LA-7396-M, Rev. 2
Manual

UC-32
Issued: September 1986

MCNP—A General Monte Carlo Code for Neutron and Photon Transport

Version 3A

Judith F. Briesmeister, Editor 7277

CONTRIBUTORS

T. E. Booth
J. F. Briesmeister
D. G. Collins
J. J. Devaney
G. P. Estes
H. M. Fisher
R. A. Forster
T. N. K. Godfrey
J. S. Hendricks
H. G. Hughes
R. C. Little
R. E. Prael
R. G. Schrandt
R. E. Seamon
E. C. Snow
W. L. Thompson
W. T. Urban
J. T. West

Los Alamos
Los Alamos National Laboratory
Los Alamos, New Mexico 87545

FOREWORD

This manual is a practical guide for the use of our general-purpose Monte Carlo code MCNP. The first chapter is a primer for the novice user. The second chapter describes the mathematics, data, physics, and Monte Carlo simulation found in MCNP. This discussion is not meant to be exhaustive - details of the particular techniques and of the Monte Carlo method itself will have to be found elsewhere. The third chapter shows the user how to prepare input for the code. The fourth chapter contains several examples, and the fifth chapter explains the output. The appendices show how to use MCNP on particular computer systems at the Los Alamos National Laboratory and also give details about some of the code internals that those who wish to modify the code may find useful.

The Monte Carlo method emerged from work done at Los Alamos during World War II. The invention is generally attributed to Fermi, von Neumann, and Ulam; Metropolis and Richtmyer should be credited also with early development. Much of the early work is summarized in the first book to appear on Monte Carlo by Cashwell and Everett in 1957. Shortly thereafter the first Monte Carlo neutron transport code MCS was written, followed in 1967 by MCN. The photon codes MCG and MCP were then added and in 1973 MCN and MCG were merged to form MCNG. The above work culminated in Version 1 of MCNP in 1977. The first two large user manuals were published by W. L. Thompson in 1979 and 1981. This manual draws heavily from its predecessors.

Neither the code nor the manual is static. The code is changed as the need arises and the manual is changed to reflect the latest version of the code. This particular manual refers to Version 3A.

MCNP and this manual are the product of the combined effort of the people in the Monte Carlo Section of the Radiation Transport Group (X-6) of the Applied Theoretical Physics Division (X Division) at the Los Alamos National Laboratory and reflect approximately 300 person years of effort. We would like to acknowledge the assistance of Michael Keeve, a 1985 summer student, Joe Mack and Alexandra Heath, Los Alamos National Laboratory personnel outside Group X-6, and Lynn Byers, editor on assignment to X Division.

The code and manual can be obtained from the Radiation Shielding Information Center (RSIC), P. O. Box X, Oak Ridge, TN, 37831.

J. F. Briesmeister
Editor
505-667-7277
843-7277 (FTS)

DISCLAIMER OF CERTIFICATION

MCNP is used throughout the world on many projects, including ones carried out by organizations outside of Los Alamos National Laboratory. No document exists that certifies the code for specific problems and there is no intention of ever producing such a document. It is correct results that develop a user's confidence in a code and qualifies the code for various purposes.

The history of MCNP is discussed in Chapter 2, Sec. A. In the 40 years since World War II, the Monte Carlo techniques and data now in MCNP represent over 300 person years of effort and have been used to calculate many tens of thousands of practical problems. MCNP is used over 500 times and 100 Cray hours per month at Los Alamos alone. The probability of a serious error in the code or its data bases is greatly reduced by this heavy usage. There is a large user clientele for MCNP, including the Magnetic Fusion Energy community. MCNP has been used to analyze physics experiments and reactor designs; design nuclear safeguards nondestructive assay systems, radiation shields, and nuclear instrumentation; calculate material activations and magnetic fusion neutronics; perform criticality analyses; and address health-physics problems. It is also used for well logging calculations, determinations of radiological doses, physics experiments, spacecraft radiation modeling, radiation damage studies, and Monte Carlo applications at major universities. The satisfaction of these many users lends credibility to the results of calculations obtained using MCNP.

One of the strongest features of MCNP is its use of continuous-energy data. In reproducing the intentions of cross-section evaluators, MCNP stands second to none in the faithful utilization of even the most sophisticated ENDF/B evaluations. One can model the geometry and particle transport in difficult problems nearly exactly. The possibilities of representing complicated geometries in MCNP are limited only by the dedication of the user. The rich collection of variance reduction techniques allows MCNP to be used very efficiently and effectively for a wide spectrum of applications.

We know of no other Monte Carlo code with better state-of-the-art physics, data, and mathematical methods than MCNP. Furthermore, these state-of-the-art attributes rest on a very strong, time-proven foundation.

CONTENTS

MCNP - A General Monte Carlo Code
for Neutron and Photon Transport
Version 3A

Monte Carlo Section
Los Alamos Radiation Transport Group

ABSTRACT

MCNP is a general-purpose Monte Carlo code that can be used for neutron, photon, or coupled neutron/photon transport, including the capability to calculate eigenvalues for critical systems. The code treats an arbitrary three-dimensional configuration of materials in geometric cells bounded by first- and second-degree surfaces and some special fourth-degree surfaces (elliptical tori).

Pointwise cross-section data are used. For neutrons, all reactions given in a particular cross-section evaluation (such as ENDF/B-V) are accounted for. Thermal neutrons are described by both the free gas and $S(\alpha,\beta)$ models. For photons, the code takes account of incoherent and coherent scattering, the possibility of fluorescent emission after photoelectric absorption, and absorption in pair production with local emission of annihilation radiation.

Important standard features that make MCNP very versatile and easy to use include a powerful general source; a geometry plotter; a very rich collection of variance reduction techniques; an elaborate tally structure and user interface that allow a user to easily calculate almost anything conceivable; and an extensive collection of cross-section data.

CHAPTER 1
PRIMER

This chapter, which will enable the novice to start using MCNP, assumes very little knowledge of the Monte Carlo method and no experience with MCNP. Other available teaching aids are videotapes[1] and a related report[2] that explain many aspects of MCNP, from the basic level to more-advanced concepts.

MCNP is a general-purpose, continuous-energy, generalized-geometry, time-dependent, coupled neutron/photon Monte Carlo transport code. It solves neutral particle transport problems and may be used in any of three modes: neutron transport only, photon transport only, or combined neutron/photon transport, where the photons are produced by neutron interactions. The neutron energy regime is from 10^{-11} MeV to 20 MeV, and the photon energy regime is from 1 keV to 100 MeV. The capability to calculate k_{eff} eigenvalues for fissile systems is also a standard feature.

The user creates an input file that is subsequently read by MCNP. This file contains information about the problem in areas such as: the geometry specification, description of the materials, which cross-section evaluations to use, the location and characteristics of the neutron or photon source, the type of answers or tallies desired, and any variance reduction techniques used to make the problem run more efficiently. These areas will be discussed in the primer by use of a simple problem.

The primer begins with short discussions of the Monte Carlo method. Five features of MCNP are introduced: nuclear data and reactions, source specifications, tallies and output, estimation of errors, and variance reduction. The third section explains MCNP geometry setup, including the concept of cells and surfaces. A general description of an input deck is followed by a sample problem and a detailed description of the input cards used in the sample problem. Section V tells how to run MCNP at Los Alamos and the last section lists tips for setting up correct problems and running them efficiently.

Remember five "rules" when running a Monte Carlo calculation. They will be more meaningful as you read this manual and gain experience with MCNP, but no matter how sophisticated a user you may become, never forget the following five points:

1. Sample the source well;
2. You cannot recover lost information;
3. Question the stability and reliability of results;
4. Be conservative and cautious with biasing; and
5. The number of histories run is not indicative of the quality of the answer.

CHAPTER 1
Introduction

I. PARTICLE TRANSPORT CALCULATIONS WITH MONTE CARLO

The following sections compare Monte Carlo and deterministic methods and provide a simple description of the Monte Carlo method.

A. *Monte Carlo Method vs Deterministic Method*

Monte Carlo methods are very different from deterministic transport methods. Deterministic methods, the most common of which is the discrete ordinates method, solve the transport equation for the average particle behavior. By contrast, Monte Carlo does not solve an explicit equation, but rather obtains answers by simulating individual particles and recording some aspects (tallies) of their average behavior. The average behavior of particles in the physical system is then inferred (using the central limit theorem) from the average behavior of the simulated particles. Not only are Monte Carlo and deterministic methods very different ways of solving a problem, even what constitutes a solution is different. Deterministic methods typically give fairly complete information (for example, flux) throughout the phase space of the problem, whereas Monte Carlo supplies information only about specific tallies requested by the user.

When Monte Carlo and discrete ordinates methods are compared it is often said that Monte Carlo solves the integral transport equation, whereas discrete ordinates solves the integro-differential transport equation. Two things are misleading about this statement. First, the integral and integro-differential transport equations are two different forms of the same equation; if one is solved, the other is solved. Second, Monte Carlo "solves" a transport problem by simulating particle histories rather than by solving an equation. No transport equation need ever be written to solve a transport problem by Monte Carlo. Nonetheless, one can derive an equation that describes the probability density of particles in phase space; this equation turns out to be the same as the integral transport equation.

Without deriving the integral transport equation, it is instructive to investigate why the discrete ordinates method is associated with the integro-differential equation and Monte Carlo with the integral equation. The discrete ordinates method visualizes the phase space to be divided into many small boxes, and the particles move from one box to another. In the limit as the boxes get progressively smaller, particles moving from box to box take a differential amount of time to move a differential distance in space. In the limit this approaches the integro-differential transport equation, which has derivatives in space and time. By contrast, Monte Carlo transports particles between events (for example, collisions) that are separated in space and time. Thus differential space or time or both do not appear in Monte Carlo transport. The integral equation does not have time or space derivatives.

B. The Monte Carlo Method

Monte Carlo can be used to theoretically duplicate a statistical process (such as the interaction of nuclear particles with materials) and is particularly useful for complex problems that cannot be modeled by computer codes of deterministic methods. The individual probabilistic events that comprise a process are simulated sequentially, and the probability distributions governing these events are statistically sampled to describe the total phenomenon. In general, the simulation is performed on a digital computer because the number of trials necessary to adequately describe the phenomenon is usually quite large. The statistical sampling process is based on the selection of random numbers - analogous to throwing dice in a gambling casino - hence the name "Monte Carlo." In particle transport, the major application for Group X-6, the Monte Carlo technique is pre-eminently realistic (a theoretical experiment). It consists of actually following each of many particles from a source throughout its life to its death in some terminal category (absorption, escape, etc.). Probability distributions are randomly sampled from the transport data to determine the outcome at each step of its life.

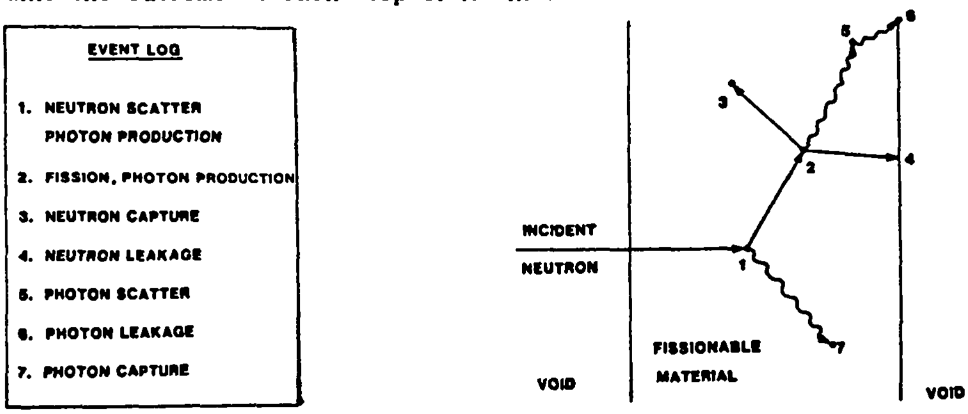


Figure 1.1

Figure 1.1 represents the random history of a neutron incident on a slab of material that can undergo fission. Numbers between 0 and 1 are selected randomly to determine what (if any) and where interaction takes place, based on the rules (physics) and probabilities (transport data) governing the processes and materials involved. In this particular example, a neutron collision occurs at event 1. The neutron is scattered in the direction shown, which is selected randomly from the physical scattering distribution. A photon is also produced and is temporarily stored, or banked, for later analysis. At event 2, fission occurs, resulting in the termination of the incoming neutron and the birth of two outgoing neutrons and one photon. One neutron and the photon are banked for later analysis. The first fission neutron is captured at event 3 and terminated. The banked neutron is now retrieved and, by random sampling,

leaks out of the slab at event 4. The fission-produced photon has a collision at event 5 and leaks out at event 6. The remaining photon generated at event 1 is now followed with a capture at event 7. Note that MCNP retrieves banked particles such that the last particle stored in the bank is the first particle taken out.

This neutron history is now complete. As more and more such histories are followed, the neutron and photon distributions become better known. The quantities of interest (whatever the user decides) are tallied, along with estimates of the statistical precision (uncertainty) of the results.

II. INTRODUCTION TO MCNP FEATURES

Various features, concepts, and capabilities of MCNP are summarized in this section. More detail concerning each topic is available in later chapters or appendices. Attention is drawn to reference 32 also.

A. Nuclear Data and Reactions

MCNP uses continuous-energy nuclear data libraries. The primary sources of nuclear data are evaluations from the Evaluated Nuclear Data File (ENDF)[3] system, the Evaluated Nuclear Data Library (ENDL)[4] and the Activation Library (ACTL)[5] compilations from Livermore, and evaluations from the Applied Nuclear Science (T-2) Group[6-8] at Los Alamos. Evaluated data are processed into a format appropriate for MCNP by codes such as NJOY.[9] The processed nuclear data libraries retain as much detail from the original evaluations as is feasible.

Nuclear data tables exist for neutron interaction, photon interaction, neutron dosimetry or activation, and thermal particle scattering $S(\alpha,\beta)$. It is understood that photon data are atomic rather than nuclear in nature. Each data table available to MCNP is listed on a directory file, XSDIR. Users may select specific data tables through unique identifiers for each table, called ZAIDs. These identifiers generally contain the nuclear charge Z and mass number A.

Over 500 neutron interaction tables are available for approximately 100 different isotopes or elements. Multiple tables for a single isotope are provided primarily because data have been derived from several evaluated sources, but also because of different temperature regimes and different processing tolerances. More neutron interaction tables are constantly being added as new and revised evaluations become available. Photon production data are given as part of the neutron interaction tables when such data are included in the evaluations.

Photon interaction tables exist for all elements from Z=1 through Z=94. The data in the photon interaction tables allow MCNP to account for coherent and incoherent scattering, photoelectric absorption with the possibility of fluorescent emission, and pair production. Scattering

angular distributions may be modified by atomic form factors and incoherent scattering functions.

Cross sections for nearly 2000 dosimetry or activation reactions involving over 400 target nuclei in ground and excited states are part of the MCNP data package. These cross sections may be used as energy-dependent response functions in MCNP to determine reaction rates but may not be used as transport cross sections.

Thermal data tables are appropriate for use with the $S(\alpha,\beta)$ scattering treatment in MCNP. The data include chemical binding and crystalline effects that become important as the neutron's energy becomes sufficiently low. Data are available for light and heavy water, beryllium metal, beryllium oxide, benzene, graphite, polyethylene, and zirconium and hydrogen in zirconium hydride.

B. Source Specification

MCNP's generalized user-input source capability allows the user to specify a wide variety of source conditions without having to make code modification. Independent probability distributions may be specified for the source variables of energy, time, position and direction, and for other parameters such as starting cell(s) or surface(s). Information about the geometrical extent of the source can also be given. In addition, source variables may depend on other source variables (for example, energy as a function of angle) thus extending the built-in source capabilities of the code. The user can bias all input distributions.

In addition to input probability distributions for source variables, certain built-in functions are available. These include various analytic functions for fission and fusion energy spectra such as Watt, Maxwellian and Gaussian spectra; Gaussian for time; and isotropic, cosine, and monodirectional for direction. Biasing may also be accomplished by special built-in functions.

C. Tallies and Output

The user can instruct MCNP to make various tallies related to particle current and particle flux. All standard MCNP tallies are normalized to be per starting particle. Currents can be tallied as a function of direction across any set of surfaces, surface segments, or sum of surfaces in the problem. Fluxes across any set of surfaces, surface segments, sum of surfaces, and in cells, cell segments, or sum of cells are also available. Similarly, the fluxes at designated detectors (points or rings) are standard tallies. Heating and fission tallies give the energy deposition in specified cells. In addition, particles may be flagged when they cross specified surfaces or enter designated cells, and the contributions of these flagged particles to the tallies are listed separately. Tallies such as the number of fissions, the number of absorptions, the total helium

production, or any product of the flux times the approximately 100 standard ENDF reactions plus several nonstandard ones may be calculated with any of the MCNP tallies. In fact, any quantity of the form

$$C \int \varphi(E)\ f(E)\ dE$$

may be tallied, where $\varphi(E)$ is the energy-dependent fluence, and $f(E)$ is any product or summation of the quantities in the cross-section libraries or a response function provided by the user. The tallies may also be reduced by line-of-sight attenuation. Tallies may be made for segments of cells and surfaces without having to build the desired segments into the actual problem geometry. All tallies are functions of time and energy as specified by the user and are normalized to be per starting particle.

In addition to the tally information, the output file contains standard summary information in ledger tables to give the user a better idea of how the problem ran. This information can give insight into the physics of the problem and the adequacy of the Monte Carlo simulation. If errors occur during the running of a problem, detailed diagnostic prints for debugging are given. Printed with each tally is also its relative error corresponding to one standard deviation from the corresponding average. Tally fluctuation charts, which are described in the following section, are also automatically printed to show how a tally mean and error fluctuate as a function of the number of histories run.

A postprocessing code, MCPLOT, is available at Los Alamos to graphically display the MCNP tally output.

