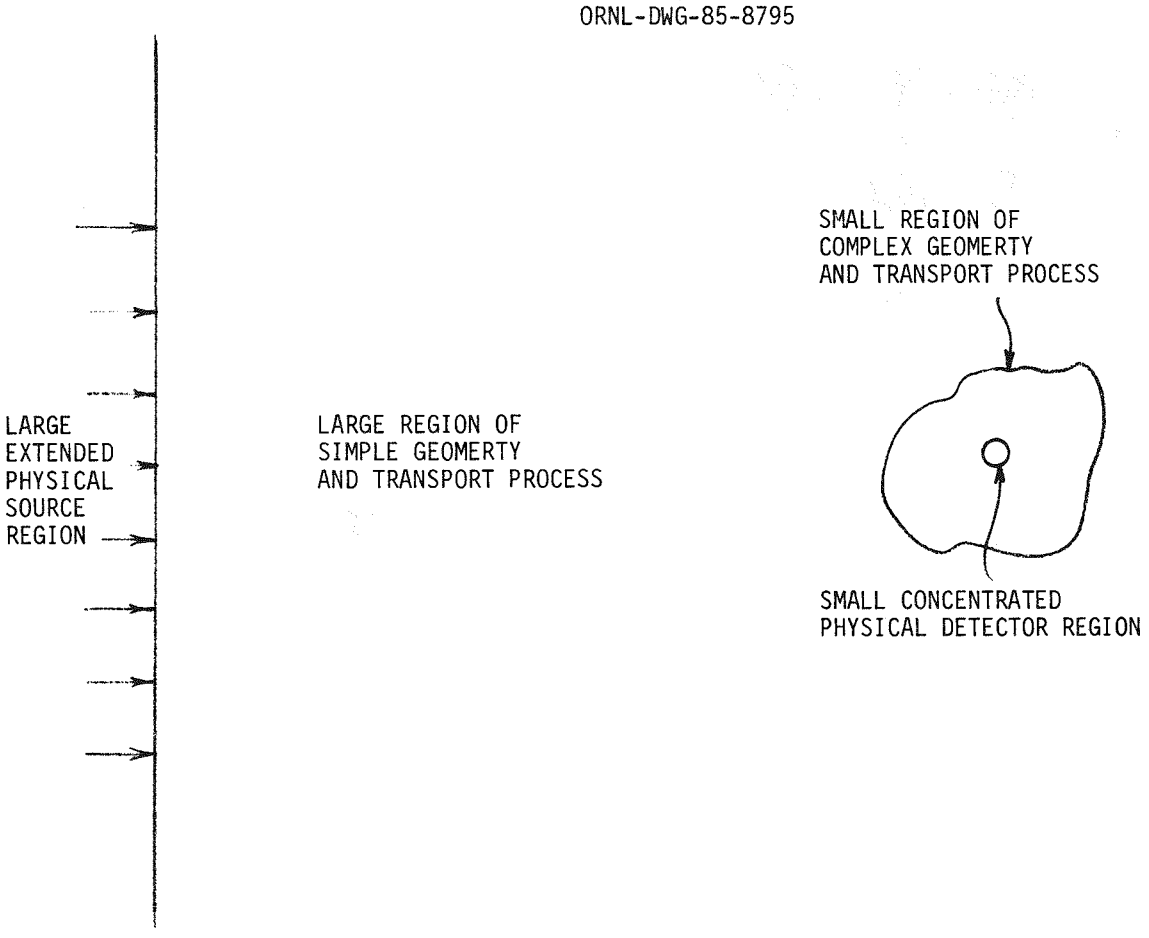

Fig. 14. Geometry suitable for adjoint analysis.

loss in generality due to multigroup theory is no more than that in the forward mode. As a result, multigroup codes are often used for adjoint Monte Carlo. MORSE is a multigroup code with an adjoint option, and MCNP has a multigroup "patch" available.

Even when applicable, practical implementation of the adjoint Monte Carlo method is beset by most of the problems of the forward mode plus several new ones. Deep penetration requires biasing, and methods of empirically setting the importance function (physical insight, intuition, experience, etc.) become more questionable for the adjoint mode. Sampling the adjoint source to simulate the forward detector response and scoring in the adjoint detector region to simulate the forward source distribution are often not straightforward. The same is true for the adjoint source normalization, a usually simple task in the forward mode. Some aspects of these difficulties are discussed elsewhere in connection with a presentation of the MORSE code.[2]

IV.B. Adjoint Importance Generation by Monte Carlo

Most of the techniques for deep-penetration variance reduction discussed in this presentation have been in existence since the early days of Monte Carlo use, and the mechanics for their implementation are firmly embedded in any general code. The methods for creating importance functions, other than empirically, have included analytic approximations,[46] diffusion theory,[47] and one- and two-dimensional discrete ordinates.[25,30] Adjoint Monte Carlo calculated results have been used as importance functions to bias forward calculations, and forward-adjoint Monte Carlo iteration techniques have been reported where the results from one mode are used to bias the other mode in a following calculation, which in turn produces a new set of results for importances in the first mode, etc.[48] These iteration techniques have been developed exclusively for multigroup Monte Carlo. They have also been plagued with long computation times and large uncertainties in the iterated importance function to the extent that convergence in the final results is not guaranteed.

Between the two extremes of empirically set and adjoint generated importances, there are several adaptive, or learning, techniques where biasing parameters are generated and improved upon during a calculation.[49] Additionally, there is a large area of mathematically related variance reduction work summarized in Section V. Some of the simplest of the learning methods are procedures for updating splitting and Russian roulette parameters in the course of a calculation by attempting to equally populate all appropriate regions of phase space.[50,51] Although these methods are in no way optimal, they are usually better than empirically set parameters for large systems, and they can be used as initial values for more exact methods.

In the past decade and continuing at present, there has been considerable effort, and some success, in generating realistic three-dimensional importance function information by forward Monte Carlo methods for use in subsequent forward Monte Carlo calculations. The difficulties associated with running a Monte Carlo code in the adjoint mode and any multigroup restrictions are eliminated.

The result from a transport calculation has the following forms, derived from manipulations of the forward and adjoint integral equations and application of the Gauss divergence theorem:[52]

$$\lambda = \int_{V_R} \phi(x)R(x)dx = \int_{V_S} \phi^*(x)S(x)dx = \int_{A,\bar{n}\cdot\bar{\Omega}>0} \phi(x)\phi^*(x)(\bar{n}\cdot\bar{\Omega})dx \quad , \tag{43}$$

where

V_S is the volume of the source $S(x)$,
V_R is the volume of the detector response $R(x)$,
A is any intermediate surface area enclosing either V_S or V_R,
$(\bar{n}\cdot\bar{\Omega})$ is the cosine of the angle between the particle direction $\bar{\Omega}$ and the normal vector at the surface $\bar{n}$, inward for V_R and outward for V_S (the adjoint $\bar{\Omega}$ is in the opposite sense as that for the the forward $\bar{\Omega}$), and
x is a general phase space variable with only the spatial interval shown explicitly.

In the right-most integral of Eq. (43) the product $\phi(x)\phi^*(x)$, or the "particles" represented by it, is called the "contributon" term after its discrete ordinates application. A Monte Carlo scheme has been devised to evaluate this integral,[53] but it has had little general use, the procedure for evaluating ϕ and ϕ^* being similar in concept to the methods which follow here. This right-hand term can be used to evaluate ϕ^* directly in a forward calculation. The particle weight WT in a Monte Carlo simulation is analogous to the mathematically described current $J(x)$, and at a boundary crossing in a particle history, $WT = J(x) = \phi(x)(\bar{n}\cdot\bar{\Omega})$. The right side of Eq. (43) becomes [see Eq. (41)]

$$\lambda = \int_{A,\bar{n}\cdot\bar{\Omega}>0} \phi(x)\phi^*(x)(\bar{n}\cdot\bar{\Omega})dx = \int_{A,\bar{n}\cdot\bar{\Omega}>0} J(x)\phi^*(x)dx \quad , \tag{44}$$

and the average adjoint flux on surface A is

$$\phi_A^* = \frac{\int_{A,\bar{n}\cdot\bar{\Omega}>0} J(x)\phi^*(x)dx}{\int_{A,\bar{n}\cdot\bar{\Omega}>0} J(x)dx} = \frac{\lambda}{\int_{A,\bar{n}\cdot\bar{\Omega}>0} J(x)dx} \quad . \tag{45}$$

In Eq. (45) the result λ is evaluated by any convenient estimator from particles which reach the detector region [V_R integral in Eq. (43)]. The denominator is the summation of the partial current WT crossing A in the direction going from V_S to V_R. Since all particles which contribute to λ must cross A (or have had predecessors which crossed A), then ϕ_A^* is the expected score, or importance, of particles at A for the response function R in the detector region. The denominator provides the proper normalization for the relative importance at different surfaces. In general, this importance ϕ_A^* will increase on nearing

the detector region since the number of particles, or their weight, will decrease. The converse is true for surfaces near the source region. The ϕ_A^* can be used in later forward calculations as surface importance functions. The source adjoint method in Section III.C.3 and Eq. (12) is a special case of this general method.