D. Estimation of Monte Carlo Errors

All standard MCNP tallies are normalized to be per starting particle and are printed in the output accompanied by a second number R, which is the estimated relative error defined to be one estimated standard deviation of the mean $S_{\bar{x}}$ divided by the estimated mean $\bar{x}$. In MCNP, the quantities required for this error estimate – the tally and its second moment – are computed after each complete Monte Carlo history, which accounts for the fact that the various contributions to a tally from the same history are correlated. For a well-behaved tally, R will be proportional to $1/\sqrt{N}$, where N is the number of histories. Thus, to halve R, we must increase the total number of histories fourfold. For a poorly behaved tally, R may increase as the number of histories increases.

The estimated relative error may be used to form confidence intervals about the estimated mean to make a statement about what the true result is. The Central Limit Theorem states that as N approaches infinity there is a 68% chance that the true result will be in the range $\bar{x}(1\pm R)$ and a 95% chance in the range $\bar{x}(1\pm 2R)$. *It is extremely important to note that these*

confidence statements refer only to the precision of the Monte Carlo calculation itself and not to the accuracy of the result compared to the true physical value. A statement regarding accuracy requires a detailed analysis of the uncertainties in the physical data, modeling, sampling techniques and approximations, etc., used in a calculation.

The guidelines for interpreting the quality of the confidence interval for various values of R are listed in Table 1.1.

Table 1.1
Guidelines for Interpreting the Relative Error R*

Range of R	Quality of the Tally
0.5 to 1	Not meaningful
0.2 to 0.5	Factor of a few
0.1 to 0.2	Questionable
< 0.10	Generally reliable
< 0.05	Generally reliable for point detectors

*$R = S_{\bar{x}}/\bar{x}$ and represents the estimated relative error at the 1σ level. These interpretations of R assume that all portions of the problem phase space are being sampled well by the Monte Carlo process.

For all but point detector tallies, the quantity R should be less than 0.10 to produce generally reliable confidence intervals. Point detector results tend to have larger third and fourth moments of the individual tally distributions, so a smaller value of R, < 0.05, is required to produce generally reliable confidence intervals. The estimated uncertainty in the Monte Carlo result must be presented with the tally so that all are aware of the estimated precision of the results.

Keep in mind the footnote to Table 1.1. For example, if an important but highly unlikely particle path in phase space has not been sampled in a problem, the Monte Carlo results will not have the correct expected values and the confidence interval statements may not be correct. The user can guard against this occurring by setting up the problem to not exclude completely any regions of phase space and by trying to sample all regions of the problem adequately.

Despite one's best effort, an important path may not be sampled often enough, causing confidence interval statements to be incorrect. To try to inform the user about this behavior, MCNP calculates a figure of merit (FOM) for one tally bin of each tally as a function of the number of histories and prints the results in the tally fluctuation charts at the end of the output. The FOM is defined as

$$\mathrm{FOM} \equiv 1/(R^2T),$$

where T is the computer time in minutes. The more efficient a Monte Carlo

calculation is, the larger the FOM will be because less computer time is required to reach a given value of R.

The FOM should be approximately constant as N increases because R^2 is proportional to 1\N and T is proportional to N. *Always examine the tally fluctuation charts to be sure that the tally appears well behaved, as evidenced by a fairly constant FOM.* A sharp decrease in the FOM indicates that a seldom-sampled particle path has significantly affected the tally result and relative error estimate. In this case, the confidence intervals may not be correct the fraction of the time that statistical theory would indicate. Examine the problem to determine what path is causing the large scores and try to redefine the problem to sample that path much more frequently.

E. Variance Reduction

As noted in the previous section, R (the estimated relative error) is proportional to $1/\sqrt{N}$, where N is the number of histories. For a given MCNP run the computer time T consumed is proportional to N. Thus $R = C/\sqrt{T}$, where C is a positive constant. There are two ways to reduce R: (1) increase T and (2) decrease C. Computer budgets often limit the utility of the first approach. For example, if it has taken 2 hours to obtain R=0.10, then 200 hours will be required to obtain R=0.01. For this reason MCNP has special variance reduction techniques for decreasing C. (Variance is the square of the standard deviation.) The constant C depends on the tally choice and/or the sampling choices.

1. Tally Choice

As an example of the tally choice, note that the fluence in a cell can be estimated either by a collision estimate or a track length estimate. The collision estimate is obtained by tallying $1/\Sigma_t$ (Σ_t=macroscopic total cross section) at each collision in the cell and the track length estimate is obtained by tallying the distance the particle moves while inside the cell. Note that as Σ_t gets very small, very few particles collide but give enormous tallies when they do, a high variance situation (see page 112). In contrast, the track length estimate gets a tally from every particle that enters the cell. For this reason MCNP has track length tallies as standard tallies, whereas the collision tally is not standard in MCNP.

2. Nonanalog Monte Carlo

Explaining how sampling affects C requires understanding of nonanalog Monte Carlo models.

The simplest Monte Carlo model for particle transport problems uses the natural probabilities that various events occur (for example, collision, fission, capture, etc.). Particles are followed from event to

event by a computer, and the next event is always sampled (using the random number generator) from a number of possible next events according to the natural event probabilities. This is called the analog Monte Carlo model because it is directly analogous to the naturally occurring transport.

The analog Monte Carlo model works well when a significant fraction of the particles contribute to the estimate and can be compared to detecting a significant fraction of the particles in the physical situation. There are many cases for which the fraction of particles detected is very small, less than 10^{-6}. For these problems analog Monte Carlo fails because few, if any, of the particles tally, and the statistical uncertainty in the answer is unacceptable.

Although the analog Monte Carlo model is probably the simplest conceptual probability model, there are other probability models for particle transport that will estimate the same average value as the analog Monte Carlo models, and they are useful because although the average value remains unchanged, the variance (uncertainty) of the estimate can often be made much smaller than the variance for the analog estimate. Practically, this means that problems that would be impossible to solve in days of computer time can be solved in minutes of computer time.

A nonanalog Monte Carlo model attempts to follow "interesting" particles more often than "uninteresting" ones. An "interesting" particle is one that contributes a large amount to the quantity (or quantities) that needs to be estimated. There are many nonanalog techniques, and they all are meant to increase the odds that a particle scores (contributes). To ensure that the average score is the same in the nonanalog game as in the analog game, the score is modified to remove the effect of biasing (changing) the natural odds. Thus, if a particle is artificially made q times as likely to execute a given random walk, then the particle's score is weighted by (multiplied by) $1/q$. The average score is thus preserved because the average score is the sum, over all random walks, of the probability of a random walk multiplied by the score resulting from that random walk.

A nonanalog Monte Carlo technique will have the same expected tallies as an analog technique if the expected weight executing any given random walk is preserved. For example, a particle can be split into two identical pieces and the tallies of each piece are weighted by 1/2 of what the tallies would have been without the split. Such nonanalog, or variance reduction, techniques can often decrease the relative error by sampling naturally rare events with an unnaturally high frequency and weighting the tallies appropriately.

3. Variance Reduction Tools in MCNP

Standard optional variance reduction schemes in MCNP include geometry splitting/Russian roulette, weight cutoff, time and energy cutoff, correlated sampling, analog capture or implicit capture by weight

reduction, an energy- and cell-dependent weight window to control particle weight fluctuations, the exponential transformation, energy splitting/roulette, forced collisions in designated cells, flux estimates at point or ring detectors, track length estimators, and source biasing (both in energy and direction).

DXTRAN, which stands for deterministic transport, improves sampling in the vicinity of detectors or other tallies. It involves deterministically transporting particles on collision to some arbitrary, user-defined sphere in the neighborhood of a tally and then calculating contributions to the tally from these particles. Contributions to the detectors or to the DXTRAN spheres can be controlled as a function of geometric cell or as a function of the relative magnitude of the contribution to the detector or DXTRAN sphere.

Variance reduction techniques, used correctly, can be a great help to the user. Used poorly, they can result in a wrong answer with good statistics and no clue that anything is amiss. Some variance reduction methods have general application and are not easily misused. Others are more specialized and attempts to use them carry high risk. The use of weight windows tends to be more powerful than the use of importances but typically requires more input data and more insight into the problem. The exponential transform for thick shields is not recommended for the inexperienced user; rather, use many cells with increasing importances (or decreasing weight windows) through the shield. Forced collisions are used to increase the frequency of random walk collisions within optically thin cells but should be used only by an experienced user. The point detector estimator should be used with caution, as should DXTRAN.

For many problems, variance reduction is not just a way to speed up the problem but is absolutely necessary to get any answer at all. Deep penetration problems and pipe detector problems, for example, will run too slowly by factors of trillions without adequate variance reduction. Consequently, users have to become skilled in using the variance reduction techniques in MCNP.

The following summarizes briefly the main MCNP variance reduction techniques. Detailed discussion is in Chapter 2 section VII, page 116.

1. *Importance sampling:* Particles transported from a region of higher importance to a region of lower importance (where they will probably contribute little to the desired problem result) undergo Russian roulette; that is, some of those particles will be killed a certain fraction of the time, but survivors will be counted more by increasing their weight the remaining fraction of the time. In this way, unimportant particles are followed less often, yet the problem solution remains undistorted. On the other hand, if a particle is transported to a region of higher importance (where it will likely contribute to the desired problem result), it may be split into two or more particles (or tracks), each with less weight and therefore counting less. In this way, important particles are followed

more often, yet the solution is undistorted because on average total weight is conserved.

2. ***Weight cutoff/Russian roulette***: If a particle weight becomes so low that the particle becomes insignificant, it undergoes Russian roulette. That is, some particles are killed, and some particles survive with increased weight. Again the solution is not distorted because total weight is conserved, but computer time is not wasted on insignificant particles.

3. ***Time and energy cutoff***: Particles that have a time or energy or both out of the range of interest are killed so that computation time is not wasted following them.

4. ***Implicit capture vs analog capture***: When a particle collides with a nucleus, there is a probability that it is captured by the nucleus. In analog capture, the particle is killed with that probability. In implicit capture, also known as survival biasing, the particle is never killed by capture; instead, its weight is reduced by the capture probability at each collision. Thus, important particles are permitted to survive by not being lost to capture. On the other hand, if particles are no longer considered useful after undergoing a few collisions, analog capture efficiently gets rid of them.

5. ***Exponential transformation***: To transport particles long distances, the distance between collisions in the preferred direction is artificially increased and the weight is correspondingly artifically decreased. Because this often results in large weight fluctuations, it is highly recommended the weight window be used with the exponential transform.

6. ***Forced collisions***: A particle may be forced to undergo a collision each time it enters a designated cell that is almost transparent to the particle by appropriately splitting the particle and its weight into a collided and uncollided part.

7. ***Energy splitting/Russian roulette***: Particles can be split or rouletted upon entering various user-supplied energy ranges. Thus important energy ranges can be sampled more frequently by splitting the weight among several particles and less important energy ranges can be sampled less frequently by rouletting particles.

8. ***Correlated sampling***: The sequence of random numbers in the Monte Carlo process is chosen so that statistical fluctuations in the problem solution will not mask small variations in that solution resulting from slight changes in the problem specification. The i^{th} history will always start at the same point in the random number sequence no matter what the previous $i-1$ particles did in their random walks.

9. ***Source biasing***: Source particles with phase space variables of more importance are emitted with a higher frequency but with a compensating lower weight than are less important source particles.

10. ***Point detectors***: When the user wishes to tally a flux-related quantity at a point in space, the probability of transporting a particle precisely to that point is vanishingly small. Therefore, pseudoparticles are directed to the point instead. Every time a particle history is born

in the source or undergoes a collision, the user may require that a pseudoparticle be tallied at a specified point in space. In this way, many pseudoparticles of low weight reach the detector, which is the point of interest, even though no particle histories could ever reach the detector. For problems with rotational symmetry, the point may be represented by a ring to enhance the efficiency of the calculation.

11. ***DXTRAN***: The DXTRAN method is a way of obtaining large numbers of particles on a user-specified "DXTRAN sphere." DXTRAN makes it possible to obtain many particles in a small region of interest that would otherwise be difficult to sample. Upon sampling a collision or source exit density, DXTRAN estimates the correct weight fraction that should scatter toward, and arrive without collision at, the sphere. The DXTRAN method then puts this correct weight on the sphere. The exit event is sampled in the usual manner, except that the particle is killed if it tries to enter the sphere because all particles entering the sphere have already been accounted for deterministically.

12. ***Weight window***: As a function of energy, geometrical location, or both, low-weighted particles are eliminated by Russian roulette and high-weighted particles are split. This technique helps keep the weight dispersion within reasonable bounds throughout the problem.

III. MCNP GEOMETRY

We will present here only basic information about geometry setup, surface specification, and cell and surface card input. Areas of further interest would be the complement operator and use of parentheses, found in Chapter 2. Chapter 4 contains geometry examples and is recommended as a next step. Chapter 3 has detailed information on the format and entries on cell and surface cards.

The geometry of MCNP treats an arbitrary three-dimensional configuration of user-defined materials (using up to 50 different tables chosen from the MCNP cross-section libraries) in geometric cells bounded by first- and second-degree surfaces and some special fourth-degree surfaces (elliptical tori). The cells are defined by the intersections, unions, and complements of the regions bounded by the surfaces. Surfaces are defined by supplying coefficients to the analytic surface equations or for certain types of surfaces by supplying known points on the surfaces.

MCNP has a more general combinatorial geometry than is available in most other codes. Rather than combining several predefined geometrical bodies as in a more conventional combinatorial geometry scheme, MCNP gives the user the added flexibility of defining geometrical regions from all the first and second degree surfaces of analytical geometry and elliptical tori and then of combining them with Boolean operators. The code does extensive internal checking to find input errors. In addition, elaborate

geometry-plotting capability in MCNP helps the user define and check the geometry.

MCNP treats geometric cells in a Cartesian coordinate system. The surface equations recognized by MCNP are listed in Table 3.1 on page 178. The particular Cartesian coordinate system used is arbitrary and user defined, but the right-handed system shown in Figure 1.2 is often chosen.

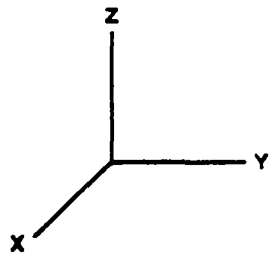


Figure 1.2

Using the cell specifications, MCNP tracks particles through the geometry. MCNP knows all of the bounding surfaces of a cell and calculates the intersection of a track's trajectory with each bounding surface and finds the minimum positive distance to an intersection. If the distance to the next collision is greater than this minimum distance and there are no DXTRAN spheres along the track, the particle leaves the current cell. At the appropriate surface intersection, MCNP finds the correct cell that the particle will enter by checking the sense of the intersection point for each surface listed for the cell. When a complete match is found, MCNP has found the correct cell on the other side and the transport continues.

A. Cells

When cells are defined, an important concept is that of the *sense* of all points in a cell with respect to a bounding surface. Suppose that $s=f(x,y,z)=0$ is the equation of a surface in the problem. For any set of points (x,y,z), if $s=0$ the points are on the surface. However, for points not on the surface, if s is negative the points are said to have a negative sense with respect to that surface and, conversely, a positive sense if s is positive. For example, a point at $x=3$ has a positive sense with respect to the plane $x-2 = 0$. That is, the equation $x-D = 3-2 = s = 1$ is positive for $x = 3$ (where D=constant).

To specify cells in MCNP, list on a cell card the cell number, material number, and material density followed by a list of operators and signed surfaces that bound the cell. If the sense is positive, the sign can be omitted. The material number and material density can be replaced by a single zero to indicate a void cell. The cell number must be in columns 1-5, with the remaining blank-delimited entries in columns 6-72. A

more complete description of the cell card format can be found on page 22. Each surface divides all space into two regions, one with positive sense with respect to the surface and the other with negative sense. The geometry description defines the cell to be the intersection, union, and/or complement of the listed regions.

The subdivision of the physical space into cells is not necessarily governed only by the different material regions, but may be affected by problems of sampling and variance reduction techniques (such as splitting and Russian roulette), the need to specify an unambiguous geometry, and the requirements of the tallies. Using the tally segmentation feature will eliminate most of these requirements for tallies.

You are cautioned about making any one cell very complicated. With the union operator and disjointed regions, a very large geometry can be set up with just one cell. The problem is that for each track flight between collisions in a cell, the intersection of the track with each bounding surface of the cell is calculated, a calculation that can be costly if a cell has many surfaces. As an example, consider Figure 1.3a. It is just a lot of parallel cylinders and is easy to set up. However, the cell containing all the little cylinders is bounded by fourteen surfaces (counting a top and bottom). A much more efficient geometry is seen in Figure 1.3b, where the large cell has been broken up into a number of smaller cells.

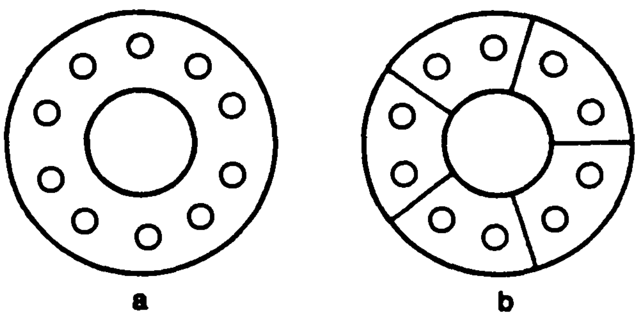


Figure 1.3

1. Cells by Intersections

The intersection operator in MCNP is implicit; it is simply the blank space between two surface numbers on the cell card.