In order to extend ϕ_A^* to general phase space dependence, the λ must be scored in terms of the value of the variable that the particle, or its predecessor, had on crossing surface A, not in terms of the variable in the detector region. For energy dependent $\phi_{A,\Delta E}^*$ the energy integral implicit in the denominator of Eq. (45) would be over ΔE only, i.e., the summation of particle weights crossing A is separated into energy intervals for scoring λ. (Recall that x is a general phase variable, not just position.) The numerator becomes $\lambda_{\Delta E}$, and if the particle in the detector region or its predecessor had an energy within ΔE when it crossed surface A, its weight WT is scored in $\lambda_{\Delta E}$ after being multiplied by the response function R evaluated at its energy in the detector region. Otherwise, no score is made in $\lambda_{\Delta E}$ for $\phi_{A,\Delta E}^*$, but in either case no account is given to the energy that the particle has in the detector region in relation to ΔE. The detector region energy and other variables are, of course, used to determine the individual contributions to λ, $\phi(x)R(x)$, in the left side of Eq. (43) [see Eq. (12) and the comments following Eq. (14)]. If no score is made for one ΔE, there may be a score for another ΔE. Any one particle in the detector region may, at the same time, contribute to $\phi_{A,\Delta E}^*$ for several surfaces A for a different ΔE on each surface. In fact, the condition that all particles or their progeny must pass through all surfaces is not necessary. The surfaces may be subdivided giving $\phi_{\Delta A,\Delta E}^*$, and a further test must be made on the scoring particles in the detector region to determine if they or any predecessors passed through a particular surface ΔA. Thus, each scoring particle may be a potential contribution to each $\phi_{\Delta A,\Delta E}^*$, and the storage and retrieval of these data in the course of a calculation can become an enormous task.[54]

This method of importance generation has been extended to surfaces completely enclosing volumes ΔV exterior to the source or detector regions, although either of these two regions may also be a specified volume.[15] Analogous to the surface treatment, the importance in the volume $\phi_{\Delta V}^*$ is the ratio of the total score in the detector region V_R due to particles which have passed (or had predecessors pass) through ΔV on the way to V_R, to the total weight of all particles which have entered ΔV (see Fig. 15). Any particle in V_R will contribute to $\phi_{\Delta V}^*$ for any ΔV that it or its predecessors have passed through. All particles in V_R contribute to $\phi_{V_S}^*$. This method assumes that particles entering ΔV constitute an equivalent source there, and the middle term of Eq. (43) is applicable where the V_S integral is replaced by ΔV for this equivalent source in determining $\phi_{\Delta V}^*$. The comments on scoring, differentiality, and treatment of the data for the surface importance also apply to the volume calculations.

It can be seen that, in general, these surface and volume importance generation schemes are not restricted to one detector response or even to one detector region. However, as the number and differentiality of the ϕ^* to be generated increases, the computation time and complexity must increase proportionally to ensure reasonable statistical uncertainty for all regions of phase space. Use of a history file here (see Section III.F.9) would be appropriate to separate the Monte Carlo calculation from the actual compilation of ϕ^*. Use of either of these two methods, surface or volume, is directly applicable to any general Monte Carlo code with some modification to the estimation process.

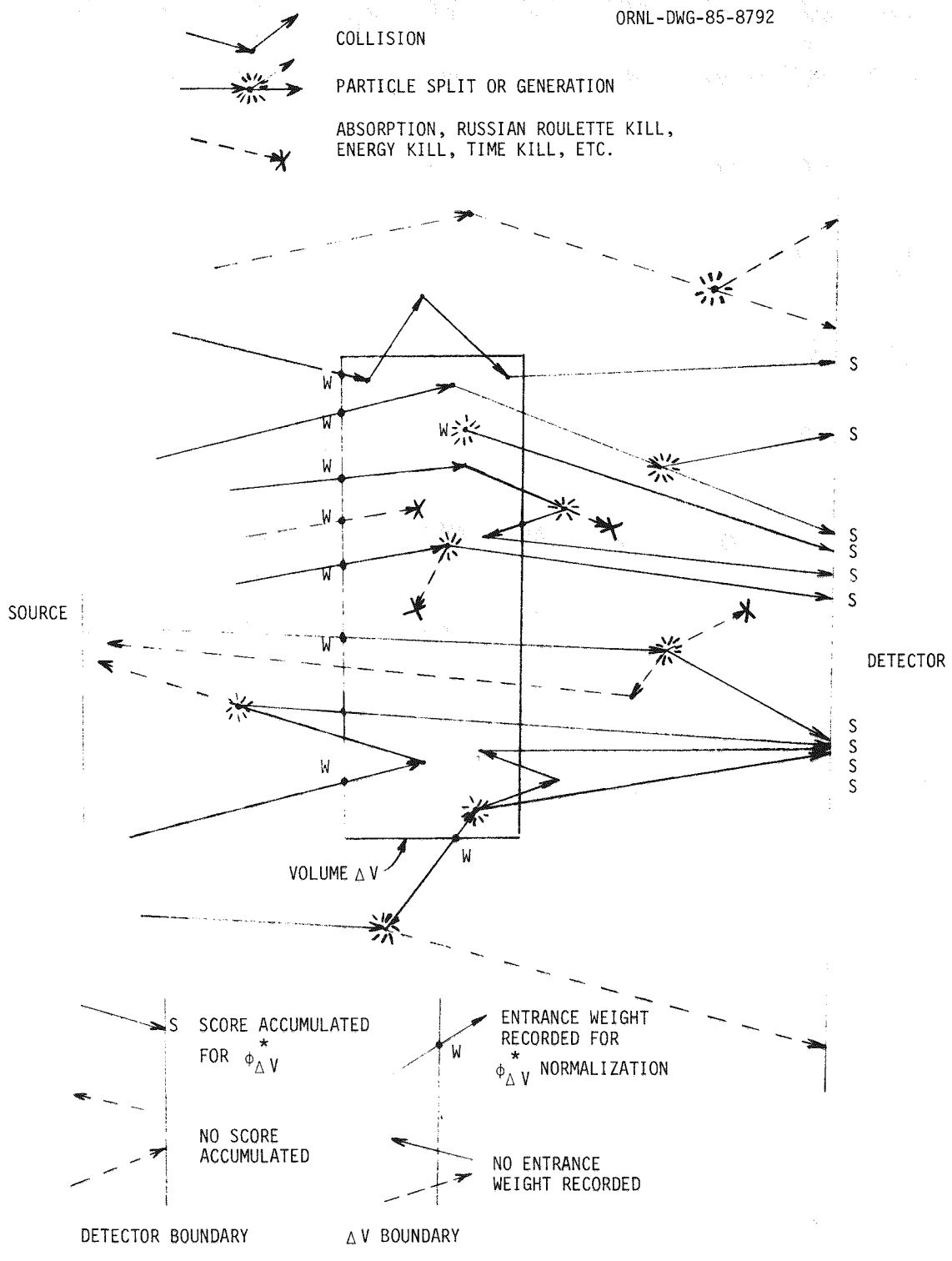

Fig. 15. Contributions to $\phi^*_{\Delta V}$.

The volume generation of ϕ^* has been employed in MCNP for a volume and angular weight-window importance function that was then used in a calculation with DXANG biasing.[31] An empirical method has also been used in MCNP as a means of providing initial values for the ϕ^* generation.[51]

IV.C. Recursive Monte Carlo Generation of Importance

One of the difficulties with the procedures just described is that the most (highest) important regions are usually far from the source near the detector regions. As for any other quantity in a deep-penetration calculation, ϕ^* can be difficult to evaluate. A situation then arises where one needs beforehand a good estimate of the quantity to be calculated, leading to a possible time-consuming iteration of successive Monte Carlo calculations. A method which in principle overcomes this difficulty in the generation of importance functions is called "recursive Monte Carlo." This is not a code learning technique in the sense of the previous methods, and it is not connected with any specific forward calculation source spectrum-response function system. The procedure begins at the detector region and proceeds recursively in backward steps toward the source region. The principal feature is that each step is a forward non-deep-penetration Monte Carlo calculation and only one pass is made through the system from detector to source.

The system is initially divided into volumes by defining surfaces (see Fig. 16), typically one or two mean free paths apart, on which the importance functions will be calculated for subsequent use in a standard calculation of the entire deep-penetration calculation. In the method currently in use, these surfaces are set empirically.[55] This represents a simplification from an earlier, not generally successful attempt to have the surfaces located automatically in an optimal manner by a computer-generated decision process.[56]

The formal mathematical presentation of recursive Monte Carlo is somewhat complicated, but the method can be illustrated by use of selected parts of Eqs. (43) and (44):

$$\int_{V_S} \phi^*(x)S(x)dx = \int_A J(x)\phi_A^*(x)dx \quad . \tag{46}$$

On surface A the importance $\phi_A^*(x)$ is known. For the equivalent source application, volume V_S is defined by surface A and the next surface A' toward the source on which $\phi_{A'}^*(x)$ is to be determined. A spatial delta function source $\delta_V(x)$ is introduced in V_S with a non-zero value only on this next surface A' giving

$$\phi_{A'}^*(x) = \int_{V_S} \phi^*(x)\delta_V(x)dx = \int_A J(x)\phi_A^*(x)dx \quad , \tag{47}$$

where $J(x)$ is now determined from the forward integral transport equation with $S(x) = \delta_V(x)$:

$$J(x) = \delta_V(x) + \int_{V_S} J(x')K(x'\rightarrow x)dx' \quad . \tag{48}$$

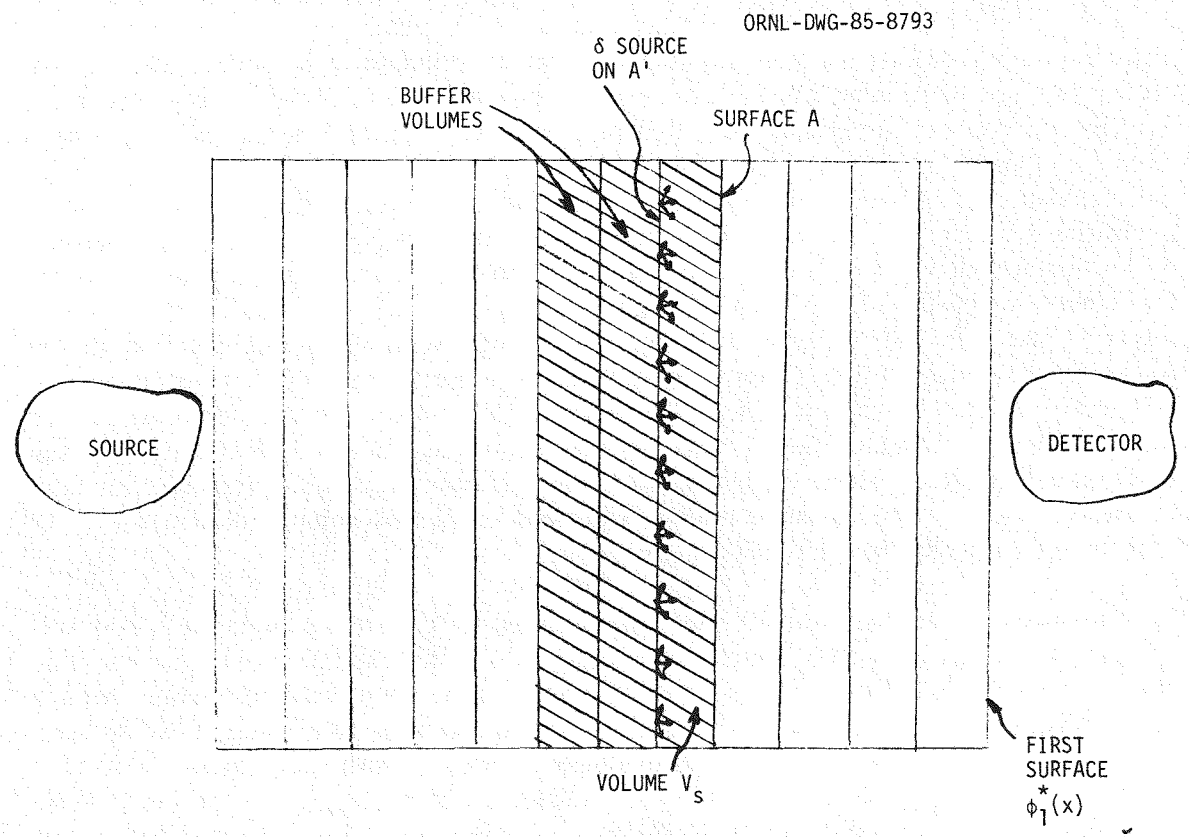

Fig. 16. Recursive Monte Carlo geometry.

A standard forward calculation is begun by uniform selection of source particles on surface A'. In addition to V_S, tracking is also done in one or two "buffer" volumes toward the source to allow particles to re-enter V_S. When a particle crosses surface A, a contribution is made to $\phi_A^*(x)$ according to Eq. (47) since $\phi_A^*(x)$ has been calculated in the previous step. The initial $\phi_1^*(x)$ on the surface nearest the detector must be determined from an actual adjoint calculation from any method (discrete ordinates if possible). The particle history in V_S is terminated when it crosses surface A and scores. This scoring procedure, like the two previous importance generation methods, is based on the particle source variables, not those at the surface crossing. In fact, Eq. (47) is identical to Eq. (45) with a unit denominator. Although importance functions do not generally need any absolute normalization for their use, such a normalization could be determined in the recursive method on the surface surrounding the physical source in the last step of the calculation. As for the other two methods, the recursive method can generate importances for multiple response functions and multiple detector regions with a corresponding increase in the $\phi_A^*(x)$ data at each surface. However, the recursive method treats all detector regions with equal efficiency, whereas the surface and volume methods in the preceding section will need biasing techniques in the calculation to determine the multiple importances, and this biasing cannot in general be optimized for all detector regions and response functions in one calculation. The recursive method will need one initial adjoint solution $\phi_1^*(x)$ for each separate response function and detector region.

Another attractive feature of the recursive method is "geometrical imaging." Any volume anywhere in the system which is identical in size and composition (or mean free paths) to another previously calculated volume needs no random walk calculation through it. A history file from the previous volume is used to determine $J(x)$ in Eq. (48) for scoring with the appropriate $\phi_A^*(x)$. A possible serious problem with the recursive method is the propagation of statistical error from the calculation of one $\phi_A^*(x)$ to the next. Also, empirically setting all the necessary surfaces in a complicated three-dimensional system can become a difficult task. These two items have hindered the application of the method to the extent that its use has never become widespread.

The recursive Monte Carlo method has been incorporated into a special version of the MORSE code.[55] However, through clever use of history files and some programming, this method should be applicable to any general Monte Carlo code. It can be seen that the recursive generation of ϕ^* for use in a forward Monte Carlo calculation could also be effected by discrete ordinates methods if appropriate geometric simplifications are applicable.[57]

IV.D. Application of Adjoint Information

It should be remembered that the importance generation schemes presented here are needed to provide only approximate adjoint fluxes for a forward problem. Exact results, if achieved, would eliminate the need for any further work. The importances can be realized from either approximate fluxes from the exact forward problem or more exact fluxes from an approximation to this problem. The techniques in the preceding sections are currently undergoing development and testing at various institutions, and their use should not be considered to guarantee immediate and ultimate success in any deep-penetration problem.

However, the success of the volume generation of ϕ^* in Section IV.B has progressed to the point where its application may soon become available as a standard option in the MCNP code.

Mention was given in the introduction of this section concerning the interchange of forward and adjoint modes of calculation. The same is true for these importance generation techniques; forward fluxes may be generated by adjoint means for use in biasing later adjoint calculations. But if the intent is to perform one complete and final adjoint calculation, use of the scoring methods just described will give the differential forward flux $\phi(x)$, i.e., reverse the forward and adjoint modes in the two preceding sections. Thus one has the ability, in theory, to calculate $\phi(x)$ and $\phi^*(x)$ and obtain results from both multiple sources and multiple detectors in one calculation of either mode. For multiple sources or multiple detectors covering large portions of phase space, or for too much differentiality, much of the generated data may be unusable due to its statistical uncertainty.

The practical use of general phase space adjoint data in a forward Monte Carlo calculation requires some sort of automated procedures.[15,54] A recent example is that of discrete ordinates adjoint data used in a multigroup Monte Carlo code.[25] Although this is a one-dimensional example, it represents a realistic application of multiple biasing procedures in a spent-fuel shipping cask study (Section III.D.5). Individually these variance reduction items are (g is the energy group):

- Source energy biasing in the source volume ΔS using $\phi^*_{g,\Delta S}$ as $I(x)$ in Eqs. (1) and (2), where $\phi^*_{g,\Delta S}$ is the average discrete ordinates adjoint flux in ΔS for group g.
- Setting the average weight in the weight-window for Russian roulette survival for each spatial importance region ΔV relative to that in the source region

$$WT_{\mathrm{AV}_{\Delta V}} = \frac{S_g}{\hat{S}_g} \; x \; \frac{\phi^*_{g,\Delta S}}{\phi^*_{g,\Delta V}} \quad , \tag{49}$$

  where S_g and $\hat{S}_g$ are the natural and biased source spectra. The particle weight for which splitting is applied in ΔV for group g is set similarly to Eq. (49) except that the average flux $\phi^*_{g,\Delta V}$ is replaced by the minimum spatial flux for group g in ΔV, $\phi^*_{g,\mathrm{MIN}}$. Likewise the weight below which Russian roulette is played is determined using $\phi^*_{g,\mathrm{MAX}}$, the maximum adjoint flux in ΔV for group g. Thus, the weight-window for the entire system is created relative to the source region.
- Secondary particle generation. This study was done in the MORSE "primary particle" mode of calculation where gamma-ray generation is treated automatically in the code as a neutron group-to-gamma-ray group transfer. However, there was some fission neutron creation due to the spent fuel. The weights of the fission neutrons in ΔV were set as a weighted average of $WT_{\mathrm{AV}_{\Delta V}}$, Eq. (49), and the fission spectrum. There is then a correspondence between the weight of generated particles and the average weight of all particles in the region.