If a cell is specified using only intersections, all points in the cell must have the same sense with respect to a given bounding surface. This means that, for each bounding surface of a cell, all points in the cell must remain on only one side of any particular surface. Thus, there can be no concave corners in a cell. Figure 1.4, a cell formed by intersection of five surfaces (ignore surface 6 for the time being), illustrates the problem of concave corners, allowing a particle (or point)

to be on two sides of a surface in one cell. Surfaces 3 and 4 form a

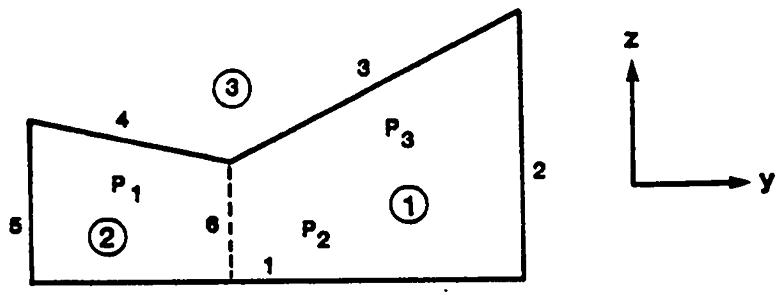


Figure 1.4

concave corner in the cell such that points p_1 and p_2 are on the same side of surface 4 (that is, have the same sense with respect to 4) but point p_3 is on the other side of surface 4 (opposite sense). Points p_2 and p_3 have the same sense with respect to surface 3, but p_1 has the opposite sense. One way to remedy this dilemma (and there are others) is to add surface 6 between the 3/4 corner and surface 1 to divide the original cell into two cells.

With surface 6 added to Figure 1.4, call the cell to the right of surface 6 number 1 (cells indicated by circled numbers); to the left number 2; and the outside cell number 3. The cell cards (in two dimensions and all cells void) are

```
1    0   1    -2  -3  6
2    0   1    -6  -4  5
```

Line 1 says that cell 1 is a void and is formed by the intersection of the region above (positive sense) surface 1 with the region to the left (negative sense) of surface 2 intersected with the region below (negative sense) surface 3 and finally intersected with the region to the right (positive sense) of surface 6. Line 2 describes cell 2 similarly.

Cell 3 cannot be specified with the intersection operator. The following section about the union operator is needed to describe cell 3.

2. Cells by Unions

The union operator, signified by a colon on the cell cards, allows concave corners in cells and also cells that are completely disjoint. The intersection and union operators are binary Boolean operators, so their use follows Boolean algebra methodology; unions and intersections may be used in combination in any cell description.

Spaces on either side of the union operator are irrelevant, but remember that a space without the colon signifies an intersection. In the hierarchy of operations, intersections are performed first and then unions. There is no left to right ordering. Parentheses may be used to clarify

operations and in some cases are required to force a certain order of operations. Spaces are optional on either side of a parenthesis. In fact, a parenthesis is equivalent to a space (after any ordering has taken place) and therefore signifies an intersection.

For example, let A and B be two regions of space. The region containing points that belong to both A and B is called the intersection of A and B. The region containing points that belong to A alone or to B alone or to both A and B is called the union of A and B. The lined area in Figure 1.5a represents the union of A and B (or A : B), and the lined area in Figure 1.5b represents the intersection of A and B (or A B). The only way regions of space can be added is with the union operator. An intersection of two spaces always results in a region no larger than either of the two spaces. Conversely, the union of two spaces always results in a

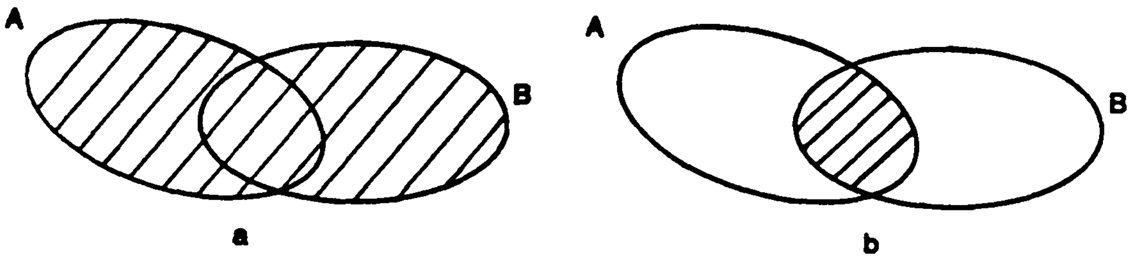


Figure 1.5

region no smaller than either of the two spaces.

A simple example will further illustrate the concept of Figure 1.5 and the union operator to solidify the concept of adding and intersecting regions of space to define a cell. See also the second example in Chapter 4. In Figure 1.6 we simply have two planes that meet to form two cells. Cell 1 is simple to define; it is everything in the universe to the right of surface 1 (that is, a positive sense) that is also in common with (or intersected with) everything in the universe below surface 2 (that is, a negative sense). Therefore, the surface relation of cell 1 is 1 −2.

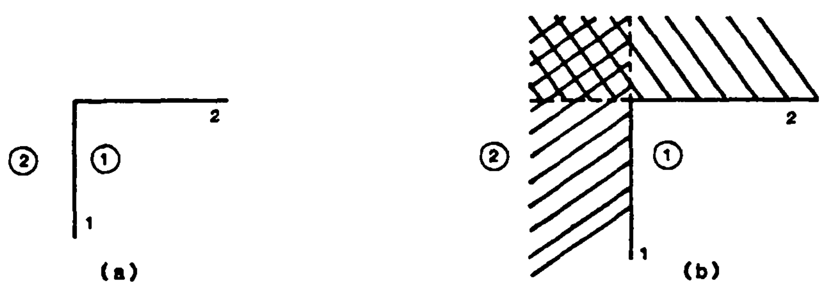


Figure 1.6

Cell 2 is everything in the universe to the left (negative sense) of

surface 1 plus everything in the universe above (positive sense) surface 2, or −1 : 2, illustrated in Figure 1.6 by all the shaded regions of space. If cell 2 were specified as −1 2, that would represent the region of space common to −1 and 2, which is only the cross-hatched region in the figure and is obviously an improper specification for cell 2.

Returning to Figure 1.4 on page 15, if cell 1 is inside the solid black line and cell 2 is the entire region outside the solid line, then the MCNP cell cards in two dimensions are (assuming both cells are voids)

```
1    0   1      -2      (-3  :  -4)  5
2    0  -5  :   -1  :     2  :   3   4
```

Cell 1 is defined as the region above surface 1 intersected with the region to the left of surface 2, intersected with the union of regions below surfaces 3 and 4, and finally intersected with the region to the right of surface 5. Note that cell 2 contains four concave corners (all but between surfaces 3 and 4), and its specification is just the converse (or complement) of cell 1. Cell 2 is the space defined by the region to the left of surface 5 plus the region below 1 plus the region to the right of 2 plus the space defined by the intersections of the regions above surfaces 3 and 4.

A simple consistency check can be noted with the two cell cards above. All intersections for cell 1 become unions for cell 2 and vice versa. The senses are also reversed.

Note that in this example, all legitimate corners (less than 180 degrees) in a cell are handled by intersections and all illegitimate corners (greater than 180 degrees) are handled by unions.

To illustrate some of the concepts about parentheses, assume an intersection is thought of mathematically as multiplication and a union is thought of mathematically as addition with multiplication being done first unless parentheses are present. The cell cards for the example cards above from Figure 1.4 may be written in the form

```
1    a·b·(c+d)·e
2    e+a+b+c·d
```

Note that parentheses are required for the first cell but not for the second, although the second could have been written as e+a+b+(c·d), (e+a+b)+(c·d), (e)+(a)+(b)+(c·d), etc.

Several more examples using the union operator are given in Chapter 4. Study them to get a better understanding of this powerful operator that can greatly simplify geometry setups.

B. Surface Type Specification

The first- and second-degree surfaces plus the fourth-degree elliptical and degenerate tori of analytical geometry are all available in MCNP. The surfaces are designated by mnemonics such as C/Z for a cylinder parallel to the z-axis. A cylinder at an arbitrary orientation is designated by the general quadratic GQ mnemonic. A paraboloid parallel to a coordinate axis is designated by the special quadratic SQ mnemonic. The 29 mnemonics representing various types of surfaces are listed in Table 3.1 on page 178.

C. Surface Parameter Specification

There are two ways to specify surface parameters in MCNP: (1) by supplying the appropriate coefficients needed to satisfy the surface equation, and (2) by specifying known geometrical points on a surface that is symmetric about a coordinate axis.

1. Coefficients

The first way to define a surface is to use one of the surface-type mnemonics from Table 3.1 on page 178 and to calculate the appropriate coefficients needed to satisfy the surface equation. For example, a sphere of radius 3.62 with the center located at the point (4,1,-3) is specified simply by

```
S     4  1  -3  3.62
```

An ellipsoid whose axes are not parallel to the coordinate axes is defined by the GQ mnemonic plus up to 10 coefficients of the general quadratic equation. Calculating the coefficients can be (and frequently is) nontrivial, but the task is greatly simplified by defining an auxiliary coordinate system whose axes coincide with the axes of the ellipsoid. The ellipsoid is easily defined in terms of the auxiliary coordinate system, and the relationship between the auxiliary coordinate system and the main coordinate system is specified on a TRn card, described on page 197.

The use of the SQ (special quadratic) and GQ (general quadratic) surfaces is determined by the orientation of the axes. One should always use the simplest possible surface in describing geometries; for example, using a GQ surface instead of an S to specify a sphere will require more computational effort for MCNP.

2. Points

The second way to define a surface is to supply known points on the surface. This method is convenient if you are setting up a geometry from

something like a blueprint where you know the coordinates of intersections of surfaces or points on the surfaces. When three or more surfaces intersect at a point, this second method also produces a more nearly perfect point of intersection if the common point is used in the surface specification. It is frequently difficult to get complicated surfaces to meet at one point if the surfaces are specified by the equation coefficients. Failure to achieve such a meeting can result in the unwanted loss of particles.

There are, however, restrictions that must be observed when specifying surfaces by points that do not exist when specifying surfaces by coefficients. Surfaces described by points must be either skew planes or surfaces symmetric about the x, y, or z axes. They must be unique, real, and continuous. For example, points specified on both sheets of a hyperboloid are not allowed because the surface is not continuous. However, it is valid to specify points that are all on one sheet of the hyperboloid. (See the X,Y,Z, and P input cards description on page 180 for additional explanation.)

IV. DESCRIPTION OF MCNP INPUT AND SAMPLE PROBLEM

The main input file for the user is the INP (the default name) file, which contains the input information to describe the problem. We will present here only the subset of cards required to run our simple demonstration problem. All input cards are discussed in Chapter 3 and summarized in Table 3.4 starting on page 267.

MCNP does extensive input checking but is not foolproof. A geometry should be checked by looking at several different views with the PLOT option and then running a short job and studying the output to see if you are calculating what you think you are calculating.

You should also surround the entire geometry with a sphere and flood the geometry with particles from a source with an inward cosine distribution on the spherical surface, using a VOID card to remove any materials specified in the problem. If there are any incorrectly specified places in your geometry, this procedure will usually find them. Make sure the importance of the cell just inside the source sphere is not zero.

The units used throughout MCNP are:

1. lengths in centimeters,
2. energies in MeV,
3. times in shakes (10^{-8} sec),
4. temperatures in MeV (kT),
5. atomic densities in units of atoms/barn-cm,
6. mass densities in g/cm^3,
7. cross sections in barns (10^{-24} cm^2),
8. heating numbers in MeV/collision, and
9. atomic weights based on a neutron mass of 1.0.

In these units, Avogadro's number is 0.59703109.

A simple sample problem illustrated in Figure 1.7 will be referred to throughout the remainder of this chapter. We wish to start 14-MeV neutrons at a point isotropic source in the center of a small sphere of oxygen that is embedded in a chunk of carbon. A small sphere of iron is also embedded in the carbon. The carbon is a cube 10 cm on each side; the spheres have a 0.5-cm radius and are centered between the front and back faces of the cube. We wish to calculate the total and energy-dependent flux in increments of 1 MeV from 14 to 1 MeV and then order of magnitude increments below 1 MeV to 1×10^{-5} MeV

1. on the surface of the iron sphere and
2. averaged in the iron sphere volume.

This geometry has four cells, indicated by circled numbers, and eight surfaces - six planes and two spheres. Surface numbers are written next to the appropriate surfaces. Surface 5 comes out from the page in the +x direction and surface 6 goes back into the page in the -x direction.

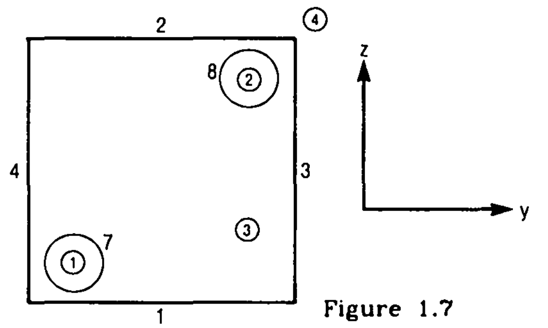


Figure 1.7

With knowledge of the cell card format, the sense of a surface, and the union and intersection operators, we can set up the cell cards for the geometry of our example problem. To simplify this step, assume the cells are void, for now. Cells 1 and 2 are described by the following cards:

```
1 0 -7
2 0 -8
```

where the negative signs denote the regions inside (negative sense) surfaces 7 and 8. Cell 3 is everything in the universe above surface 1 intersected with everything below surface 2 intersected with everything to the left of surface 3 and so forth for the remaining three surfaces. The region in common to all six surfaces is the cube, but we need to exclude the two spheres by intersecting everything outside surface 7 and outside surface 8. The card for cell 3 is

```
3 0 1 -2 -3 4 -5 6 7 8
```

Cell 4 requires the use of the union operator and is similar to the idea illustrated in Figure 1.6. Cell 4 is the outside world, has zero importance, and is defined as everything in the universe below surface 1 plus everything above surface 2 plus everything to the right of surface 3 and so forth. The cell card for cell 4 is

```
4 0 -1:2:3:-4:5:-6
```

A. INP File

The input file that will enable us to initiate and run a problem has the following form:

Message Block } Optional
Blank Line Delimiter
Title Card
Cell Cards
.
.
Blank Line Delimiter
Surface Cards
.
.
Blank Line Delimiter
Data Cards
.
.
Blank Line Terminator

The optional message block, discussed in detail on page 165, is used to change file names and specify running options such as a continuation run. On most systems other than IBM, these options and files may alternatively be specified with an execution line message (see page 30). The blank line delimiter signals the end of the message block.

The first card in the file after the optional message block is the required problem title card. If there is no message block, this must be the first card in the INP file. It is limited to one line from columns 1–80 and is used as a title in various places in the MCNP output. It can contain any information you desire but usually contains information describing the particular problem.

All input lines (referred to also as "cards" in this manual) of INP are limited to columns 1–80 and consist of card images. Columns 1–72 are used for data, and columns 73–80 can be used for comments. In addition, if the symbol **$** (dollar sign) appears in a line, data entry on that line is terminated and anything following the **$** is interpreted as a comment. A future version of the code will drop the distinction between columns 1–72 and columns 73–80 and will rely entirely on the dollar sign to signal a comment field within a data line. Blank cards are used as delimiters and terminators.

Comment cards may be used anywhere in the INP file after the problem title card and before the last blank terminator card. These cards must have the letter C in column 1 followed by four blanks, and then columns 6–80 are available to the user for any comments.

Cell cards, surface cards, and data cards all conform to the same format. Columns 1–5 are reserved for the name or number associated with the card. The name or number can appear anywhere in columns 1–5. Blanks filling the first five columns indicate a continuation of the data from the last named card. Completely blank cards are reserved to be delimiters between major sections of the input file. Columns 6–72 are for free-field format data entry associated with the card name. Separation between data entries is by one or more blank columns. In general, data entries may be of any type (integer or floating point) inasmuch as MCNP makes the appropriate conversion. Each data entry must be completed entirely on one card.. Continuation lines are considered as part of the same card.

MCNP makes extensive checks of the input file for user errors. A FATAL error occurs if a basic constraint of the input specification is violated, and MCNP will terminate before running any particles. The first fatal error is real; subsequent error messages may or may not be real because of the nature of the first fatal message.

B. Cell Cards

The cell number is in columns 1–5. Columns 6–72 will contain, in the following order:

1. the cell material number, which may be arbitrarily assigned by the user. The material is described by a material card having the same material number (see page 28). If the cell is a void, a zero should be entered for the material number.

2. the cell material density. A positive entry is interpreted as atom density in units of 10^{+24} atoms/cm^3. A negative entry is interpreted as mass density in units of g/cm^3. No density is entered for a void cell.

3. a complete specification of the geometry of the cell. This specification includes a list of the signed surfaces bounding the cell where the sign denotes the sense of the regions defined by the surfaces. The Boolean relations between the regions bounded by the surfaces are a blank for intersection and a colon for union.

4. optionally, cell parameters, instead of in the data card section of the INP file. The form is keyword=value.

As an example of the above format, cell 1 in our problem contains material 1 (defined on the M1 card described on page 28) with density 0.0014 g/cm^3, is bounded by only one surface, and has an importance of 1. The proper cell card is

```
1   1   -0.0014   -7   INP:N=1
```

If cell 1 were a void, the cell card would be

```
1   0   -7   INP:N=1
```

The complete list of cell cards for this problem is

```
1    1 -0.0014  -7
2    2 -7.86    -8
3    3 -1.60     1 -2 -3 4 -5 6 7 8
4    0          -1:2:3:-4:5:-6
blank card
```

The blank card terminates the cell card section of the input file. Again, we strongly suggest that the cells be numbered sequentially starting with one. A more complete explanation of the cell card can be found in Chapter 3, Sec. II, page 175.

C. *Surface Cards*

The surface number appears in columns 1-5. Columns 6-72 contain, in the following order:

1. an alphabetic mnemonic indicating the surface type, and
2. the required numerical entries for the specific coefficients of the surface in proper order.

This simplified description will enable us to proceed with our example problem. For a full description of the surface card see page 177.