- Collision energy biasing in ΔV using $\phi^*_{g,\Delta V}$ as the importance function, the application being similar to that for source biasing.
- Derivation of the exponential transform biasing parameter, p in Eq. (21), from the angular dependent discrete ordinates event value function[30] $W_g(\bar{r},\Omega)$, where, in general notation,

$$W_g(\bar{r},\Omega) = R_g(\bar{r},\bar{\Omega}) + \sum_{g'} \int_{4\pi} \frac{\Sigma_S^{g\to g'(\bar{r},\bar{\Omega}\to\bar{\Omega}')}}{\Sigma_T^g(\bar{r})} \phi^*_{g'}(\bar{r},\bar{\Omega}')d\bar{\Omega}' \quad , \tag{50}$$

where

$R_g(\bar{r}, \Omega)$ = response (zero except in the detector region),
$\phi^*_g(\bar{r},\Omega)$ = adjoint flux.

The transform parameter p was determined from a minimized-error, least square fit of the adjoint data in the most preferential (forward, $\mu=1$) direction. Angular dependence is included from the μ in Eq. (21).

- Application of the next-event estimation probabilities (see Section III.F.5) using $\phi^*_{g,\Delta V}$.

IV.E. Discussion of Adjoint and Optimal Biasing

In the discussion of Section III.D.5 it is pointed out that exponential transform biasing in specific MCNP and MORSE cases was found to be ineffective when adjoint-optimized splitting and Russian roulette (weight window) parameters were used. However, it is well known that prudent use of empirically-set transform and associated input parameters in codes such as TRIPOLI can produce accurate results of high precision. In TRIPOLI the weight window is not used as a primary transport biasing procedure, but only as a weight control device to guard against large fluctuations in the exponential transform weight correction. Experience and insight in setting the parameter, the equal-weight surfaces, the preferential directions, and the weight window minimize the need for splitting and Russian roulette. But it can never be known how good empirically set input parameters are in terms of calculational efficiency as compared to some theoretically optimized procedure.

From the point of view that weight differences in the Monte Carlo scoring process cause an increase in variance, σ^2, it can be argued that Russian roulette increases variance while splitting decreases variance. Conversely, on a per source particle basis, the calculation time, τ, is reduced by Russian roulette and increased by splitting. It is the overall gain in efficiency, $(\sigma^2\tau)^{-1}$, of splitting and Russian roulette that has been found to make use of the weight window beneficial, even necessary, in practice. The use of an optimized weight window for splitting and Russian roulette, such as that given by Eq. (49), can actually be inefficient when used with exponential transformation if this optimization is done without regard to path length biasing. That is, for paths stretched in the preferential directions, the adjusted weights, on average, will fall below WT_{AV}, and in unpreferred directions the adjusted weight will be above WT_{AV}. These effects lead to splitting and Russian roulette of the transformed particle histories as discussed in the second paragraph in Section III.D.5.

It is apparent that a weight window for splitting and Russian roulette can negate the effects of exponential transformation and also collision biasing, especially if the window is relatively narrow. It must be remembered that adjoint-generated weight windows set the optimal particle weight in phase space which will give the smallest variance σ^2 in the ultimate results. But this theoretical concept does not extend to setting the width of the window or to increasing the efficiency by decreasing the running time. In the two examples cited in Section III.D.5 the weight window widths (the ratio of the weight for splitting to that for Russian roulette) are on the order of 5, being set empirically for the MCNP case and semi-empirically for MORSE. For completely empirically set biasing parameters, weight windows widths on the order of several hundred or larger are often used, but the window itself may not even encompass the theoretically optimal value.

In the two cited cases the weight windows were normalized to that in the source region, and as a result any source energy biasing effects were carried throughout the calculation. However, there seems to be no convenient way to include subsequent random walk biasing weight adjustments in the weight window similar to that for the source, Eq. (49). Perhaps a wider window or a directional dependent window would help. But adding angular dependence to a spatial and energy dependent window would greatly increase the difficulty in generating reliable windows, either empirically or by code learning techniques. In the discrete ordinates generated weight window, Eq. (49), the adjoint fluxes and source distributions have been angularly integrated. However, the application of angular dependence here would be straightforward. The angular dependence is that of the polar angle relative to the preferential direction of the exponential transform. The higher $\phi^*_{g,\Delta V,\Delta\mu}$ in the forward directions will produce a lower WT_{AV} more closely corresponding to the stretched-path weight correction, Eq. (27), than the average WT_{AV} in Eq. (49). Likewise, the lower $\phi^*_{g,\Delta V,\Delta\mu}$ in the backward direction will produce a higher WT_{AV} corresponding to the shrunk-path weight correction. The weight window width for splitting and Russian roulette would be set, as before, using the minimum and maximum spatial values of $\phi^*_{g,\bar{r},\Delta\mu}$. However, the determination of the weight window width in any setting, either optimal or empirical, is an item requiring further investigation in the overall concept of calculational efficiency.

Although it seems obvious that the use of an angular dependent weight window would improve the efficiency of optimized exponential transformation, the extension to collision biasing is not so apparent in the multigroup application where the outgoing energy group is selected before the outgoing direction. The standard collision biasing in MORSE is for energy group only, and an angular dependent weight window would not be fully utilized unless one of the non-standard angular biasing schemes was employed.[29,30] However, the general collision angular biasing procedure in Section III.E.1 could easily be included in a multigroup Monte Carlo code and used with an angular dependent weight window.

It is assumed that for each group-to-group energy transfer term in the multigroup code there is an angular term of the form $f(\mu_i,\phi) = f_i/2\pi$, where the f_i are the probabilities associated with discrete scattering (polar) angle cosines μ_i or discrete scattering cosine intervals.[2,58,59] The azimuthal angle ϕ is chosen isotropically. The angular importance function from Section III.E.1 (also see Section III.D.4) is

$$I(\bar{\Omega}) = \frac{\Sigma_T(\bar{r},E)}{\Sigma_T^*(\bar{r},E)} = \frac{\Sigma_T(\bar{r},E)}{\Sigma_T(\bar{r},E) - \chi(E)k(\bar{r})\mu} , \tag{51}$$

where $\mu = \bar{\Omega}\cdot\bar{\Omega}_0$ is the cosine of the angle between $\bar{\Omega}$, the particle direction after collision (the direction to be biased) and the important direction $\bar{\Omega}_0$. In Fig. 17 the incident particle direction $\bar{\Omega}'$ is taken to lie along the vertical axis, and the $\bar{\Omega}_0$ is placed in the $\phi=0,\pi$ plane. The important direction $\bar{\Omega}_0$ will normally correspond to the preferential direction of the exponential transform, Eq. (22), when the transform is applied independently from the constant importance-surface concept of Section III.E.1. With no loss in generality, the following procedures will be developed using the simpler notation of Section III.D.2 for the importance function, i.e., $I(\bar{\Omega}) = (1-p\mu)^{-1}$. The relationship between the p and k parameters, $\Sigma_T p = \chi k$, is explained in Section III.D.4. The normalized, biased marginal distribution function for the μ_i selection is

$$f'(\mu_i) = \frac{2\int_0^{\pi} \frac{f_i d\phi}{2\pi(1-p\mu)}}{\sum_{i=1}^{N} 2\int_0^{\pi} \frac{f_i d\phi}{2\pi(1-p\mu)}} \quad \text{for } i=1,\ldots, N \quad . \tag{52}$$

The addition theorem gives, for $\phi'=0$,

$$\mu = \mu_i\mu_0 + \sqrt{1-\mu_i^2}\ \sqrt{1-\mu_0^2}\ \cos\phi \ . \tag{53}$$

The evaluation of Eq. (52) gives ($0 \leqslant p < 1$)

$$f'(\mu_i) = \frac{\frac{f_i}{\sqrt{(1-p\mu_{i\mathrm{MX}})(1-p\mu_{i\mathrm{MN}})}}}{\sum_{i=1}^{N} \frac{f_i}{\sqrt{(1-p\mu_{i\mathrm{MX}})(1-p\mu_{i\mathrm{MN}})}}} \tag{54}$$

where $\mu_{i\mathrm{MX}}$ is the maximum value of μ ($\phi=0$) for a given μ_i, and $\mu_{i\mathrm{MN}}$ is the minimum value ($\pi=\pi$). When a cosine μ_n has been selected from Eq. (54), the conditional distribution for a biased azimuthal angle selection is

$$f'(\phi)_{\mu_n} = \frac{\frac{1}{2\pi(1-p\mu_n)}}{2\int_0^{\pi} \frac{d\phi}{2\pi(1-p\mu_n)}} \ . \tag{55}$$