Our problem uses planes normal to the x, y, and z axes and two general spheres. The respective mnemonics are PX, PY, PZ, and S. Table 1.2 shows the equations that determine the sense of the surface for the cell cards and the entries required on the surface cards. A complete list of available surface equations is contained in Table 3.1 on page 178.

Table 1.2
Surface Equations

Mnemonic	Equation	Card Entries
PX	$x-D=0$	D
PY	$y-D=0$	D
PZ	$z-D=0$	D
S	$(x-\bar{x})^2+(y-\bar{y})^2+(z-\bar{z})^2-R^2=0$	$\bar{x}$ $\bar{y}$ $\bar{z}$ R

For the planes, D is the point where the plane intersects the axis. If we place the origin in the center of the 10-cm cube shown in Figure 1.7, the planes will be at x=-5, x=5, etc. The two spheres are not centered at the origin or on an axis, so we must give the x,y,z of their center as well as their radii. The complete list of surface cards for this problem is shown below. The blank card terminates the surface card portion of the input.

```
1   PZ  -5
2   PZ   5
3   PY   5
4   PY  -5
5   PX   5
6   PX  -5
7    S   0  -4  -2.5  .5
8    S   0   4   4    .5
blank card
```

D. Data Cards

All input cards to MCNP other than cell and surface cards are after the second blank card delimiter or third blank card if there is a message block. Several of the input cards require a particle designator to distinguish between input data for neutrons and data for photons. The particle designator consists of the symbol : (colon) and the letter N or P immediately following the name of the card. For example, to enter neutron importances, use an IMP:N card; enter photon importances on an IMP:P card. No data card can be used more than once with the same mnemonic, that is, M1 and M2 are acceptable, but two M1 cards are not allowed. Defaults have been set for cards in some categories. A summary starting on page 267 shows whether cards are required or optional and whether defaults exist and if so, what they are. The sample problem will use cards in the following categories:

1. mode,
2. cell and surface parameters,
3. source specification,
4. tally specification,
5. material specification, and
6. problem cutoffs.

A more complete description of all the data cards can be found on page 183 in Chapter 3. The format of the data cards consists of the mnemonic in columns 1–5 followed by the required entries in a free-field format in columns 6–72. Sometimes the mnemonic plus particle designator will spill past column 5. Then the colon must be in column 6 and the N or P must be followed by a space.

1. MODE card

MCNP can be run in three different modes:

Mode N – neutron transport (default)
N P – neutron and neutron-induced photon transport
P – photon transport

The MODE card consists of the mnemonic MODE in columns 1-5 and either an N, N P, or P in columns 6-72, depending on which mode is being used. If the MODE card is omitted, mode N is assumed.

Mode N P does not account for photo-neutrons but only neutron-induced photons. Gamma-production cross sections do not exist for all nuclides. If they are not available for a Mode N P problem, MCNP will print out warning messages. To find out whether a particular table for a nuclide has gamma-production cross sections available, check the Appendix G cross-section list.

The sample problem is a neutron-only problem, so the MODE card can be omitted because MODE N is the default.

2. Cell and surface parameter cards

Most of these cards define values of cell parameters. Entries on these cards correspond in order to the cell or surface cards that appear earlier in the INP file. The mnemonic goes in columns 1-5 and the data entries go in columns 6-72. A listing of all available cell and surface parameter cards is found on page 184. A few examples are neutron and photon importance cards (IMP:N,IMP:P), weight window cards (WWE:N, WWE:P, WWNi:N, WWNi:P), etc. Some method of specifying relative cell importances is required; the majority of the other cell parameter cards are for optional variance reduction techniques. The number of entries on a cell or surface parameter card must equal the number of cells or surfaces in the problem or MCNP prints out a warning.

The IMP:N card will be used to specify relative cell importances in the sample problem. There are four cells in the problem, so the IMP:N card will have four entries. The IMP:N card is used (a) for terminating the particle's history if the importance is zero and (b) for geometry splitting and Russian roulette to help particles move more easily to important regions of the geometry. An IMP:N card for the sample problem would be

```
IMP:N    1  1  1  0
```

Cell parameters also can be defined on cell cards using the keyword=value format. If a cell parameter is specified on any cell card, it must be specified only on cell cards and not at all in the data card section.

3. Source specification cards

A source definition card SDEF is one of four available methods of defining starting particles. In Chapter 3 Section D is the complete discussion of source specification. The mnemonic is entered in columns 1-5 and the data are entered in columns 6-72. The SDEF card defines the basic source parameters, some of which are

```
POS = x y z                default is 0 0 0;
CEL = starting cell number;
ERG = starting energy      default is 14 MeV;
```

WGT = starting weight   default is 1;
TME = time   default is 0.

MCNP will determine the starting cell number for a point isotropic source, so the CEL entry is not always required. The default starting direction for source particles is isotropic.

For the example problem, a fully specified source card is

SDEF POS=0 -4 -2.5 CEL=1 ERG=1 4 WGT=1 TME=0

Particles will start at the center of the oxygen sphere (0 -4 -2.5), in cell 1, with an energy of 14 MeV, and with weight 1 at time 0. All these source parameters except the starting position are the default values, so the most concise source card is

SDEF POS=0 -4 -2.5

If all the default conditions were satisfactory for the problem, only the mnemonic would be required.

4. Tally specification cards

The tally cards are used to specify what you want to learn from the Monte Carlo calculation, that is, current across a surface, flux at a point, etc. You request this information with a combination of cards. No cards in this category are required, but if none is supplied, no tallies will be printed when the problem is run. As with other cards, the mnemonic goes in columns 1–5 and the data entries go in columns 6–72. Much of the information on other cards in this category is used to describe tally "bins." A few examples are energy (En), time (Tn), and cosine (Cn) cards.

MCNP provides six standard neutron and five standard photon tallies, all normalized to be per starting particle. Chapter 2, Sec. V, page 80, discusses tallies more completely and Chapter 3, Sec. IV.E., page 219, lists all the cards associated with tallies with a full description of each one.

Tally Mnemonic	Description
F1:N or F1:P	Surface current
F2:N or F2:P	Surface flux
F4:N or F4:P	Track length estimate of cell flux
F5a:N or F5a:P	Flux at a point (point detector)
F6:N or F6:N,P or F6:P	Track length estimate of energy deposition
F7	Track length estimate of fission energy deposition

The tallies are identified by tally type and particle type as follows. Tallies are given the numbers 1, 2, 4, 5, 6, 7, or increments of 10 thereof, and are given the particle designator :N or :P (or :N,P only in

the case of tally type 6). Thus you may have as many of any basic tally as you need, each with different energy bins or flagging or anything else. F4:N, F14:N, F104:N, and F234:N are all legitimate neutron cell flux tallies; they could all be for the same cell(s) but with different energy or multiplier bins, for example. Similarly F5:P, F15:P, and F305:P are all photon point detector tallies. Having both an F1:N card and an F1:P card in the same INP file is not allowed. The tally number may not exceed three digits.

For our sample problem we will use Fn cards (Tally type) and En cards (Tally energy).

<u>a. Tally (Fn) Cards</u>: The sample problem has a surface flux tally and a track length cell flux tally. Thus, the tally cards for the sample problem shown in Figure 1.7 are

```
F2:N    8
F4:N    2
```

Printed out with each tally bin is the relative error of the tally corresponding to one standard deviation. Read page 6 for an explanation of the relative error. Results are not reliable until they become stable as a function of the number of histories run. An indication of tally stability can be seen in the FOMs listed in the tally fluctuation charts at the end of the printed output.

<u>b. Tally Energy (En) Card</u>: Another tally specification card used in the sample input deck is the card that establishes energy bins because we wish to calculate flux in increments of 1 MeV from 14 to 1 MeV and then order of magnitude increments below 1 MeV to 1×10^{-5} MeV.

The entries on this card are the upper bounds in MeV of the energy bins for tally n. The entries must be given in order of increasing magnitude. If a particle has an energy greater than the last entry, it will not be tallied, but you will be warned this has happened. MCNP automatically provides the total over all specified energy bins unless inhibited by putting the symbol NT as the last entry on the selected En card.

The following cards will create the desired energy bins for the sample problem:

```
E2    1E-5 1E-4 1E-3 .01 .1 1 2 3 4 5 6 7 8 9 10 11 12 13
E4    1E-5 1E-4 1E-3 .01 .1 1 2 3 4 5 6 7 8 9 10 11 12 13
```

If no En card exists for tally n, a single bin over all energy will be used. To change this default, an E0 (zero) card may be used to set up a default energy bin structure for <u>all</u> tallies. A specific En card will override the default structure for tally n. Therefore, we could replace the E2 and E4 cards with one E0 card for the sample problem, thus setting up bins for both tallies.

5. Materials specification

The cards in this section specify both the isotopic composition of the materials and the cross-section evaluations to be used in the cells. For a more comprehensive discussion of materials specification, see page 247.

a. Material (Mm) Card: The following card is used to specify a material for all cells containing material m:

$$\text{Mm} \qquad \text{ZAID}_1 \quad \text{fraction}_1 \quad \text{ZAID}_2 \quad \text{fraction}_2 \quad \ldots.$$

The m on a material card corresponds to the material number on the cell card (see page 22). The Mm goes in columns 1–5, and the data entries go in columns 6–72.

The consecutive pairs of entries on the material card consist of the identification number (ZAID) of the constituent element or nuclide followed by the atomic fraction (or weight fraction if entered as a negative number) of that element or nuclide, until all the elements and nuclides needed to define the material have been listed.

i. Nuclide Identification Number (ZAID). This number is used to identify the element or nuclide desired. The form of the number is ZZZAAA.nnX, where

ZZZ is the atomic number of the element or nuclide,
AAA is the mass number of the nuclide, and
nn is the cross-section evaluation identifier; if blank or zero, a default cross-section evaluation will be used.
X is the class of data: C is continuous energy; D is discrete reaction; T is thermal; Y is dosimetry; and P is photon.

For naturally occurring elements, AAA=000. Thus ZAID=74182.01 represents the isotope ${}^{182}_{74}W$, and ZAID=74000.01 represents the element tungsten.

ii. Nuclide Fraction. The nuclide fractions may be normalized to 1 or left unnormalized. For example, if the material is H_2O the fractions can be entered as .667 and .333 or as 2 and 1 for H and O respectively. If the fractions are entered with negative signs, they are weight fractions; otherwise they are atomic fractions. Weight fractions and atomic fractions cannot be mixed on the same card.

The material cards for the sample problem are

```
M1    8016.50C  1
M2   26000.55C  1
M3    6000.50C  1
```

b. VOID Card: A VOID card is a useful part of the MCNP vocabulary. The mnemonic VOID is entered in columns 1-5 and, in the sample problem case, no data entries follow.

When the VOID card is used, all materials and cross sections are negated and all nonzero importances are set to unity; therefore, many particles can be run in a short time. Flooding the geometry with many particles enhances greatly the chance of particles going to most parts of the geometry - in particular, to an incorrectly specified part of the geometry - and getting lost. The VOID card is very effective for finding errors in the geometry description. The other actions of and uses for the VOID card are discussed on page 251.

The sample input deck could have a VOID card while testing the geometry for errors. When you are satisfied that the geometry is error-free and ready to run for tallying, the VOID card is removed.

6. Problem Cutoffs

Problem cutoff cards are used to specify parameters for some of the ways to terminate execution of MCNP. The full list of available cards and a complete discussion of problem cutoffs is found on page 257. For our problem we will use only the history cutoff (NPS) card. The mnemonic NPS goes in columns 1-5 and is followed by a single entry that specifies the number of histories to transport. MCNP will terminate after NPS histories unless it has terminated earlier for some other reason.

7. Sample Problem Summary

The entire input deck for the sample problem follows. Note that the mnemonics are shown as capital letters. The input typed at your terminal may have to be lower case letters.

```
SAMPLE PROBLEM INPUT DECK
1    1 -0.0014  -7
2    2 -7.86    -8
3    3 -1.60     1 -2 -3 4 -5 6 7 8
4    0          -1:2:3:-4:5:-6
blank card
1    PZ  -5
2    PZ   5
3    PY   5
4    PY  -5
5    PX   5
6    PX  -5
7     S   0  -4  -2.5  .5
8     S   0   4   4    .5
blank card
```

```
IMP:N 1 1 1 0
 SDEF POS=0 -4 -2.5 CEL=1 ERG=14 WGT=1 TME=0
   F2 8
   F4 2
   E0 1E-5 1E-4 1E-3 .01 .1 1 2 3 4 5 6 7 8 9 10 11 12 13
   M1  8016.50C 1
   M2 26000.55C 1
   M3  6000.50C 1
  NPS 1000000
blank card
```

V. *HOW TO RUN MCNP AT LOS ALAMOS*

We will assume you have a basic knowledge of CTSS. If this is not true, contact the Consultants in C Division.

The executable binary file for version 3A is called MCNP and is a public file that you do not have to get from the common file storage. The cross-section directory XSDIR is also a public file on CTSS.

A. *Execution Line*

The MCNP execution line has the following form:

MCNP *Files Options / t p*

Files and *Options* are described below. The order on the execution line of *Files* and *Options* is irrelevant.

1. *Files*

MCNP uses several files for input and output. The files pertinent to the sample problem are shown in Table 1.3. File INP must be present as a local file. MCNP will create OUTP and RUNTPE.

Table 1.3
MCNP Files

Default Name	Description
INP	Problem input specification
OUTP	BCD output for printing
RUNTPE	Binary start-restart data
XSDIR	Cross-section directory

The default name of any of the files in Table 1.3 can be changed on the MCNP execution line by entering

Default $Name_1$=User File $Name_1$

For example, if you have an input file called MCIN and wish the BCD output file to be MCOUT, the execution line would be

MCNP INP=MCIN OUTP=MCOUT / *t p*

If there are no changes in default names, nothing is entered for *Files*.

If a file in your local file space has the same name as a file MCNP needs to create, the file is created with a different unique name by changing the last letter of the name of the new file to the next letter in the alphabet. For example, if you already have an OUTP, MCNP will create OUTQ.

2. *Options*

There are two kinds of options, overlay execution options and other options. The other options are discussed on page 417. Execution options are discussed next.

MCNP consists of up to four distinct operations, each performed by a single overlay. These operations, their corresponding overlays, and a one-letter mnemonic for each operation are listed in Table 1.4.

Table 1.4
Execution Options

Mnemonic	Overlay	Operation
I	IMCN	Process problem input file
P	PLOT	Plot geometry
X	XACT	Process cross sections
R	MCRUN	Particle transport

When *Options* are omitted, the default is IXR. The execution of the overlays can be controlled by entering the proper mnemonic on the execution line. If more than one operation is desired, the single characters (in any order) may be combined to form a string. Examples of use are type I to look for input errors, IP to debug a geometry by plotting, and IX to see how much cross-section space is required.

Upon execution, the first thing MCNP will do is create a drop file named +MCNP (or +MCNPA, etc., if a file of the first name already exists). If MCNP terminates because of insufficient time, restart the drop file by typing:

+MCNP / t p

After a job has been run, the BCD print file OUTP in your local file space may be examined with an editor on the computer and/or sent to a printer and/or microfiche with HSP as illustrated in Section C, following.

B. Interrupts

MCNP allows four interactive interrupts while it is running:

(ctrl e)IS	MCNP status
(ctrl e)IT *time*	Change time remaining to *time* in minutes
(ctrl e)IQ	Terminates MCNP
(ctrl e)IP	Pause; continue with RETURN key

The status of MCNP consists of the overlay being run, the computer time used so far, and if in MCRUN the number of particles run so far.

The IQ interrupt has no effect until MCRUN is executed. Then this command causes the code to stop after the current particle history. The process allows MCNP to terminate "gracefully," producing a final print output file.

C. Running MCNP

Press the RETURN key after each input line described in this section.

To run the example problem have the input file in your local file space. For illustration, assume the file is called SAMPLE. Type

MCNP INP=SAMPLE / *t p*

MCNP will produce an output file OUTP that you can examine at your terminal, send to a printer, or both by

HSP OUTP -UC -SEQ T="*your title*"

To look at the geometry with the PLOT overlay, type in

MCNP IP INP=SAMPLE / *t p*

After you are prompted with a question mark, use an interactive graphics terminal and type in

PX=0 EX=20

This plot will show an intersection of the surfaces of the problem by the plane X=0 with an extent in the x-direction of 20 cm on either side of the origin. If you want to do more with PLOT, see the instructions on

page 422. Otherwise type END after the next prompt to terminate the session.

Instructions for running the code with more complicated procedures such as setting up controllers and updating the code are in Appendix A.

VI. *TIPS FOR CORRECT AND EFFICIENT PROBLEMS*

This section has a brief checklist of helpful hints that apply to three phases of your calculation: defining and setting up the problem, preparing for the long computer runs that you may require, and making the runs that will give you results. Not everything mentioned in the checklist has been covered in this chapter, but the list can serve as a springboard for further reading in preparation for tackling more difficult problems.

A. *Problem Setup*

1. Use the best problem cutoffs.
2. Use zero for the neutron energy cutoff (MODE 1).
3. Do not use too many variance reduction techniques.
4. Use conservative variance reduction techniques.
5. Do not use cells with many mean free paths.
6. Use uncomplicated cells.
7. Use the simplest surfaces.
8. Study warning messages.
9. Always plot the geometry.
10. Use the VOID card when checking geometry.
11. Use separate tallies for the fluctuation chart.
12. Generate the best output (consider PRINT card).
13. RECHECK THE INP FILE (materials, densities, sources, etc.).

B. *Preproduction*

1. Run some short jobs.
2. Examine the outputs carefully.
3. Study the ledger tables.
4. Compare the figures of merit and stability.
5. Consider the collisions per source particle.
6. Examine the track populations.
7. Scan the mean free path column.
8. Check detector diagnostic tables.
9. Understand large detector contributions.
10. Strive to eliminate unimportant tracks.
11. Check MODE 1 photon production.
12. Do a back-of-the-envelope check of the results.
13. DO NOT USE MCNP AS A BLACK BOX.

C. *Production*

1. Save RUNTPE (output file, continue run, tally plotting).
2. Look at figure of merit stability.
3. Make sure answers seem reasonable.
4. Make continue runs if necessary.
5. See if stable errors decrease by $1/\sqrt{N}$
   (that is, be careful of the brute force approach).
6. Remember, accuracy is only as good as the nuclear data, modeling, MCNP sampling approximations, etc.

CHAPTER 2
GEOMETRY, DATA, PHYSICS, AND MATHEMATICS

I. INTRODUCTION

Chapter 2 discusses in more detail the mathematics and physics of MCNP, including geometry, cross-section libraries, sources, variance reduction schemes, Monte Carlo simulation of neutron and photon transport, and tallies. This discussion is not meant to be exhaustive; many details of the particular techniques and of the Monte Carlo method itself will be found elsewhere. Carter and Cashwell's book *Particle-Transport Simulation with the Monte Carlo Method*,[10] a good general reference on radiation transport by Monte Carlo, is based upon what is in MCNP. Methods of sampling from standard probability densities are discussed in reference 11.

MCNP was originally developed by the Monte Carlo Group, currently the Radiation Transport Group, (Group X-6) in the Applied Theoretical Physics Division (X Division) at the Los Alamos National Laboratory. Group X-6 improves MCNP (releasing a new version about every 18 months), maintains it at Los Alamos and on the Magnetic Fusion Energy (MFE) computers at Livermore, California, and provides consulting and support for MCNP users. MCNP is distributed to other users through the Radiation Shielding Information Center (RSIC) at Oak Ridge, Tennessee.

Reference 12 is a collection of four papers relating to MCNP: (1) an overview of Group X-6 and the features and uses of MCNP, (2) the geometry in MCNP, (3) flux at a point in MCNP, and (4) deep penetration calculations with MCNP. Reference 13 contains lectures and papers presented at a joint Monte Carlo conference held April 1985.

MCNP has approximately 23 000 lines of source coding (including Comments and with the COMMON blocks listed only once and not in every subroutine). There is only one source code; it is used for all systems. At Los Alamos, MCNP and patched versions of the code are used about 1000 times a month by many users, totaling about 200 hours of CPU time.

MCNP has not been successfully vectorized because the overhead required to set up and break apart vector queues at random decision points is more significant than the savings from vectorizing the simple arithmetic between the decision points. MCNP (and any general Monte Carlo code) is little more than a collection of random decision points with some simple arithmetic in between.

MCNP has been made as system independent as possible to enhance its portability, and has been written to comply with the ANSI FORTRAN 77 standard. With one source code, MCNP has been running on CRAY, CDC, IBM, VAX, PRIME, SIEMENS, RIDGE, APOLLO, and SUN machines.

CHAPTER 2
Introduction

A. *History*

Although the first use of Monte Carlo techniques is difficult to trace, the emergence of the Monte Carlo method as a radiation transport research tool springs from work done at Los Alamos during World War II. Credit for the so-called invention of Monte Carlo as a mathematical discipline is generally given to Fermi, von Neumann, and Ulam. Their ideas were developed by many followers in various laboratories, but the name of von Neumann seems to be attached to many, if not most, of the fundamental ideas and techniques. Metropolis and Richtmyer should also be credited with some of the early Monte Carlo development at Los Alamos. In 1947 while at Los Alamos, Fermi invented a mechanical device called FERMIAC[14] to trace neutron movements through fissionable materials by the Monte Carlo method.

Since its inception, Monte Carlo at Los Alamos has had a number of devoted and well-known disciples. In the early 1950s there was assembled at Los Alamos, under Ulam's guidance, a small band of scientists engaged in investigating transport methods and solving transport problems by Monte Carlo. At first the calculational effort was rudimentary, but with the completion of the MANIAC computer, the work gained momentum and rapidly became a factor in much of the research at Los Alamos. Much of the early work is summarized in the first book to appear on the subject of Monte Carlo.[15] Shortly thereafter at Los Alamos, the Monte Carlo neutron transport code MCS,[16] which incorporated a general geometry treatment, was written.

The addition of several more features resulted in the Monte Carlo neutron code MCN[17] in 1965. These features included improvement and expansion of the geometry treatment, provision for standard cross-section libraries, inclusion of standard variance reduction techniques, inclusion of general sources, treatment of thermalization of neutrons by the free-gas model, and more general tallies and output. The photon codes MCG and MCP[18] were then added to the Los Alamos family of Monte Carlo codes. Both of these dealt with photon transport, MCG treating higher-energy photons or gamma rays and MCP treating photons of energies down to 1 keV. MCN and MCG were merged to form MCNG (the predecessor of MCNP) in 1973. Photon production cross sections were added to enable the gamma rays produced by neutron interactions to be generated.

B. *MCNP Structure*

MCNP is the culmination of all the above work and codes plus many additions in mathematics, physics, and user-oriented features. The first version of MCNP was released in June 1977. There have since been nine revisions, Versions 1A, 1B, 2, 2A, 2B, 2C, 2D, 3, and now 3A. Version 3

was, for the user, almost the same as 2D but was completely rewritten to comply with the ANSI FORTRAN 77 standard. Version 3A, however, does have significant changes in code capabilities.

The general structure of MCNP on CDC machines is based on overlays or segments, as they are called on NOS. The main overlay MCNP calls up to four other overlays, depending upon the user's requirements. IMCN (overlay 1C0 and commonly called the initiation code) processes the problem input file and calculates volumes and areas; it calls four lower-level overlays (1C1, 1C2, 1C3, and 1C4). PLOT (overlay 2C0) allows interactive plotting of the problem geometry and calls two lower-level overlays, 2C1 and 2C2. XACT (overlay 3C0) processes cross sections for the specified materials. MCRUN (overlay 4C0, commonly called the run code) actually transports the particle. It calls two lower-level overlays, 4C1 where the actual transport is done and 4C2 for the output.

On other systems, MCNP does not use overlays (or segments) but the order in which things are done is the same as on CDC machines.

MCNP can be run in one of three modes: neutron transport only, combined neutron/photon transport, and photon transport only. In the combined neutron/photon mode, photon histories are followed as part of their parent neutron histories.

C. Variants and Future Work

Variants of MCNP offer a material perturbation feature; delayed neutrons; a limited geometrical lattice capability; neutron coincidence counter simulation; a multigroup adjoint capability; a state-space splitting capability; flux, importance, and contributon flux data generation with subsequent plotting; a radiography capability using detector mechanics in a plane; a calculation of the variance of the variance; a thick target bremsstrahlung generation model; and multitasking. Some of these features may become a permanent part of MCNP at a later date.

We have also developed the ability to transport high-energy (up to the GeV range) particles such as protons, pions, muons, etc. For example, from an 800-MeV proton source the complete cascade of charged particles is followed, accounting for actual energy deposition along the way until the last low-energy photon and thermal neutron have been stopped. This work uses MCNP as part of the HETC[19] code system.

Future work includes a general geometrical lattice capability, the probability table method for unresolved resonances, photo/neutron reactions, improved variance reduction schemes, links to discrete ordinates codes, higher-energy neutron cross sections (perhaps up to 60 MeV in some cases), incorporation of our stand-alone output plotter, and electron transport. Further, X-6 always strives to be current with the best

cross-section data such as ENDF/B-V and local sets generated by Group T-2 at Los Alamos.

D. MCRUN Flow

The basic flow of the MCRUN overlay for a coupled neutron/photon problem is as follows. MCRUN calls TRNSPT to run histories until output is required. A particle is started by calling HSTORY. In HSTORY, STARTP is called; the starting location in the random number sequence for the history is set and the number of the history NPS being run is incremented. The flag IPT is set for the type of particle being run, 1 for a neutron and 2 for a photon.

Next, the appropriate source routine is called. Source options are the standard fixed sources, the surface source, the criticality (KCODE) source, or a user-provided source in which case SOURCE is called. All of the parameters describing the particle are set in these source routines, including position, direction of flight, energy, weight, time, and starting cell (and possibly surface), by sampling the various distributions described on the source input control cards. Several checks are made at this time to verify that the particle is in the correct cell or on the correct surface, and directed toward the correct cell; then control is returned to STARTP.

At the next point in STARTP, some of the summary information is incremented (see Appendix E for an explanation of these arrays). Then the energy of the particle is checked against energy cutoff. The track is terminated (NTER=4) if it is below the energy cutoff and logged with other terminations. TALLYD is called to score any detector contributions, and then DXTRAN is called (if used in the problem) to create particles on the spheres which are saved with BANKIT for later tracking. The weight window game is played, with any additional particles from splitting put into the bank and any losses to Russian roulette terminated (NTER=7). Energy splitting or roulette is then done if required with the subsequent possibility of banking or termination (NTER=9).

Back in HSTORY, the actual particle transport is started. TRACK is called next to calculate the intersection of the particle trajectory with each bounding surface of the cell. The minimum positive distance DLS to the cell boundary indicates the next surface JSU the particle is heading toward. The distance to the nearest DXTRAN sphere DXL is calculated, as is the distance to time cutoff DTC. The cross sections for cell ICL are calculated using a binary table lookup in ACETOT for neutrons and in PHOTOT for photons. The total cross section is modified in EXTRAN by the exponential transformation if necessary. The distance PMF to the next collision is determined (if a forced collision is required, FORCOL is called and the uncollided part is banked). The track length D of the

particle in the cell is found as the minimum of the distance PMF to collision, the distance DLS to the surface JSU, the distance DXL to a DXTRAN sphere, or the distance DTC to time cutoff. TALLY then is called to increment any cell tallies. Some summary information is incremented. The particle's parameters (time, position, and energy) are then updated. If the particle's distance DXL to a DXTRAN sphere (of the same type as the current particle) is equal to the minimum track length D, the particle is terminated with NTER=10. If the particle exceeds the appropriate time cutoff, the track is terminated (NTER=5) and logged. If the particle was detected leaving a DXTRAN sphere, the DXTRAN flag IDX is set to zero and the weight cutoff game is played. The particle is either terminated to weight cutoff (NTER=6) or survives with an increased weight. Weight adjustments are then made for the exponential transformation.

If the minimum track length D is equal to the distance-to-surface crossing DLS, the particle crosses a surface and SURFAC is called to do any surface tallies (by calling TALLY) and process the particle through the surface into the next cell by calling NEWCEL. It is in SURFAC that reflecting surfaces, geometry splitting, Russian roulette (NTER=8), and loss to escape (NTER=1) are treated. For splitting, one bank entry of NPA particle tracks is made in BANKIT for an (NPA+1)-for-1 split. The bank is the IBNK array, and entries or retrievals are made with the PBL array (the bank operates strictly on a last-in, first-out basis). The history is continued by calling TRACK in HSTORY.

If the distance to collision PMF is less than the distance to surface DLS, the particle undergoes a collision. Everything about the collision is determined in COLIDN for neutrons and COLIDP for photons. COLIDN determines which nuclide is involved in the collision, generates and banks any photons, handles analog capture (NTER=3) or capture by weight reduction, plays the weight cutoff game (NTER=6), and handles thermal collisions and nonthermal collisions such as elastic or inelastic scattering. Any additional tracks generated in the collision are put in the bank. ACECAS and ACECOS determine the energies and directions of particles exiting the collision. COLIDP for photons is similar to COLIDN, but it covers the simple or the detailed physics treatments. (see Section 2.IV.D on page 74).

At this point, the history of a viable particle is continued by going back to the beginning of the actual transport of HSTORY where the distance to cell boundary is calculated. Otherwise, the bank is checked for any remaining progeny, and if none exists, the history is terminated. Appropriate summary information is incremented, the tallies of this particular history are added to the total tally data by TALSHF, and a return is made to TRNSPT.

In TRNSPT, checks are made to see if output is required or if the job should be terminated because enough histories have been run or too little

time remains to continue. For continuation, HSTORY is called again. Otherwise a return is made to MCRUN. MCRUN calls OUTPUT, which calls SUMARY to print the summary information. Then SUMARY calls TALLYP to print the tally data. Appendix E defines all of the MCNP variables that are in COMMON as well as detailed descriptions of some important arrays.

II. GEOMETRY

The basic MCNP geometry concepts, discussed in Chapter 1, include the sense of a cell, the intersection and union operators, and surface specification. Covered in this section are the complement operator; an explanation of two surfaces, the cone and the torus; and a description of ambiguity and reflecting surfaces.

A. Complement Operator

This operator provides no new capability over the intersection and union operators; it is just a shorthand cell-specifying method that implicitly uses the intersection and union operators.

The symbol # is the complement operator and can be thought of as standing for ***not in***. There are two basic uses of the operator:

> #*n* means that the description of the current cell is the complement of the description of cell *n*.
>
> #(...) means complement the portion of the cell description in the parentheses (usually just a list of surfaces describing another cell).

In the first of the two above forms, MCNP performs five operations: (1) the symbol # is removed, (2) parentheses are placed around *n*, (3) any intersections in *n* become unions, (4) any unions in *n* are replaced by back-to-back parentheses)(which is an intersection, and (5) the senses of the surfaces defining *n* are reversed.

A simple example is a cube. We define a two-cell geometry with six surfaces, where cell 1 is the cube and cell 2 is the outside world:

```
1    0 -1  2 -3  4 -5  6
2    0  1:-2:  3:-4:  5:-6
```

Note that cell 2 is everything in the universe that is ***not in*** cell 1, or

```
2    0  #1
```

The form #(n) is not allowed; it is functionally available as the equivalent of $-n$.

CAUTION: Using the complement operator can destroy some of the necessary conditions for some cell volume and surface area calculations by MCNP. See page 286 for an example of this.

The complement operator can be easily abused if it is used indiscriminately. A simple example can best illustrate the problems. Fig. 2.1 consists of two concentric spheres inside a box. Cell 4 can be described using the complement operator as

```
4    0  #3 #2 #1
```

Although cells 1 and 2 do not touch cell 4, to omit them would be incorrect. If they were omitted, the description of cell 4 would be everything in the universe that is not in cell 3. Since cells 1 and 2 are not part of cell 3, they would be included in cell 4. Even though surfaces

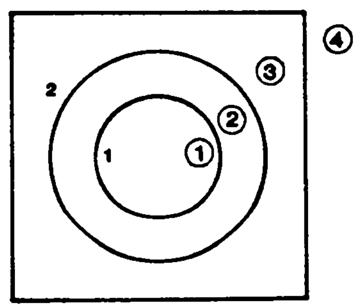


Figure 2.1

1 and 2 do not physically bound cell 4, using the complement operator as in this example causes MCNP to think that all surfaces involved with the complement do bound the cell. Even though this specification is correct and required by MCNP, the disadvantage is that when a particle enters cell 4 or has a collision in cell 4, MCNP must calculate the intersection of the particle's trajectory with all real bounding surfaces of cell 4 plus any extraneous ones brought in by the complement operator. This intersection calculation is very expensive and can add significantly to the required computer time.

A better description of cell 4 would be to complement the description of cell 3 (omitting surface 2) by reversing the senses and interchanging union and intersection operators as illustrated in the cell cards that describe the simple cube in the preceding paragraphs.

B. Surfaces

Two surfaces, the cone and torus, require more explanation. The general equation for a cone describes a cone of two sheets (just like a hyperboloid of two sheets) - one sheet is a cone of positive slope, and the other has a negative slope. A cell whose description contains a two-sheeted cone may require an ambiguity surface to distinguish between the two sheets. MCNP provides the option to select either of the two sheets; this option frequently simplifies geometry setups and eliminates any ambiguity. The +1 or the −1 entry on the cone surface card causes the one sheet cone treatment to be used. If the sign of the entry is positive, the specified sheet is the one that extends to infinity in the positive direction of the coordinate axis to which the cone axis is parallel. The converse is true for a negative entry. This feature is available only for cones whose axes are parallel to the coordinate axes of the problem.

The treatment of fourth degree surfaces in Monte Carlo calculations has always been difficult because of the resulting fourth order polynomial ("quartic") equations. These equations must be solved to find the intersection of a line of flight of a particle with a toroidal surface. In MCNP these equations must also be solved to find the intersection of surfaces to compute the volumes and surface areas of geometric regions of a given problem. In either case, the quartic equation,

$$x^4 + Bx^3 + Cx^2 + Dx + E = 0$$

is difficult to solve on a computer because of roundoff errors. For many years the MCNP toroidal treatment required 30 decimal digits (CDC double-precision) accuracy to solve quartic equations. Even then there were roundoff errors that had to be corrected by Newton-Raphson iterations. Schemes using a single-precision quartic formula solver followed by a Newton-Raphson iteration were inadequate because if the initial guess of roots supplied to the Newton-Raphson iteration is too inaccurate, the iteration will often diverge when the roots are close together.

The single-precision quartic algorithm in MCNP basically follows the quartic solution of Cashwell and Everett.[20] When roots of the quartic equation are well separated, a modified Newton-Raphson iteration quickly achieves convergence. But the key to this method is that if the roots are double roots or very close together, they are simply thrown out because a double root corresponds to a particle's trajectory being tangent to a toroidal surface, and it is a very good approximation to assume that the particle then has no contact with the toroidal surface. In extraordinarily rare cases where this is not a good assumption, the particle would become "lost." Additional refinements to the quartic solver include a carefully selected finite size of zero, the use of a cubic rather than a quartic

equation solver whenever a particle is transported from the surface of a torus, and a gross quartic coefficient check to ascertain the existence of any real positive roots. As a result, the single-precision quartic solver is substantially faster than double-precision schemes and also somewhat more accurate.

In MCNP, elliptical tori symmetric about any axis parallel to a coordinate axis may be specified. The volume and surface area of various tallying segments of a torus usually will be calculated automatically.

1. Ambiguity Surfaces

The description of the geometry of a cell must eliminate any ambiguities as to which region of space is included in the cell. That is, a particle entering a cell should be able to uniquely determine which cell it is in from the senses of the bounding surfaces. This is not possible in a a geometry such as shown in Fig. 2.2 unless an *ambiguity surface* is specified. Suppose the figure is rotationally symmetric about the y-axis.