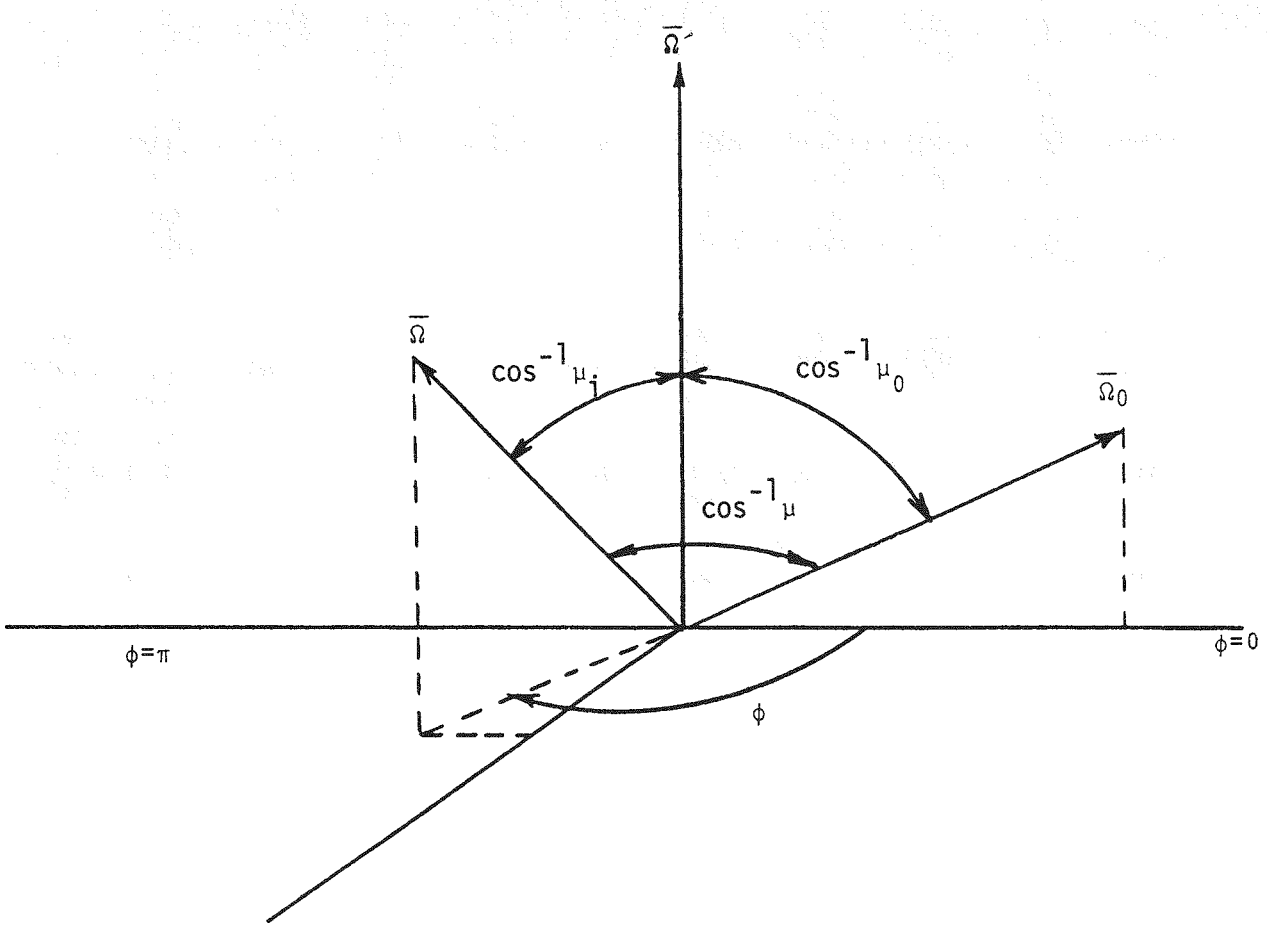

Fig. 17. Angular biasing geometry.

The realization of ϕ is

$$\phi = \pm 2 \tan^{-1} \left\{ \sqrt{\frac{1-p\mu_{n\mathrm{MX}}}{1-p\mu_{n\mathrm{MN}}}} \tan \left(\frac{\pi}{2} \zeta \right) \right\} \tag{56}$$

where ζ is a uniform random number, $0 \leq \zeta \leq 1$. It is seen that for $\zeta = \frac{1}{2}$ the $\tan^{-1}$ argument is <1 (except for $\bar{\Omega} = \bar{\Omega}'$, when it is unity). Then for more than half the random selections $|\phi| < \pi/2$, and the entire direction $\bar{\Omega}$ is biased toward $\bar{\Omega}_0$, not just the scattering angle cosine μ_n.

The final direction $\bar{\Omega}$ can be determined from the three dot product equations defining μ, μ_0, and μ_i in Fig. 17, the three unknowns being the direction cosines of $\bar{\Omega}$. The more conventional approach would be to use the standard cosine transformations (see Ref. 2, page 40) in which the unbiased azimuthal angle selection is isotropic. However, the ϕ from Eq. (56) must be adjusted by a ϕ' since the standard form of Eq. (53) contains a $\cos(\phi - \phi')$ term. This ϕ' is determined from the azimuthal angle in the laboratory system corresponding to μ_0, which must be converted to the system in Fig. 17 using the cosine transformations. That is, if W' and W_0 are the z-directed laboratory system cosines corresponding to $\bar{\Omega}'$ and $\bar{\Omega}_0$ respectively, then

$$\cos\phi' = \frac{\mu_0 W' - W_0}{\sqrt{1-\mu_0^2}\ \sqrt{1-W'^2}} \ . \tag{57}$$

The combined particle weight correction for the two biased angle selections is

$$WT = (1 - p\mu_n) \sum_{i=1}^{N} \frac{f_i}{\sqrt{(1-p\mu_{i\mathrm{MX}})(1-p\mu_{i\mathrm{MN}})}} \ . \tag{58}$$

If this angular biasing is used in conjunction with the exponential transform, the $(1-p\mu_n)$ term in Eq. (58) cancels with a similar transform correction term – see Eq. (27). General application of these procedures would suggest a similar source angular biasing of $S(\mu,\phi)$, where μ is either continuous or discrete and $\bar{\Omega}$ is the initial (biased) particle direction. The cancelation of weight correction terms would apply to the initial transformed flight path selection.

Another biasing approach which might be investigated would be a generalization of that given in Section III.C.2 for the source region. One could separate the non-physical process biasing (splitting and Russian roulette) from the physical processes (collision and transport kernel biasing), with the possible exception of survival biasing, such that all

weight corrections are applied at the time of the estimation process. That is, proceed with collision and exponential biasing in the normal manner, but carry along the multiplicative weight adjustment from event to event independent from the random walk weight. In this manner, splitting and Russian roulette would not undo the physical process biasing. In general, this scheme will be totally unacceptable for empirically set biasing parameters due to large weight variation in the estimation process and resulting large variance. However, if all biasing were done with an optimal or near-optimal importance function (adjoint), perhaps an efficiency increase could be achieved over that for splitting and Russian roulette alone. A discrete ordinates adjoint will provide sufficient differentiality where it is applicable, even giving an optimal exponential transform parameter p. But it seems questionable that, for general three-dimensional systems, the Monte Carlo learning techniques in Sections IV.B, IV.C, and V will be able to provide the complete and precise adjoint function necessary to ensure success in this estimation weighting scheme for general phase space biasing.

Although this section deals with optimal biasing applications, it is beneficial to examine methods of empirically-set "adjoints" as employed in the TRIPOLI code (see Sections III.D.4 and III.D.5). Here, in concept, the transformed equation is solved (the general form of Eqs. (16) and (18), see Section III.D.4); i.e., the weight corrections have been eliminated, and a final normalization is applied to the end result. The collision biasing correction cancels with part of that from the exponential transform, and the remaining exponential term is absorbed into the weight window, as proposed above. In TRIPOLI the weight window is constructed by the code from the input data, not from adjoint generated information.[27] (These concepts are further explained in the short Appendix B of Ref. 1 which, unfortunately, has been inadvertently omitted from the English translation.)