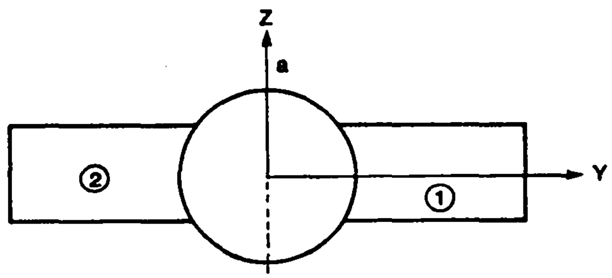


Figure 2.2

A particle entering cell 2 from the inner spherical region might think it was entering cell 1 because a test of the senses of its coordinates would satisfy the description of cell 1 as well as that of cell 2. In such cases, an ambiguity surface is introduced such as *a*, the plane y = 0. An ambiguity surface need not be a bounding surface of a cell, but it may be and frequently is. It can also be the bounding surface of some cell other than the one in question. However, the surface must be listed among those in the problem and must not be a reflecting surface (see page 44). The description of cells 1 and 2 in Fig. 2.2 is augmented by listing for each its sense relative to surface *a* as well as that of each of its other bounding surfaces. A particle in cell 1 cannot have the same sense relative to surface *a* as does a particle in cell 2. More than one ambiguity surface may be required to define a particular cell.

A second example may help to clarify the significance of ambiguity surfaces. We would like to describe the geometry of Fig. 2.3a. Without

the use of an ambiguity surface, the result will be Fig. 2.3b. Surfaces 1

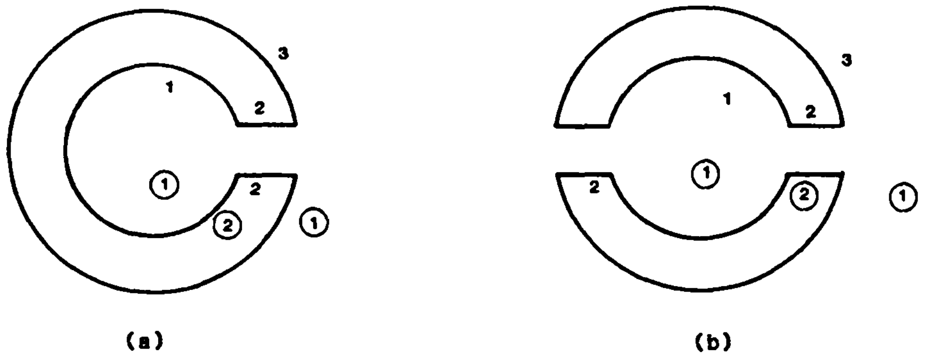


Figure 2.3

and 3 are spheres about the origin, and surface 2 is a cylinder around the y-axis. Cell 1 is both the center and outside world of the geometry connected by the region interior to surface 2.

At first glance it may appear that cell 1 can easily be specified by −1 : −2 : 3 whereas cell 2 is simply #1. This results in Figure 2.3b, in which cell 1 is everything in the universe interior to surface 1 plus everything in the universe interior to surface 2 (remember the cylinder goes to plus and minus infinity) plus everything in the universe exterior to surface 3.

An ambiguity surface (a plane at y=0) will solve the problem. Everything in the universe to the right of the ambiguity surface (call it surface 4) *intersected* with everything in the universe interior to the cylinder is a cylindrical region that goes to plus infinity but terminates at y=0. Therefore, −1 : (4 −2) : 3 defines cell 1 as desired in Figure 2.3a. The parentheses in this last expression are not required because intersections are done before unions. Another expression for cell 2 rather than #1 is 1 −3 #(4 −2).

For the user, ambiguity surfaces are specified the same way as any other surface − simply list the signed surface number as an entry on the cell card. For MCNP, if a particular ambiguity surface appears on cell cards with only one sense, it is treated as a true ambiguity surface. Otherwise, it still functions as an ambiguity surface but the TRACK subroutine will try to find intersections with it, thereby using a little more computer time.

2. Reflecting Surfaces

Any surface (except an ambiguity surface) may be designated a reflecting surface by preceding its number on the surface card with an

asterisk. Any particle hitting a reflecting surface is specularly (mirror) reflected. Reflecting planes are valuable because they can simplify a geometry setup (and also tracking) in a problem. They can, however, make it difficult (or even impossible) to get the correct answer. The user is cautioned to check the source weight and tallies to ensure that the desired result is achieved. Any tally in a problem with reflecting planes should have the same expected result as the tally in the same problem without reflecting planes. Detectors or DXTRAN used with reflecting surfaces usually give wrong answers (see page 96).

The following example will illustrate the above points and we hope make you very cautious in the use of reflecting surfaces; they should never be used in any situation without a lot of thought.

Consider a cube of carbon 10 cm on a side sitting on top of a 5-MeV neutron source distributed uniformly in volume. The source cell is a 1-cm-thick void completely covering the bottom of the carbon cube and no more. The average neutron flux across any one of the sides (but not top or bottom) is calculated to be 0.150 ($\pm$0.5%) per cm^2 per starting neutron from an MCNP F2 tally, and the flux at a point at the center of the same side is 1.55E-03 n/cm^2 ($\pm$1%) from an MCNP F5 tally.

The cube can be modeled by half a cube and a reflecting surface. All dimensions remain the same except the distance from the tally surface to the opposite surface (which becomes the reflecting surface) is 5 cm. The source cell is cut in half also. Without any source normalization, the flux across the surface is now 0.302 ($\pm$0.5%), which is twice the flux in the nonreflecting geometry. The detector flux is 2.58E-03 ($\pm$1%), which is *less* than twice the point detector flux in the nonreflecting problem.

The problem is that for the surface tally to be correct, the starting weight of the source particles has to be normalized; it should be half the weight of the nonreflected source particles. The detector results will always be wrong (and lower) for the reason discussed on page 96.

In this particular example, the normalization factor for the starting weight of source particles should be 0.5 because the source volume is half of the original volume. Without the normalization, the full weight of source particles is started in only half the volume. These normalization factors are problem dependent and should be derived very carefully.

Another way to view this problem is that the tally surface has doubled because of the reflecting surface; two scores are being made across the tally surface when one is made across each of two opposite surfaces in the nonreflecting problem. The detector has doubled, too - except that the contributions to it from beyond the reflecting surface are not being made, as explained on page 96.

III. CROSS SECTIONS

The MCNP code package is incomplete without the associated nuclear data tables. The kinds of tables available and their general features are outlined in this section. The manner in which information contained on nuclear data tables is used in MCNP is described in Sec. IV of this chapter.

There are two broad objectives in preparing nuclear data tables for MCNP. First, it is our responsibility to ensure that the data available to MCNP reproduce the original evaluated data as much as is practical. Second, new data should be brought into the MCNP package in a timely fashion, thereby giving users access to the most recent evaluations.

Five classes of nuclear data tables exist for MCNP. They are: (1) continuous-energy neutron interaction data, (2) discrete reaction neutron interaction data, (3) photon interaction data, (4) neutron dosimetry cross sections, and (5) neutron $S(\alpha,\beta)$ thermal data. It is understood that photon data are atomic rather than nuclear. In Mode N and Mode N P problems, one continuous-energy or discrete-reaction neutron interaction table is required for each isotope or element in the problem. Likewise, one photon interaction table is required for each element in a Mode N P or Mode P problem. Dosimetry and thermal data are optional. Cross sections from dosimetry tables may be used as response functions with the FM card to determine reaction rates. Thermal $S(\alpha,\beta)$ tables may be appropriate if the neutrons are transported at sufficiently low energies where molecular binding effects are important.

MCNP may read from data tables in any of three formats. Data tables are transmitted between computer installations in 80-column card-image BCD format (Type-1 format). An auxiliary processing code, MAKXSF, will convert the BCD files to standard unformatted binary files (Type-2 format) allowing more economical access during execution of MCNP. At Los Alamos (and on the MFE system) a third type of MCNP formatted data table is available - ACE (A Compact ENDF) tables (Type-3 format). The data contained on a table for a specific ZAID (10-character name for a nuclear data table) are independent of the format of the table.

The format of nuclear data tables is given in considerable detail in Appendix F. This appendix may be useful for users making extensive modifications to MCNP involving cross sections or for users debugging MCNP at a fairly high level.

The available nuclear data tables are listed in Appendix G. Each nuclear data table is identified by a ZAID. The general form of a ZAID is ZZZAAA.nnX, where ZZZ is the atomic weight, AAA is the atomic number, nn is the evaluation identifier, and X indicates the class of data. For elemental evaluations AAA=000. Nuclear data tables are selected by the user with the Mn and MTn cards.

In the remainder of this section we describe several characteristics of each class of data such as evaluated sources, processing tools, and any differences between data on the original evaluation and on the MCNP data tables. The means of accessing each class of data through MCNP input will be detailed and some hints will be provided on how to select the appropriate data tables.

A. Neutron Interaction Data: Continuous-Energy and Discrete Reaction

In neutron problems, one neutron interaction table is required for each isotope or element in the problem. The form of the ZAIDs is ZZZAAA.nnC for a continuous-energy table and ZZZAAA.nnD for a discrete reaction table. The neutron interaction tables available to MCNP are listed in Table G.2 of Appendix G. (It should be noted that although all nuclear data tables in Appendix G are available to users at Los Alamos, users at other installations will generally have only a subset of the tables available.)

For most materials there are many cross-section sets available (represented by different values of nn in the ZAIDs) because of multiple sources of evaluated data and different parameters used in processing the data. An evaluated nuclear data set is produced by analyzing experimentally measured cross sections and combining those data with the predictions of nuclear model calculations in an attempt to extract the most accurate cross-section information. Preparing evaluated cross-section sets has become a discipline in itself and has developed since the early 1960s. People in most of the national laboratories and several of the commercial reactor design firms are involved in such work. American evaluators joined forces in the mid-1960s to create the national ENDF system.[3] The ENDF effort has contributors from the thermal and fast reactor programs, as well as the controlled thermonuclear reactor program, all of whom collaborate through the Cross Section Evaluation Working Group (CSEWG).

In recent years the primary evaluated source of neutron interaction data for MCNP has been the ENDF/B system. As this manual is written, the latest version is ENDF/B-V Revision 2. It is anticipated that ENDF/B-VI will be released later in the 1980s. Recently evaluated neutron interaction data tables are also extracted from two other sources: Lawrence Livermore National Laboratory's Evaluated Nuclear Data Library[4] (ENDL), and supplemental evaluations performed in the Applied Nuclear Science Group at Los Alamos.[6-8] Older evaluations come from previous versions of ENDF/B and ENDL, the Los Alamos Master Data File,[21] and the Atomic Weapons Research Establishment in Great Britain.

MCNP does not access evaluated data directly; these data must first be processed into ACE format. The very complex processing codes used for this

purpose include NJOY[9] for evaluated data in ENDF/B format and MCPOINT[22] for ENDL data.

Data on the MCNP neutron interaction tables include cross sections and much more. Cross sections for all reactions given in the evaluated data are specified. For a particular table, the cross sections for each reaction are given on one energy grid, which is sufficiently dense that linear-linear interpolation between points reproduces the evaluated cross sections within a specified tolerance that is generally 1% or less. Depending primarily on the number of resolved resonances for each isotope, the resulting energy grid may contain as few as ~250 points (for example, H-1) or as many as ~22,500 points (for example, the ENDF/B-V version of Au-197). Other information, including the total absorption cross section, the total photon production cross section, and the average heating number (for energy deposition calculations), is also tabulated on the same energy grid.

Angular distributions of scattered neutrons are included in the neutron interaction tables for all nonabsorption reactions. The distributions are given in the center-of-mass system for elastic scattering and discrete-level inelastic scattering, and in the laboratory system for all other inelastic reactions. Angular distributions are given on a reaction-dependent grid of incident neutron energies. These tables are sampled to conserve energy for many collisions but will not necessarily conserve energy for a single collision; that is, energy is conserved on average.

The sampled angle of scattering uniquely determines the secondary energy for elastic scattering and discrete-level inelastic scattering. For other inelastic reactions, energy distributions of the scattered neutrons are provided in the neutron interaction tables. As with angular distributions, the energy distributions are given on a reaction-dependent grid of incident neutron energies.

When evaluations contain data about secondary photon production, that information appears in the MCNP neutron interaction tables. Recently processed data sets contain photon production cross sections, photon angular distributions, and photon energy distributions for each neutron reaction that produces secondary photons. The information is given in a manner similar to that described in the last few paragraphs for neutron cross sections and secondary neutron distributions.

Other miscellaneous information on the neutron interaction tables includes the atomic weight ratio of the target nucleus, the Q-values of each reaction, and nubar (the average number of neutrons per fission) data for fissionable isotopes.

Approximations must be made when processing an evaluated data set into ACE format. As mentioned above, cross sections are reproduced only within a certain tolerance, which is generally very small; to decrease it further

would result in excessively large data tables. Evaluated angular distributions for secondary neutrons and photons are approximated on MCNP data tables by 32 equally probable cosine bins. This approximation is clearly necessary when contrasted to the alternative that might involve sampling from a 20th-order Legendre polynomial distribution. Secondary neutron energy distributions given in tabular form by evaluators are sometimes approximated on MCNP data tables by 32 equally probable energy bins. Older cross-section tables include a 30x20 matrix approximation of the secondary photon energy spectra (described on page 62). On the whole, the approximations are small, and MCNP neutron interaction data tables are extremely faithful representations of evaluated data.

Discrete-reaction tables are identical to continuous-energy tables except that in the discrete reaction tables all cross sections have been averaged into 262 groups. The averaging is done with a flat weighting function. This is not a multigroup representation; the cross sections are simply given as histograms rather than as continuous curves. The remaining data (angular distributions, energy distributions, nubar, etc.) are identical in discrete-reaction and continuous-energy tables. Discrete-reaction tables are provided primarily as a method of shrinking the required data storage to enhance the ability to run MCNP on small machines or in a time-sharing environment. The tables may also be useful for preliminary scoping studies or for isotopes that exist only in trace quantities in a problem. They are not, however, recommended as a substitute for the continuous-energy tables when performing final calculations, particularly for problems involving transport through the resonance region.

The matter of how to select the appropriate neutron interaction tables for your calculation will now be discussed. Multiple tables for the same isotope are differentiated by the "nn" portion of the ZAID. The easiest choice for the user, although by no means the recommended one, is not to enter the nn at all. This will force MCNP to select the default tables for you based simply on those tables that are found first in the directory file XSDIR. Default continuous-energy neutron interaction tables are accessed by entering ZZZAAA for the ZAID. Including a DRXS card in the input file will force MCNP to choose the default discrete reaction tables.

Careful users will want to think about what neutron interaction tables to choose. There is, unfortunately, no strict formula for guidance in choosing the tables. The following guidelines and observations are the best that can be offered:

1. Users should be aware of the differences between the ".50C" series of data tables and the ".51C" series. Both are derived from ENDF/B-V. The ".50C" series is the most faithful reproduction of the evaluated data. The ".51C" series, also called the "thinned" series, has been processed with a less rigid tolerance than the ".50C" series. As with discrete reaction

data tables, although by no means to the same extent, users should be careful when using the "thinned" data for transport through the resonance region.

2. Consider differences in evaluators' philosophies. The Physical Data Group at Livermore is justly proud of its extensive cross-section efforts; their evaluations manifest a philosophy of reproducing the data with the fewest number of points. Livermore evaluations are available mainly in the ".31C" series. We at Los Alamos are particularly proud of the evaluation work being carried out in the Applied Nuclear Science Group T-2; generally, these new evaluations are the most complex because they are the most thorough. Recent evaluations from Los Alamos are available in the ".55C" series.

3. Be aware of the neutron energy spectrum in your problem. For high-energy problems, the "thinned" and discrete reaction data are probably not bad approximations. Conversely, it is essential to use the most detailed continuous-energy set available for problems influenced strongly by transport through the resonance region.

4. Check the temperature at which various data tables have been processed. Do not use a set that is Doppler broadened to 12000000K for a room temperature calculation.

5. The careful user might want to check the sensitivity of his results to various sets of nuclear data. Try, for example, a calculation with ENDF/B-V cross sections, and then another with ENDL cross sections. If the results of a problem are extremely sensitive to the choice of nuclear data, it is advisable to find out why.

6. For a coupled neutron/photon problem, be careful that the tables you choose have photon production data available. If possible, use the more-recent sets that have been processed into expanded photon production format.

7. Frequently, data tables are recommended that are not the defaults. This leads to questions about what is wrong with the defaults. The answer is that nothing is wrong with them. They are frequently the very best evaluated data sets we have to offer. In other cases they are not, because of size limitations imposed on the default library at Los Alamos. In still other cases, a nondefault data table may be appropriate for one or more of the reasons given in the previous paragraphs.

8. Usually, use the best data you can afford. It is understood that the latest evaluations tend to be more complex and therefore require more memory and longer execution times. If you are limited by available memory, try to use smaller data tables such as thinned or discrete reaction for the minor isotopes in the calculation. Discrete reaction data tables might be used for a parameter study, followed by a calculation with the full continuous-energy data tables for confirmation.

To select the neutron interaction data tables, the nn portion of the ZAIDs must be entered on the Mn card(s). For a continuous-energy set, ZZZAAA.nn is equivalent to ZZZAAA.nnC. To use a discrete reaction table (unless there is a DRXS card in the input) the full ZAID, ZZZAAA.nnD, must be entered.

In conclusion, the additional time necessary to choose appropriate neutron interaction data tables rather than simply to accept the defaults will often be well worth it in gaining understanding of your calculation.

B. Photon Interaction Data

Photon interaction cross sections are required for all photon problems. The form of the ZAIDs is ZZZ000.nnP. Presently, nn=01 only. For a neutron-induced photon problem, enter ZZZAAA.nnC for the neutrons and ZZZ000.01P will automatically be used for the photons.