V. MATHEMATICAL ASPECTS OF VARIANCE REDUCTION

In recent years there have been many and various studies in Monte Carlo variance reduction which are of a purely mathematical nature, rather than those presented in previous sections for utilizing biasing techniques and importance functions. Many of these developments are oriented toward deep-penetration of radiation, but their direct application by users of standard codes is not yet wide spread. Among these items are:

- Artificial intelligence and pattern recognition applied to automated adjustment of splitting and Russian roulette procedures and other biasing procedures.[60,61]
- General multivariate approach where a mathematical formulation of variance is subjected to various minimization procedures.[62] An application to source energy biasing uses the method of Lagrangian multipliers.[63] In principle, several biasing parameters can be investigated simultaneously, and periodically throughout a calculation these parameters are updated to those giving the minimum variance to intermediate results. This procedure can be thought of as a perturbation method with the biasing parameters being the perturbation.[49] It has been observed that biasing parameters optimized by this method seem to have no connection to any conventional importance function, but may be related to stratified source sampling.[64]

- A method of variance analysis which is potentially useful for (1) predicting independently and in advance the variance of a contemplated Monte Carlo calculation, (2) optimizing the effects of Monte Carlo variance reducing techniques and estimators, and (3) improving the variance analysis of measured data.[65] The basis of the method is the mathematical construction of the score accumulation probability $\psi(P,s)ds$ defined as the probability that a particle emerging from a collision or source event in phase space P will ultimately accumulate, before its death, a total score of ds about s.[65] For non-analog Monte Carlo applications this probability is also a function of the particle statistical weight w. The probability, now $\psi(P,s,w)ds$, depends on the proposed random walk model as well as the physical model, and it can be very complicated when various biasing processes are included. The ψ is constructed as an infinite, Neumann-like series where the i-th term ($i = 0, 1, 2, \ldots$) represents the probability contribution from the i-th collision. Each individual term is a multiplicative set of problem-dependent integral quantities.

This complete ψ probability, once constructed, is used to define a series of equations based on the mathematical moments of the ultimate score s.

$$M_n(P,w) = \int_{-\infty}^{\infty} s^n \, \psi(P,s,w)ds \qquad n=0,1,2,... \tag{59}$$

Due to the form of ψ, the M_n appear in the above integral as $M_n(P')$,$M_n(P'')$, etc., and Eq. (59) is actually a set of transport-like integral equations. The M_0 is unity and the M_1 equation is an adjoint equation defining the ultimate expected total score of a particle emerging from P. The solution of the M_1 and M_2 equations permits the construction of the variance (by any means, analytic if possible) $\sigma^2 = [M_2 - M_1^2]N^{-1}$ of a proposed Monte Carlo calculation of N histories. In principle, the variance of the variance can be studied by solving for the first four M_n of Eq. (59). However, the simultaneous solution of these equations would be very difficult even for very simple proposed problems. The variance σ^2 has been investigated for many aspects of Monte Carlo processes and estimators. A partial list includes (1) track-length estimators[66], (2) survival biasing and expectation estimators,[67] (3) splitting, survival biasing, exponential transformation, and next-event estimation,[68] (4) time-dependent multiplying systems,[69] (5) computation cost of splitting and Russian roulette,[70] (6) geometric splitting,[71] (7) zero variance biasing schemes,[72] and (8) contributon Monte Carlo.[73]

VI. INTERPRETATION OF CODE-GENERATED STATISTICAL UNCERTAINTY

In most Monte Carlo codes the results are accompanied by an estimate of the statistical uncertainty in the form of absolute, fractional, or percent standard deviation or relative error, i.e., some form of the square root of the variance. In earlier sections the concept of

accuracy versus precision has been discussed, and it will be summarized here along with general comments on Monte Carlo statistical uncertainty. There is the maxim that multiplying system calculations will always have enough particles available to calculate an accurate result within some precision (statistical uncertainty). The problem here is that an uncertainty of a few *tenths* of a percent is often required. In contrast, for deep penetration an accuracy of a few hundred percent with an uncertainty of a few *tens* of percent may be acceptable. Often even in fairly well formulated deep-penetration calculations the primary goal of accuracy is not met, leading to low results due to undersampling in important regions of phase space. Poorly or improperly applied biasing techniques may only decrease the calculated uncertainty about the erroneous result without increasing the accuracy. Fortunately, the few situations which can give results much larger than the truth (e.g., collisions near point estimators or surface estimator grazing angles) also give very large uncertainties.

Most of the problems associated with accuracy and precision in any Monte Carlo calculation would exist even if the calculated uncertainties were theoretically correct. However, it is generally accepted that many techniques, such as exponential transformation and next-event estimation, have distributions which are not Gaussian.[24,39,49] But the methods used to interpret the uncertainties (the central limit theorem) are based on a Gaussian distribution. Even the casual users of Monte Carlo methods are aware that a change in a deep-penetration biasing parameter or only in the random number sequence may produce results which differ by more than that predicted by the calculated uncertainties. Attempts to improve on the method of code-calculated uncertainties have not been generally successful,[49] and this situation remains a serious theoretical problem.

In Sections IV and V there are presented several adaptive or learning techniques in which a code is able to automatically update various biasing parameters in the course of a calculation. Except in the case of completely excluding this learning phase of a calculation in the compilation of final results, there is at present no generally acceptable criteria for utilizing these preliminary results and their statistical errors.[49] This situation is similar to k_{eff} calculations where each generation is dependent on information calculated in the previous generations.

Even in the case of actual Gaussian statistics, the calculated uncertainty in the form of one standard deviation of the mean result (σ, the square root of the variance) is often misinterpreted in determining the reliability of an answer. There is only a 68.3% probability that the Monte Carlo answer lies within $\pm\sigma$ of the "true answer." One must go to $\pm 2\sigma$ (95.4%) or even to $\pm 3\sigma$ (99.7%) to establish more credibility. If another calculation is made with any change, other than just a different random number sequence, the fact that there is an overlap or not of the two results plus or minus their respective uncertainties cannot be given too great a significance, especially if the change and/or the uncertainties are large. The theoretical percents given above apply only to a given distribution, and any change such as in a biasing parameter constitutes a different distribution. In practice it is, of course, always gratifying if two such results do overlap.

It is helpful to examine a calculated result and its uncertainty for any anomalies as they are accumulated throughout a calculation rather than accept a final answer after some arbitrary number of histories. This is a standard procedure in k_{eff} calculations, which have their own unique uncertainty problems.[64] Some general-purpose codes, such as MCNP, provide these intermediate results as a standard feature. Although this type of analysis can be useful in the ultimate acceptance or rejection of a calculation, it should never be used as a means to retain or reject certain combinations of batch results until some desired answer is attained. It is a common practice in the use of codes with batch statistics to exclude a batch if it creates some unusual contribution to the result, e.g., a collision close to a point detector. It is a much better procedure to guard against such occurrences (Section III.F.3 for point detectors) in the programming algorithms than to try to objectively exclude certain contributions to a final result.

Many codes, such as TRIPOLI and MORSE, use batches for statistical analysis. It has been observed for distributions which are not Gaussian that the batch results will usually be approximately Gaussian distributed, and the code generated uncertainties can be properly interpreted according to the percentages given above.[26] However, it is also known that particle statistics (one batch) give a more reliable uncertainty in the sense that the variance of the variance (the fourth moment of the result) is smaller than for multiple batches.[74] The average result from a calculation will, of course, be unaffected by the statistical analysis method, but the estimated variance will in general be different for different batching arrangements. There are no guidelines for what combinations of particles and batches will give a minimum variance, and any such post-calculation analysis would seem to be an artificial means of variance reduction. Typically, several tens of batches each of several hundreds of particles are necessary before confidence can be established in the results and uncertainties. Codes designed to use batches should not be run with one particle per batch in order to gain the theoritical minimum fourth moment feature. The pre- and post-batch computation time for code bookkeeping and other tasks can easily exceed the random walk calculation time for one particle, increasing the overall computer times significantly over that for normal batching of particles.

The MCNP code produces relative error estimates based on particle statistics. The intermediate results mentioned above are useful for making judgments on the distribution and convergence of the results. If the relative error is approximately proportional to the inverse of the square root of the number of histories, then it can be assumed that the scoring distributions are well behaved. MCNP also prints out the efficiency (figure of merit), $1/\sigma^2\tau$, with the intermediate results. Since the compilation time τ is proportional to the number of histories, this value should remain roughly constant throughout a calculation.

In practice, one often relies on experience as much as theoretical considerations in making decisions about a calculated result and its uncertainty. The somewhat unscientific practice of making other calculations with different random number sequences is always beneficial but often impractical, and sometimes misleading, as with point detectors and exponential transformation. For a new type of calculation, previous experience can also be misleading. The following empirical criteria for evaluation of deep-penetration uncertainties (given as a percent of the mean result) should be used for general guidance only.

$\sigma > 50\%$	Results are unreliable
$20\% < \sigma < 50\%$	Results may be off by up to a factor of 5
$10\% < \sigma < 20\%$	Results may be within 100% of the truth
$\sigma < 10\%$	Generally reliable results
$\sigma < 5\%$	Generally reliable results for a point detector in a non-void medium

VII. CONCLUSIONS

Any general study of deep-penetration Monte Carlo methods, such as that presented here, is replete with "waffle" words and phrases such as "usually," "could be," "in most cases," etc. Illustrations of a biasing technique are "often" given with simplifying assumptions in the other processes such as monoenergetic, isotropic emissions, one-dimensional, etc. As mentioned in the introductory section, there is no general recipe for success in deep-penetration calculations, and there is no substitute for experience regardless of theoretical considerations. All source, biasing, and estimation techniques should be tested singly and in combination before a long calculation is made. Short, preliminary calculations should be scrutinized for particle population and weight control throughout the important regions of phase space. It is best to begin with simple biasing procedures and proceed from there rather than initially combine several complicated biasing techniques. The MCNP literature[3,5] gives many examples and discussions of realistic applications of biasing procedures and gives guidelines for judging the variance of results.

In the discussions of the efficiency criterion in Monte Carlo (the inverse of the product of variance and calculation time), no mention was made of the personnel and computer time for all the pre-calculation analysis necessary to achieve a certain $(\sigma^2\tau)^{-1}$ in a long calculation. There is, of course, no way to include these intangible effects in a general study of Monte Carlo methods, but they may be of principal concern in relation to program funding levels and deadlines.

Three general Monte Carlo codes, MCNP, TRIPOLI, and MORSE, have been referenced here with respect to various aspects of variance reduction. Many general codes exist, but these three have been promoted for use in the public domain with a minimum of governmental, commercial, or installation-dependent red tape. An attempt to objectively compare these codes reveals that TRIPOLI has the most sophisticated variance reduction methods available by input, e.g., the completely independent importance geometry description and general exponential transform and collision biasing options. As a result, TRIPOLI is the most difficult of the three codes for a new user who attemps to fully utilize its capabilities. The use of MCNP, a state-of-the-art code in every aspect, has traditionally been oriented toward simplicity in variance reduction, i.e., boundary splitting and Russian roulette rather than the exponential transform for particle population and weight control. However, recent application of a general weight-window and automatic generation and use of adjoint importance has changed this situation somewhat. There are also several options in MCNP, such as DXTRAN, for specific variance reduction application. MORSE is a

multigroup energy code, with a corresponding loss in generality in relation to the other two codes. However, it is this multigroup structure (not a serious restriction if proper attention is given to the cross section weighting) that has given the MORSE system of codes much of its flexibility, adaptability, and installation independence. In actual practice, the decision to use a particular code is often based on such considerations as its availability on a convenient computer operating system, availability of cross section and other necessary data, and the close proximity of personnel with expertise in its use.

In considering future work in Monte Carlo methods in general and variance reduction in deep-penetration calculations in particular, it must be pointed out that there are now several three-dimensional discrete ordinates codes in existence. While this in no way spells out the demise of the Monte Carlo method, due to its more exact treatment of the physical processes and generality in geometry, its traditional bastion as the only three-dimensional transport method has been irrevocably breached. In speculating on future work, it is of interest to consider present capabilities in terms of those envisioned by the pioneers of Monte Carlo methods — large, high-speed computers; complete evaluated cross-section libraries; general geometry capability; etc. Current efforts in Monte Carlo-oriented machine software development are directed toward placing as much of the pre-analysis burden as possible on the computer.[11] Then, not too far in the future a Monte Carlo calculation sequence may have the following scenario aided by an "expert system."

- application of sensitivity analysis in pesudo-point and multigroup cross-section processing;
- a variance minimization analysis (Section V) of various possible estimators;
- a recursive Monte Carlo or discrete ordinates calculation (Section IV.C) for preliminary setting of importance parameters;
- preliminary calculations, as for criticality source convergence, where the initial importances are adjusted if necessary by the forward-calculated adjoints in Section IV.B.;
- final setting of the system importance function followed by the complete calculation.

VIII. REFERENCES

The references included here are intended only to represent the subject matter and are not meant to be a complete bibliography. In many cases a recent reference is cited from which earlier references can be traced. Thus, no claims are made to completeness, originality, or chronological order. The bibliographies in reference 75 form a comprehensive listing of Monte Carlo work up to the time of its publication. Since that time the archival journals, such as *Nuclear Science and Engineering,* are the best sources for Monte Carlo development. Proceedings of conferences such as the 1983 International Shielding Conference in Tokyo[33] are good sources for applications.

REFERENCES

1. J. C. Nimal, et al., "Programme de Monte Carlo Polycinétique à Trois Dimensions TRIPOLI," CEA.N.1919, Saclay (1976); also RSIC Computer Code Package CC-272, Oak Ridge National Laboratory (1980).
2. S. N. Cramer, *Applications Guide to the MORSE-CG Radiation Transport Code,* ORNL/TM-9355, Oak Ridge National Laboratory (1985); also M. B. Emmett, *The MORSE Monte Carlo Radiation Transport Code System,* ORNL-4972, Oak Ridge National Laboratory (1975).
3. S. N. Cramer, *Applications Guide to the RSIC Distributed Version of the MCNP Code,* ORNL/TM-9641, Oak Ridge National Laboratory (1985); also *MCNP – A Generalized Monte Carlo Code for Neutron and Photon Transport,* LA-7396-M, Los Alamos National Laboratory (1981).
4. D. C. Irving, *The Adjoint Baltzmann Equation and Its Simulation by Monte Carlo,* ORNL/TM-2870, Oak Ridge National Laboratory (1970).
5. W. L. Thompson *et al.,* "Deep Penetration Calculations," *Proc. of the Theory and Applications of Monte Carlo Methods,* ORNL/RSIC-44, April 21-23, 1980.
6. S. N. Cramer and R. W. Roussin, "Experience with TRIPOLI at ORNL," *Proc. of the Theory and Applications of Monte Carlo Methods,* ORNL/RSIC-44, April 21-23, 1980.
7. R. E. MacFarlane *et al., The NJOY Nuclear Data Processing System: User's Manual,* LA-9303-M, Los Alamos (1982), also ENDF-324.
8. G. Dejonghe *et al.*, "THEMIS-4: A Coherent Punctual and Multigroup Cross Section Library for Monte Carlo and S_N Codes from ENDF/B-IV," Sixth International Conference on Radiation Shielding, Tokyo (1983).
9. N. M. Greene *et al., AMPX, A Modular Code System for Generating Coupled Multigroup Neutron-Gamma Ray Libraries from ENDF/B,* ORNL/TM-3706 (1976).
10. H. Rief, et al., "Review of Monte Carlo Techniques for Analyzing Reactor Perturbations," *Proc. International Meeting on Advances in Nuclear Engineering Computation,* Knoxville, Tennessee, April 9-11, 1985.
11. S. N. Cramer *et al.*, "Monte Carlo Techniques for Solving Deep-Penetration Problems," *Proc. International Meeting on Advances in Nuclear Engineering Computational Methods,* Knoxville, Tennessee, April 9-11, 1985.
12. M. B. Emmett, *DOMINO-II, A General Purpose Code for Coupling DOT-IV Discrete Ordinates and Monte Carlo Radiation Transport Calculations,* ORNL/TM-7771 (1981).
13. T. J. Hoffman *et al., Nucl. Sci. Eng.* **48**, 179-188 (1972).
14. K. Ueki, *Nucl. Sci. Eng.* **79**, 253-264 (1981).
15. T. E. Booth and J. S. Hendricks, *Nucl. Tech./Fusion* **5**, 90-100 (1984).

16. H. Rief and A. Fioretti, "Monte Carlo Shielding Analysis Using Deep Penetration Biasing Schemes Combined with Point Estimators and Algorithms for the Scoring of Sensitivity Profiles and Finite Perturbation Effects," *Sixth International Conference on Radiation Shielding,* Tokyo, Japan (1983).

17. M. H. Kalos *et al.*, "Monte Carlo Methods in Reactor Computations," in *Computing Methods in Reactor Physics*, H. Greenspan *et al.*, Eds., Gordon and Breach Publishers, New York (1968).

18. F. H. Clark, *The Exponential Transform As an Importance Sample Device,* ORNL/RSIC-14 (1966).

19. J. M. Lanore, *Nucl. Sci. Eng.* **45**, 66-72 (1971).

20. M. H. Kalos, *Theory and Application of Monte Carlo Methods,* panel summary, ORNL/RSIC-44 (1980).

21. S. N. Cramer, *Nucl. Sci. Eng.* **65**, 237-253 (1978).

22. H. Lichtenstein *et al., The SAM-CE Monte Carlo System for Radiation Transport and Criticality Calculations in Complex Configurations (Revision 7.0),* MAGI report MR-705207, also EPRI CCM-8 (1980).

23. R. C. Gast, *Proc. of the NEACRP Meeting of a Monte Carlo Study Group,* ANL-75-2, NEACRP-L-118, p. 50 (1974).

24. R. C. Bending, "Direction-Dependent Exponential Biasing," *Proc. NEACRP Meeting of a Monte Carlo Study Group,* ANL-75-2, NEACRP-L-118, Argonne National Laboratory (1975).

25. T. J. Hoffman and J. S. Tang, *XSDRNPM-S Biasing of MORSE-SGC/S Shipping Cask Calculations,* ORNL/CSD/TM-175, NUREG/CR-2342 (1982).