The photon interaction tables are based on evaluated data from ENDF[23] for all elements from Z=1 through Z=94 with the exception of Z=84,85,87,88,89, and 93. Data for these six elements come from the compilation of Storm and Israel.[24] Fluorescence data are taken from work by Everett and Cashwell.[25]

Cross sections as a function of energy are given on photon interaction tables for coherent scattering, incoherent scattering, pair production, and the photoelectric effect. Energy grids are tailored specifically for each element and contain ~40–60 points. Log–log interpolation is employed to determine cross sections between adjacent energies. Heating numbers are tabulated on the same energy grid as the cross sections.

The determination of directions and energies of scattered photons requires information different from the sets of angular and energy distributions found on neutron interaction tables. Angular distributions of secondary photons are introduced through the differential Thomson cross section modified by form factors for coherent scattering and the differential Klein–Nishina cross section modified by scattering functions for incoherent scattering. Form factors and scattering functions are tabulated as a function of momentum transfer on the MCNP photon interaction tables. The energy of an incoherently scattered photon is calculated from the sampled scattering angle. Values of the integrated coherent form factor are tabulated on the photon interaction tables for use in point detector routines.

Very few approximations are made in the various processing codes used to transfer photon data from ENDF into the format of MCNP photon interaction tables. Cross sections are reproduced exactly as given. Form factors and scattering functions are reproduced as given; however, the momentum transfer grid on which they are tabulated may be different from that of the original evaluation. Heating numbers are calculated values,

not given in evaluated sets, but inferred therefrom. Fluorescence data are not provided in ENDF; therefore the data for MCNP are extracted from a variety of sources as described in Ref. 24.

The process for selecting photon interaction tables is much simpler than that for selecting neutron interaction tables, primarily because currently only one table exists for each element. There is no user input to select photon interaction tables in a coupled neutron/photon problem. A specification on the Mn card for a neutron interaction table with ZAID=ZZZAAA.nnC or ZAID=ZZZAAA.nnD immediately requires a photon interaction table with ZAID=ZZZ000.01P to be accessed as well. In a photon only problem, ZAIDs of photon interaction tables are entered on the Mn card(s). The following ZAIDs are all equivalent: ZZZ000, ZZZ000.01, and ZZZ000.01P. The data table required for ZAID=ZZZAAA.01P is identical to that required for ZAID=ZZZ000.01P; however, the atomic weight used in the calculation will likely be different.

C. Neutron Dosimetry Cross Sections

Dosimetry cross-section tables may be used in neutron problems, but they are never required because they have no effect on the particle transport of a problem. These incomplete cross-section sets may not be used for materials through which neutrons are transported; they do provide energy-dependent cross sections to MCNP for use as response functions with the FM tally feature. ZAIDs of dosimetry tables are of the form ZZZAAA.nnY.

The available dosimetry cross sections are from three sources: ENDF/B-V Dosimetry Tape 531, ENDF/B-V Activation Tape 532, and ACTL[5] – an evaluated neutron activation cross-section library from the Lawrence Livermore National Laboratory. Various codes have been used to process evaluated dosimetry data into the format of MCNP dosimetry tables.

Data on dosimetry tables are simply energy-cross-section pairs for one or more reactions. The energy grids for all reactions are independent of each other. Interpolation between adjacent energy points may be specified as histogram, linear-linear, linear-log, log-linear, or log-log. With the exception of the tolerance involved in any reconstruction of pointwise cross sections from resonance parameters, evaluated dosimetry cross sections may be reproduced on the MCNP data tables with no approximation.

ZAIDs for dosimetry tables must be entered on material cards that are referenced by FM cards, and not by cell cards. The complete ZAID, ZZZAAA.nnY, must be given; there are no defaults for dosimetry tables.

D. Neutron Thermal $S(\alpha,\beta)$ Tables

Thermal $S(\alpha,\beta)$ tables are never required. They may be used in any neutron problem and should be used in such problems involving neutron thermalization. Thermal tables have ZAIDs of the form XXXXXX.nnT, where XXXXXX is a mnemonic character string. The data on these tables encompass those required for a complete representation of thermal neutron scattering by molecules and crystalline solids.

The source of $S(\alpha,\beta)$ data is a special set of ENDF tapes.[26] The THERMR and ACER modules of the NJOY[9] system have been used to process the evaluated thermal data into a format appropriate for MCNP.

Data on the current room temperature (300 K) thermal tables are for neutron energies less than 4 eV. Cross sections are tabulated on table-dependent energy grids; inelastic scattering cross sections are always given and elastic scattering cross sections are sometimes given. Correlated energy-angle distributions are provided for inelastically scattered neutrons. A set of equally probable final energies is tabulated for each of several initial energies. Further, a set of equally probable cosines or cosine bins is tabulated for each combination of initial and final energies. Elastic scattering data may be derived from either a coherent or incoherent approximation. In the incoherent case, equally probable cosines or cosine bins are tabulated for each of several incident neutron energies. In the coherent case, scattering cosines are determined from a set of Bragg energies derived from the lattice parameters. During processing, approximations to the evaluated data are made when constructing equally probable energy and cosine distributions.

ZAIDs for the thermal tables must be entered on an MTn card that corresponds to an existing Mn card. The thermal table will generally provide data for one component of a material; for example, hydrogen in light water. Thermal ZAIDs may be entered on the MTn card(s) as XXXXXX, XXXXXX.nn, or XXXXXX.nnT.

IV. PHYSICS

Two concepts are important in the following discussion for setting up the input and for understanding the output. These are particle weight and particle tracks.

A. Particle Weight

If MCNP were used only to simulate exactly physical transport, then each MCNP particle would represent one physical particle and would have

unit weight. However, for computational efficiency, MCNP allows many techniques that do not exactly simulate physical transport. For instance, each MCNP particle might represent a number w of particles emitted from a source. This number w is the initial weight of the MCNP particle. The w physical particles all would have different random walks, but the one MCNP particle representing these w physical particles will only have one random walk. Clearly this is not an exact simulation; however, the true number of physical particles is preserved in MCNP in the sense of statistical averages and therefore in the limit of large particle numbers (of course including particle production or loss if they occur). Each MCNP particle result is multiplied by the weight so that the full results of the w physical particles represented by each MCNP particle are exhibited in the final results (tallies). This procedure allows users to normalize their calculations to whatever source strength they desire. The default normalization is to weight one per MCNP particle. A second normalization to the number of Monte Carlo histories is made in the results so that the expected means will be independent of the number of source particles actually initiated in the MCNP calculation.

The utility of particle weight, however, goes far beyond simply normalizing the source. Every Monte Carlo biasing technique alters the probabilities of random walks executed by the particles. The purpose of such biasing techniques is to increase the number of particles that sample some part of the problem of special interest (1) without increasing (sometimes actually decreasing) the sampling of less interesting parts of the problem, and (2) without erroneously affecting the expected mean physical result (tally). This procedure, properly applied, increases precision in the desired result compared to an unbiased calculation taking the same computing time. For example, if an event is made $\sqrt{2}$ times as likely to occur (as it would occur without biasing), the tally ought to be multiplied by $1/\sqrt{2}$ so that the expected average tally is unaffected. This tally multiplication can be accomplished by multiplying the particle weight by $1/\sqrt{2}$ because the tally contribution by a particle is always multiplied by the particle weight in MCNP. Note that weights need not be integers.

In short, particle weight is a number carried along with each MCNP particle that represents that particle's relative contribution to the final tallies. Its magnitude is determined so as to ensure that whenever MCNP deviates from an exact simulation of the physics, the expected physical results are nonetheless preserved in the sense of statistical averages, and therefore in the limit of large MCNP particle numbers. Its utility is in the manipulation of the number of particles, sampling just a part of the problem to improve the precision of selected results without the need for a full unbiased calculation – with its added cost in computing time – to achieve the same results and precision.

B. Particle Tracks

When a particle starts out from a source, a particle track is created. If that track is split 2 for 1 at a splitting surface, a second track is created and there are now two tracks from the original source particle, each with half the single track weight. If one of the tracks has an (n,2n) reaction, one more track is started for a total of three. A track refers to each component of a source particle during its history. Some tallies use the length of a track in a given cell to determine a quantity of interest; others tally at each collision or surface crossing.

Within a given cell of fixed composition, the method of sampling a collision along the track is determined using the following theory. The probability of a first collision for a particle between l and l+dl along its line of flight is given by

$$p(l)dl = e^{-\Sigma_t l} \, \Sigma_t \, dl,$$

where Σ_t is the macroscopic total cross section of the medium and is interpreted as the probability per unit length of a collision. Setting ξ, the random number on [0,1), to be

$$\xi = \int_0^l e^{-\Sigma_t s} \, \Sigma_t ds = 1 - e^{-\Sigma_t l},$$

it follows that

$$l = -\frac{1}{\Sigma_t} \ln(1 - \xi).$$

But, because $1-\xi$ is distributed in the same manner as ξ and hence may be replaced by ξ, we obtain the well-known expression for the distance to collision,

$$l = -\frac{1}{\Sigma_t} \ln \xi.$$

C. *Neutron Interactions*

A collision between a neutron and an atom is affected by the thermal motion of the atom, and in most cases, the collision is also affected by the presence of other atoms nearby. The thermal motion cannot be ignored in many applications of MCNP without serious error. The effects of nearby atoms are also important in some applications. MCNP routinely uses a thermal treatment based on the free gas approximation to account for the thermal motion. It also has an explicit $S(\alpha,\beta)$ capability that takes into account the effects of chemical binding and crystal structure, but is available for only a limited number of substances at room temperature.

The regular thermal treatment in MCNP assumes that the medium is a free gas and also that in the range of atomic weight and neutron energy where thermal effects are significant, the elastic scattering cross section at zero temperature is nearly independent of the energy of the neutron, and the reaction cross sections are nearly independent of temperature. These assumptions allow MCNP to have a thermal treatment of neutron collisions that runs almost as fast as a completely nonthermal treatment and that is adequate for most practical problems.

With the above assumptions, the thermal treatment of cross sections, consists of raising the value of a zero-temperature elastic cross section by the factor

$$F = (1 + 0.5/a^2)\mathrm{erf}(a) + \exp(-a^2)/(a\sqrt{\pi}) \quad ,$$

where $a = \sqrt{AE/kT}$, A = atomic weight, E = neutron energy, and T = temperature. For speed, F is approximated by $F = 1 + 0.5/a^2$ when $a \geq 2$ and by linear interpolation in a table of 51 values of aF when $a < 2$. Both approximations have relative errors less than 0.0001. The total cross section is also increased by the amount of the increase in the elastic cross section.

The thermal treatment of cross sections is done partly in the setup of a problem and partly during the actual transport calculation. If all of the cells that contain a particular nuclide have the same temperature, constant in time, the cross sections for that nuclide are adjusted to that temperature during the setup so that the transport will run a little faster. Otherwise, the cross sections are reduced, if necessary, to zero temperature during the setup and the thermal adjustment is made when the cross sections are used. For speed, the thermal adjustment is omitted if the neutron energy is greater than 500kT/A. At that energy the adjustment of the elastic cross section would be less than 0.1%.

The other aspect of the thermal treatment consists of taking into account the velocity of the target nucleus when the kinematics of a collision are being calculated. The target velocity is sampled and subtracted from the velocity of the neutron to get the relative velocity.

The collision is sampled in the target-at-rest frame and the outgoing velocities are transformed to the laboratory frame by adding the target velocity.

When a particle (representing any number of neutrons, depending upon the particle weight) collides with a nucleus, the following sequence occurs:

1. the collision nuclide is identified;
2. unless the $S(\alpha,\beta)$ treatment is to be used, the velocity of the target nucleus is sampled;
3. photons are optionally generated for later transport;
4. neutron capture (that is, neutron disappearance by any process) is modeled;
5. unless the $S(\alpha,\beta)$ treatment is used, either elastic scattering or an inelastic reaction is selected, and the new energy and direction of the outgoing track(s) are determined.
6. if the energy of the neutron is low enough and an appropriate $S(\alpha,\beta)$ table is present, the collision is modeled by the $S(\alpha,\beta)$ treatment instead of by step 5.

1. Selection of Collision Nuclide

If there are n different nuclides forming the material in which the collision occurred, and if ξ is a random number on the unit interval [0,1), then the k^{th} nuclide is chosen as the collision nuclide if

$$\sum_{i=1}^{k-1} \Sigma_{t\,i} < \xi \sum_{i=1}^{n} \Sigma_{t\,i} \le \sum_{i=1}^{k} \Sigma_{t\,i} \, .$$

where Σ_{t_i} is the macroscopic total cross section of nuclide i. If the energy of the neutron is low enough and the appropriate $S(\alpha,\beta)$ table is present, the total cross section is the sum of the capture cross section from the regular cross-section table and the elastic and inelastic scattering cross sections from the $S(\alpha,\beta)$ table. Otherwise, the total cross section is taken from the regular cross-section table and is adjusted for thermal effects as described above.

2. Sampling the Velocity of the Target Nucleus

Unless the $S(\alpha,\beta)$ treatment is to be used, the velocity of the target nucleus is required. If the energy of the neutron is greater than 400 kT

and the target is not ^1H, the velocity of the target is set to zero. Otherwise, the target velocity is sampled as follows.

The free-gas kernel is a thermal interaction model that results in a good approximation to the thermal flux spectrum in a variety of applications and can be sampled without tables. The effective scattering cross section in the laboratory system for a neutron of kinetic energy E is

$$\sigma_s^{eff}(E) = \frac{1}{v_n}\int\int \sigma_s(v_{rel})v_{rel}p(V)dV\frac{d\mu_t}{2}. \quad (2.1)$$

Here, v_{rel} is the relative velocity between a neutron moving with a scalar velocity v_n and a target nucleus moving with a scalar velocity V, and μ_t is the cosine of the angle between the neutron and the target direction-of-flight vectors. The equation for v_{rel} is

$$v_{rel} = (v_n{}^2 + V^2 - 2v_nV\mu_T)^{1/2}.$$

The scattering cross section at the relative velocity is denoted by $\sigma_s(v_{rel})$, and p(V) is the probability density function for the Maxwellian distribution of target velocities,

$$p(V) = \frac{4}{\pi^{1/2}}\beta^3V^2e^{-\beta^2V^2}$$

with β defined as

$$\beta = \left(\frac{AM_n}{2kT}\right)^{1/2} .$$

where A is the mass of a target nucleus in units of the neutron mass, M_n is the neutron mass in MeV-sh^2/cm^2, and kT is the equilibrium temperature of the target nuclei in MeV.

The most probable scalar velocity V of the target nuclei is $1/\beta$, which corresponds to a kinetic energy of kT for the target nuclei. This is not the average kinetic energy of the nuclei, which is 3kT/2. The quantity that MCNP expects on the TMPn input card is kT and not just T (see page 255). Note that kT is not a function of the particle mass and is

therefore the kinetic energy at the most probable velocity for particles of any mass.

Equation (2.1) implies that the probability distribution for a target velocity V and cosine μ_t is

$$P(V,\mu_t) = \frac{\sigma_s(v_{rel})\ v_{rel}\ P(V)}{2\sigma_s^{eff}(E)v_n} .$$

It is assumed that the variation of $\sigma_s(v_{rel})$ with target velocity can be ignored. The justification for this approximation is that (1) for light nuclei, $\sigma_s(v_{rel})$ is slowly varying with velocity, and (2) for heavy nuclei, where $\sigma_s(v_{rel})$ may vary rapidly, the moderating effect of scattering is small so that the consequences of the approximation will be negligible. As a result of the approximation, the probability distribution actually employed is

$$P(V,\mu_t) \propto \sqrt{v_n^2 + V^2 - 2Vv_n\mu_t}\ \ V^2e^{-\beta^2V^2} .$$

Note that the above expression may be written as

$$P(V,\mu_t) \propto \frac{\sqrt{v_n^2 + V^2 - 2Vv_n\mu_t}}{v_n+V}\ (V^3e^{-\beta^2V^2} + v_nV^2e^{-\beta^2V^2}) .$$

As a consequence, the following algorithm is employed to sample the target velocity.

1. With probability $\alpha=1/(1+(\sqrt{\pi}\beta v_n/2))$, the target velocity V is sampled from the distribution $P_1(V)=2\beta^4V^3e^{-\beta^2V^2}$. The transformation $V=\sqrt{y}/\beta$ reduces this distribution to the sampling distribution for $P(y)=ye^{-y}$.
2. With probability $1-\alpha$, the target velocity is sampled from the distribution $P_2(V)=(4\beta^3/\sqrt{\pi})\ V^2e^{-\beta^2V^2}$. Substituting $V=y/\beta$ reduces the distribution to the sampling distribution for y: $P(y)=(4/\sqrt{\pi})\ y^2e^{-y^2}$.
3. The cosine of the angle between the neutron velocity and the target velocity is sampled uniformly on the interval $-1 \leq \mu_t \leq +1$.
4. The rejection function $R(V,\mu_t)$ is computed using

$$R(V,\mu_t) = \frac{\sqrt{v_n^2 + V^2 - 2Vv_n\mu_t}}{v_n+V} \leq 1.$$

With probability $R(V,\mu_t)$, the sampling is accepted; otherwise, the sampling is rejected and the procedure is repeated. The minimum efficiency of this rejection algorithm averaged over μ_t is 68% and approaches 100% as either the incident neutron energy approaches zero or becomes much larger than kT.

3. Optional Generation of Photons

Photons are generated if the problem being run is a combined neutron/photon run and if the collision nuclide has a nonzero photon production cross section. The number of photons produced is a function of neutron weight, neutron source weight, photon weight limits (entries on the PWT card), photon production cross section, neutron total cross section, cell importance, and the importance of the neutron source cell. No more than 10 photons may be born from any neutron collision.