26. S. Goertzel and M. H. Kalos, "Monte Carlo Methods in Transport Problems," in *Progress in Nuclear Energy, Physics, and Mathematics*, Series I, Vol. 2, pp. 315-369, Pergamon Press, Inc., New York, 1958.

27. T. Vergnaud *et al.*, "Biasing Techniques in the TRIPOLI-2 System," *Proc. International Meeting on Advances in Nuclear Engineering Computational Methods,* Knoxville, Tennessee, April 9-11, 1985.

28. J. S. Hendricks and L. L. Carter, "Anisotropic Angle Biasing of Photons," *Nucl. Sci. Eng.* **89**, 118-130 (1985).

29. C. E. Burgart and P. N. Stevens, *Nucl. Sci. Eng.* **42**, 306 (1970).

30. J. S. Tang *et al., Nucl. Sci. Eng.* **62**, 617-626 (1977), Nucl. Sci. Eng. **64**, 837-842 (1977), ORNL/TM-5414 (1976).

31. T. E. Booth, "A Weight Window Importance Generator for Monte Carlo Streaming Problems," *Sixth International Conference on Radiation Shielding,* Tokyo, Japan (1983).

32. W. E. Selph, *Neutron and Gamma Ray Albedos,* ORNL/RSIC-21 (1968).

33. *Proceedings of the Sixth International Conference on Radiation Shielding,* Japan Atomic Energy Research Institute, Tokyo, Japan, May 16-20, 1983.

34. *Neutron Streaming Calculations with MORSE/BREESE for the ORNL-TSF Two-Legged Duct and CRBR Prototypic Coolant Pipe Chaseway Experiments,* Science Applications Report, SAI 1-138-02-034-00 (1978).

35. J. K. Thompson *et al., SNAKE: A Solid Angle Calculational System,* U.S. Nuclear Regulatory Commission, NUREG/CR-0004 (1978).

36. S. N. Cramer, *Nucl. Sci. Eng.* **79**, 417-425 (1981).

37. T. J. Hoffman and J. S. Tang, *Trans. Am. Nucl. Soc.* **46**, 654-655 (1984).

38. W. Matthes and H. Rief, *Ann. Nucl. Ener.* **6**, 305-307 (1979).

39. A. Dubi *et al., Ann. Nucl. Ener.* **9**, 675-682 (1982).

40. L. S. Abbott and F. R. Mynatt, *Review of ORNL Radiation Shielding Analyses of the FFTF Reactor,* ORNL-5027 (1975).

41. P. N. Stevens, *Nucl. Tech./Fusion* **5**, *109-114 (1984).*

42. M. H. Kalos, A Review of the Monte Carlo Method for Radiation Transport Calculations, ORNL/RSIC-29 (1971).

43. M. O. Cohen, *ANTE 2 -- A FORTRAN Computer Code for the Solution of the Adjoint Neutron Transport Equation by Monte Carlo,* MAGI Corp., DASA-2396 (1970).

44. L. L. Carter, *MCNA: A Computer Program to Solve the Adjoint Neutron Transport Equation,* LA-4488, Los Alamos (1971).

45. J. E. Hoogenboom, *Nucl. Sci. Eng.* **79**, 357-373 (1981).

46. T. W. Armstrong and P. N. Stevens, *J. Nucl. Energy* **23**, 331-359 (1969).

47. E. Shuttleworth and S. J. Chucas, "Linked Monte Carlo and Finite-Element Diffusion Methods for Reactor Shield Design," *Sixth International Conference on Radiation Shielding,* Tokyo, Japan (1983).

48. K. Ueki and P. N. Stevens, *J. Nucl. Sci. Tech.* **16**, 117-131 (1979).

49. E. M. Gelbard, "Unfinished Monte Carlo Business," *Proc. ANS/ENS Topical Meeting on Advances in Mathematical Methods for the Solution of Nuclear Engineering Problems,* Munchen, West Germany, April 27-29, 1981.

50. M. J. Grimstone, "Monte Carlo Methods for Shield Design Calculations," *Proc. NEACRP Meeting of a Monte Carlo Study Group,* ANL-75-2, NEACRP-L-118, Argonne National Laboratory (1975).

51. J. S. Hendricks, *Trans. Am. Nucl. Soc.* **41**, 307 (1982).

52. M. L. Williams and W. W. Engle, Jr., "Spatial Channel Theory — A Technique for Determining the Directional Flow of Radiation Through Reactor Systems," *Fifth International Conference on Reactor Shielding,* Knoxville, TN (1977).

53. A. Dubi and S. Gerstl, *Nucl. Sci. Eng.* **76**, 198-217 (1980).

54. O. L. Deutsch, "Simultaneous Global Calculation of Flux and Importance with Forward Monte Carlo," *Fifth International Conference on Reactor Shielding,* Knoxville, TN (1977).

55. M. Goldstein and E. Greenspan, *Nucl. Sci. Eng.* **76**, 308-322 (1980).

56. M. H. Kalos *et al.*, "Automatic Computation of Importance Sampling Functions for Monte Carlo Transport Codes," MAGI report MR-7017, DNA 2890F (1972).

57. M. Goldstein *et al., Trans. Am. Nucl. Soc.* **46**, 410-412 (1984).

58. L. L. Carter and C. A. Forest, *Nucl. Sci. Eng.* **59**, 27-45 (1976).

59. N. De Gangi and F. Munno, *MOSEC-SP: Incorporation of a Step Function Angular Distribution Option in the MORSE Cross Section Module,* RSIC report PSR-142, Oak Ridge National Laboratory (1979).

60. J. I. Macdonald and E. D. Cashwell, *Application of Artificial Intelligence Techniques to the Acceleration of Monte Carlo Transport Calculations,* LA-7475-M, Los Alamos (1978).

61. T. J. Hoffman, "An Application of Pattern Recognition Techniques to Monte Carlo Radiation Transport Problems," *Fifth International Conference on Pattern Recognition,* Miami (1980).

62. J. Spanier, "New Developments in Reactor Mathematics and Applications," CONF-710302, Idaho Falls (1971), also *SIAM J. Appl. Math.* **18**, 172-190 (1970).

63. T. J. Hoffman, *Nucl. Sci. Eng.* **69**, 76-77 (1979).

64. E. M. Gelbard, "Monte Carlo Small-Sample Perturbation Calculations," *Proc. ANS Topical Meeting on Advances in Reactor Computations, Salt Lake City,* March 28-31, 1983.

65. T. E. Booth and H. J. Amster, *Nucl. Sci. Eng.* **65**, 273-281 (1978).

66. H. Armster and M. J. Djomehri, *Nucl. Sci. Eng.* **60**, 131-142 (1976).

67. I. Lux, *Nucl. Sci. Eng.* **67**, 317-325 (1978).

68. P. K. Sarkar and M. A. Prasad, *Nucl. Sci. Eng.* **70**, 243-261 (1979).

69. T. E. Booth and E. D. Cashwell, *Nucl. Sci. Eng.* **71**, 128 (1979).

70. R. J. Juzaitis, *Nucl. Sci. Eng.* **80**, 424 (1982).

71. A. Dubi and D. J. Elperin, *Nucl. Sci. Eng.* **80**, 139-161 (1982).

72. S. R. Dwivedi, *Nucl. Sci. Eng.* **80**, 172-178 (1982).

73. P. K. Sarkar and M. A. Prasad, *Nucl. Sci. Eng.* **87**, 136-151 (1984).

74. A. Dubi, *Nucl. Sci. Eng.* **72**, 108-110 (1979).

75. L. L. Carter and E. D. Cashwell, "Particle-Transport Simulation with the Monte Carlo Method," U.S. E.R.D.A., TID-26607 (1975).

ORNL/TM-9643

INTERNAL DISTRIBUTION

1-5. L. S. Abbott
6. R. G. Alsmiller
7. D. G. Cacuci
8-12. S. N. Cramer
13. H. L. Dodds
14. M. B. Emmett
15. T. A. Gabriel
16. D. T. Ingersoll
17. J. O. Johnson
18. R. A. Lillie
19-23. F. C. Maienschein
24. J. V. Pace
25. C. V. Parks
26. L. M. Petrie
27. W. A. Rhoades
28. R. W. Roussin
29. J. C. Ryman
30. R. T. Santoro
31. C. O. Slater
32. J. S. Tang
33. D. K. Trubey
34. M. W. Waddell
35. C. R. Weisbin
36. R. M. Westfall
37. A. Zucker
38. P. W. Dickson, Jr. (Consultant)
39. G. H. Golub (Consultant)
40. R. M. Haralick (Consultant)
41. D. Steiner (Consultant)
42. Central Research Library
43. Y-12 Document Ref. Section
44. Laboratory Records ORNL, RC
45. ORNL Patent Office
46-50. EPMD Reports Office

EXTERNAL DISTRIBUTION

51. Office of the Assistant Manager for Energy Research and Development, DOE-ORO, Oak Ridge, TN 37830, Attention: S. W. Ahrends/M. Rohr
52-78. Technical Information Center
79-140. DNA Radiation Transport Distribution