Because of the many low-weight photons typically created by neutron collisions, Russian roulette is played for particles with weight below the bounds specified on the PWT card, resulting in fewer particles, each having a larger weight. The created photon weight before Russian roulette is

$$WGT_p = \frac{WGT_n\ \sigma_\gamma}{\sigma_T} ,$$

where WGT_p = photon weight
WGT_n = neutron weight
σ_γ = photon production cross section
σ_T = total neutron cross section.

This Russian roulette game is played according to neutron cell importances for the collision and source cell. For a photon produced in cell ICL where the minimum weight set on the PWT card is WM_{ICL}, let IN_{ICL} be the neutron importance in cell ICL and let IN_{SRC} be the neutron importance in the source cell. If $WGT_p > WM_{ICL}*IN_{SRC}/IN_{ICL}$, one or more photons will be produced. The number of photons created is N_p. $N_p=(WGT_p*IN_{ICL})/(5*WM_{ICL}*IN_{SRC})+1$. $N_p \leq 10$. Each photon is stored in the bank with weight WGT_p/N_p. If $WGT_p < WM_{ICL}*IN_{SRC}/IN_{ICL}$, Russian roulette is played and the photon survives with probability $WGT_p*IN_{ICL}/(WM_{ICL}*IN_{SRC})$ and is given the weight $WM_{ICL}*IN_{SRC}/IN_{ICL}$.

If weight windows are not used and if the weight of the starting neutrons is not unity, setting all the WM_{ICL} on the PWT card to a negative value will make the photon minimum weight relative to the neutron source weight. This will make the number of photons being created roughly proportional to the biased collision rate of neutrons. It is recommended for most applications that such a negative number be used and chosen to produce from one to four photons per source neutron. The default values for WM_{ICL} on the PWT card are −1, which should be adequate for most problems using cell importances.

If weight windows are used, the entries on the PWT card should be the same as on the WWN1:P card. If energy-dependent photon weight windows are used, the entries on the PWT card should be the minimum WWNn:P entry for each cell, where n refers to the photon weight window energy group. This will cause most photons to be born within the weight window bounds.

If weight windows are used, it is recommended that the values of WM_{ICL} on the PWT card be the same as the lower weight window bounds entered on the WWN card.

Any photons generated at neutron collision sites are temporarily stored in the bank. There are two methods for determining the exiting energies and directions, depending on the form in which the processed photon production data are stored in a library. The first method, used with recently processed evaluations, for example, ENDF/B-V and later, has the evaluated photon production data processed into an "expanded format."[27] In this format, energy-dependent cross sections, energy distributions, and angular distributions are explicitly provided for every photon producing neutron interaction. In the second method, used with data processed from older evaluations, the evaluated photon production data have been collapsed so that the only information about secondary photons is in a matrix of 20 equally probable photon energies for each of 30 incident neutron energy groups. The sampling techniques used in each method are now described.

a. Expanded Photon Production Method: In the expanded photon production method, the reaction n responsible for producing the photon is sampled from

$$\sum_{i=1}^{n-1} \sigma_i < \xi \sum_{i=1}^{N} \sigma_i \le \sum_{i=1}^{n} \sigma_i ,$$

where ξ is a random number on the interval [0,1), N is the number of photon production reactions, and σ_i is the photon production cross section for reaction i at the incident neutron energy. Note that there is no correlation between the sampling of the type of photon production reaction

and the sampling of the type of neutron reaction described in Sec. IV.C.5 of this chapter.

Just as every neutron reaction (for example, (n,2n)) has associated energy-dependent angular and energy distributions for the secondary neutrons, every photon production reaction (for example, (n,pγ)) has associated energy-dependent angular and energy distributions for the secondary photons. The photon distributions are sampled in much the same manner as their counterpart neutron distributions.

All nonisotropic secondary photon angular distributions are represented by 32 equiprobable cosine bins. The distributions are given at a number of incident neutron energies. All photon-scattering cosines are sampled in the laboratory system. The sampling procedure is identical to that described for secondary neutrons in Sec. IV.C.5.a of this chapter.

Secondary photon energy distributions are also a function of incident neutron energy. There are two representations of secondary photon energy distributions allowed in ENDF/B format: tabulated spectra and discrete (line) photons. Correspondingly, there are two laws used in MCNP for the determination of secondary photon energies. Law 4 is an exact representation of tabulated photon spectra. Law 2 is used for discrete photons. Both laws are described in Sec. IV.C.5.d of this chapter.

The expanded photon production method has clear advantages over the original 30 x 20 matrix method described below. In coupled neutron/photon problems, users should attempt to specify data sets that contain photon production data in expanded format. Such data sets are identified by "YES P(E)" entries in the GPD column in Table G.2 in Appendix G.

b. 30 x 20 Photon Production Method: For lack of better terminology, we will refer to the photon production data contained on older libraries as "30 x 20 photon production" data. In contrast to expanded photon production data, there is no information about individual photon production reactions in the 30 x 20 data.

The only secondary photon data are a 30 x 20 matrix of photon energies; that is, for each of 30 incident neutron energy groups there are 20 equally probable exiting photon energies. There is no information regarding secondary photon angular distributions; therefore, all photons are taken to be produced isotropically in the laboratory system.

There are several problems associated with 30 x 20 photon production data. The 30 x 20 matrix is an inadequate representation of the actual spectrum of photons produced. In particular, discrete photon lines are not well represented, and the high-energy tail of a photon continuum energy distribution is not well sampled. Also, the multigroup representation is not consistent with the continuous-energy nature of MCNP. Finally, not all photons should be produced isotropically. None of these problems exist for data processed into the expanded photon production format.

4. Capture

Capture is treated in one of two ways: analog or implicit. In analog capture, the particle is killed with probability σ_a/σ_T, where σ_a and σ_T are the absorption and total cross sections of the collision nuclide at the incoming neutron energy. The absorption cross section is specially defined for MCNP as the sum of all (n,x) cross sections where x is anything except neutrons. Thus σ_a is the sum of $\sigma_{n,\gamma}$, $\sigma_{n,\alpha}$, $\sigma_{n,d}$, ... etc.

For implicit capture, the neutron weight WGT_n is reduced to WGT_n' as follows:

$$WGT_n' = \left(1 - \frac{\sigma_a}{\sigma_T}\right) WGT_n .$$

If the new weight, WGT_n', is below the problem weight cutoff (specified on the CUT card), Russian roulette is played, resulting overall in fewer particles with larger weight.

For all particles killed by analog capture, the entire particle energy and weight are deposited in the collision cell. For implicit capture, a fraction σ_a/σ_T of the incident particle weight and energy is deposited in the collision cell corresponding to that portion of the particle that was captured. This is the default method of neutron capture in MCNP.

Implicit capture also can be done continuously along the flight path of a particle trajectory as is the common practice in astrophysics. In this case, the distance to scatter, rather than the distance to collision, is sampled. The distance to scatter is

$$l = -\frac{1}{\Sigma_s} \ln(1 - \xi).$$

The particle weight at the scattering point is reduced by the capture loss, $WGT' = \exp(-\Sigma_a * l) * WGT$.

Implicit capture along a flight path is a special form of the exponential transformation coupled with implicit capture at collisions. The path length is stretched in the direction of the particle, $\mu = 1$, and the stretching parameter is $p = \Sigma_a/\Sigma_t$. Using these values the exponential transform and implicit capture at collisions yield the identical equations as implicit capture along a flight path.

Implicit capture along a flight path is invoked in MCNP as a special option of the exponential transform variance reduction method. It is most useful in highly absorbing media, that is, Σ_a/Σ_t approaches 1. When almost every collision results in capture, it is very inefficient to sample

distance to collision. However, explicit capture along a flight path is discouraged. In highly absorbing media, there is usually a superior set of exponential transform parameters. In relatively nonabsorbing media, it is better to sample the distance to collision than the distance to scatter.

5. Elastic and Inelastic Scattering

If the conditions for the $S(\alpha,\beta)$ treatment are not met, the particle undergoes either an elastic or inelastic collision. The selection of an elastic collision is made with probability

$$\frac{\sigma_{el}}{\sigma_{in} + \sigma_{el}} = \frac{\sigma_{el}}{\sigma_T - \sigma_a} .$$

where σ_{el} is the elastic scattering cross section.
σ_{in} is the inelastic cross section; includes any neutron-out process – (n,n′),(n,f),(n,np),etc.
σ_a is the absorption cross section; $\Sigma\sigma(n,x), x\neq n$, that is, all neutron disappearing reactions.
σ_T is the total cross section, $\sigma_T=\sigma_{el}+\sigma_{in}+\sigma_a$.

The selection of an inelastic collision is made with the remaining probability

$$\frac{\sigma_{in}}{\sigma_T - \sigma_a} .$$

If the collision is determined to be inelastic, the type of inelastic reaction, n, is sampled from

$$\sum_{i=1}^{n-1} \sigma_i < \xi \sum_{i=1}^{N} \sigma_i \leq \sum_{i=1}^{n} \sigma_i .$$

where ξ is a random number on the interval [0,1), N is the number of inelastic reactions, and the σ_i's are the inelastic reaction cross sections at the incident neutron energy.

For both elastic and inelastic scattering, the direction of exiting particles is usually determined by sampling angular distribution tables from the cross-section files. This process will be described shortly. For elastic collisions and discrete inelastic scattering from levels, the exiting particle energy is determined from two body kinematics based upon the center-of-mass cosine of the scattering angle. For other inelastic

processes, the energy of exiting particles is determined from secondary energy distribution laws from the cross-section files, which vary according to the particular inelastic collision modeled.

a. Sampling of Angular Distributions:. The direction of emitted particles is sampled in the same way for most elastic and inelastic collisions. The cosine of the angle between incident and exiting particle directions, μ, is sampled from angular distribution tables in the collision nuclide's cross-section library. The angular distribution tables consist of 32 equiprobable cosine bins and are given at a number of different incident neutron energies. The cosines are either in the center-of-mass or target-at-rest system, depending on the type of reaction. If E is the incident neutron energy, if E_n is the energy of table n, and if E_{n+1} is the energy of table n + 1, then a value of μ is sampled from table n + 1 with probability $(E - E_n)/(E_{n+1} - E_n)$ and from table n with probability $(E_{n+1} - E)/(E_{n+1} - E_n)$. A random number ξ on the interval [0,1) is then used to select the i^{th} cosine bin such that $i - 1 < 32\,\xi \leq i$. The value of μ is then computed as

$$\mu = \mu_i + (32\,\xi - i)(\mu_{i+1} - \mu_i) \quad .$$

If, for some incident neutron energy, the emitted angular distribution is isotropic, μ is chosen from $\mu=\xi$, where ξ is a random number on the interval [−1,1). (Strictly, in MCNP random numbers are always furnished on the interval [0,1). Thus, to compute ξ on [−1,1) we calculate ξ=2.*R−1., where R is a random number on [0,1).)

For elastic scattering and inelastic level scattering, the scattering cosine is chosen in the center-of-mass system. Conversion must then be made to μ_{lab}, the cosine in the target-at-rest system. For other inelastic reactions, the scattering cosine is sampled directly in the target-at-rest system.

The incident particle direction cosines, (u_o,v_o,w_o), are rotated to new outgoing target-at-rest system cosines, (u,v,w), through a polar angle whose cosine is μ_{lab}, and through an azimuthal angle sampled uniformly. For random numbers ξ_1 and ξ_2 on the interval [−1,1) with rejection criterion $\xi_1^2 + \xi_2^2 \leq 1$, the rotation scheme is

$$u = u_o\mu_{lab} + \frac{\sqrt{1-\mu_{lab}^2}\,(\xi_1\, u_o w_o - \xi_2 v_o)}{\sqrt{(\xi_1^2 + \xi_2^2)(1 - w_o^2)}}$$

$$v = v_o\mu_{lab} + \frac{\sqrt{1-\mu_{lab}^2}\,(\xi_1\, v_o w_o + \xi_2 u_o)}{\sqrt{(\xi_1^2 + \xi_2^2)(1 - w_o^2)}}$$

$$w = w_o \mu_{lab} - \frac{\xi_1 \sqrt{(1 - \mu_{lab}^2)(1 - w_o^2)}}{\sqrt{(\xi_1^2 + \xi_2^2)}} .$$

If $1 - w_o^2 \sim 0$, then

$$u = \xi_1 \sqrt{\frac{1 - \mu_{lab}^2}{\xi_1^2 + \xi_2^2}}$$

$$v = \xi_2 \sqrt{\frac{1 - \mu_{lab}^2}{\xi_1^2 + \xi_2^2}}$$

$$w = \mu_{lab} .$$

If the scattering distribution is isotropic in the target-at-rest system, it is possible to use an even simpler formulation that takes advantage of the exiting direction cosines, (u,v,w), being independent of the incident direction cosines, (u_o,v_o,w_o). In this case,

$$u = 2\xi_1^2 + 2\xi_2^2 - 1$$

$$v = \xi_1 \sqrt{\frac{1 - u^2}{\xi_1^2 + \xi_2^2}}$$

$$w = \xi_2 \sqrt{\frac{1 - u^2}{\xi_1^2 + \xi_2^2}} .$$

b. Elastic Scattering: The particle direction is sampled from the appropriate angular distribution tables, and the exiting energy, E_{out}, is dictated by two-body kinematics:

$$E_{out} = \frac{1}{2} E_{in}\left[(1 - \alpha)\mu_{cm} + 1 + \alpha\right]$$

$$= E_{in}\left[\frac{1 + A^2 + 2A\,\mu_{cm}}{(1 + A)^2}\right] \quad ,$$

where E_{in} = incident neutron energy
μ_{cm} = center-of-mass cosine of the angle between incident and exiting particle directions

$$\alpha = \left(\frac{A - 1}{A + 1}\right)^2$$

and A = mass of collision nuclide in units of the mass of a neutron (atomic weight ratio)

c. Inelastic Scattering: The treatment of inelastic scattering depends upon the particular inelastic reaction chosen. Inelastic reactions are defined as (n,y) reactions such as (n,n′), (n,2n), (n,f), (n,n′α) in which y includes at least one neutron.

For many inelastic reactions, such as (n,2n), more than one neutron may be emitted for each incident neutron. The weight of each exiting particle is always the same as the weight of the incident particle minus any implicit capture. The energy of exiting particles is governed by various scattering laws that are sampled independently for each exiting particle from the cross-section files. Which law is used is prescribed by the particular cross-section evaluation used. In fact, more than one law may be specified, and the particular one used at a particular time is decided with a random number. In an (n,2n) reaction, for example, the first particle emitted may have an energy sampled from one or more laws, but the second particle emitted may have an energy sampled from one or more different laws, depending upon specifications in the nuclear data library. Because each emerging energy and scattering angle is sampled independently, there is no correlation between the emerging particles. Hence energy is not conserved in an individual reaction because, for example, a 14-MeV particle could conceivably produce two 12-MeV particles in a single reaction. But the net effect of many particle histories is unbiased because on the average the correct amount of energy is emitted. Results are biased only when quantities that depend upon the correlation between the emerging particles are being estimated.

Users should note that MCNP follows a very particular convention. The exiting particle energy and direction are always given in the target-at-rest coordinate system. For the kinematical calculations in MCNP to be performed correctly, the angular distributions for elastic and discrete inelastic level scattering *must* be given in the center-of-mass

system, and the angular distributions for all other reactions *must* be given in the target-at-rest system. MCNP does not stop if this convention is not adhered to, but the results will be erroneous. In the checking of the cross-section libraries prepared for MCNP at Los Alamos, however, careful attention has been paid to ensure that these conventions are followed.

d. Nonfission Inelastic Scattering and Emission Laws: Nonfission inelastic reactions are handled differently from fission inelastic reactions. For each nonfission reaction N_p particles are emitted, where N_p is an integer quantity specified for each reaction in the cross-section data library of the collision nuclide. The direction of each emitted particle is independently sampled from the appropriate angular distribution table, as was described earlier. The energy of each emitted particle is independently sampled from one of the following scattering or emission laws. For completeness and convenience we list all the laws together, regardless of whether the law is appropriate for nonfission inelastic scattering, (for example, Law 3), fission spectra (for example, Law 11), both (for example, Law 9), or neutron-induced photon production (for example, Law 2).

Law 1 (ENDF law 1): Equiprobable energy bins.
Exiting energies are sampled from a table of equiprobable energy bins in a manner similar to the sampling of scattering cosines (described in the "Sampling of Angular Distributions" section). See also Laws 21 and 22.

Law 2 Discrete photon energy.
The value provided in the library is EG. The secondary photon energy, E'_γ, is either
E'_γ=EG for non-primary photons
or E'_γ=EG + [A/(A+1)] E_n for primary photons,
where A is the atomic weight to neutron weight ratio of the target, and E_n is the incident neutron energy.

Law 3 (ENDF law 3): Inelastic scattering (n,n′) from nuclear levels.
Letting primes indicate the center-of-mass system and unprimes indicate the target-at-rest system, then:

$$E'_{out} = \left(\frac{A}{A+1}\right)^2 \left[E_{in} + \frac{Q\,(A+1)}{A}\right]$$

$$E_{out} = E'_{out} + \left[\frac{E_{in} + 2\mu'(A+1)(E_{in}E'_{out})^{1/2}}{(A+1)^2}\right]$$

$$\mu = \sqrt{E'_{out}/E_{out}} \cdot \mu' + \left[\sqrt{E_{in}/E_{out}}\ /(A+1)\right]$$

$\mu = \cos\theta$ is the cosine of the laboratory scattering angle, θ.

Law 4 Tabular distribution.
For each incident neutron energy, the table consists of triplets of: secondary energies, E(I), probability density functions, PDF(I), and cumulative density functions, CDF(I). Interpolation between adjacent points in the outgoing energy grid may be either linear-linear or by histogram. In either case the distribution is sampled exactly. The index K of the secondary energy interval, is sampled by $CDF(K) < \xi \leq CDF(K+1)$ where ξ is a random number on the interval [0,1).
For histogram interpolation, the secondary energy E′ is $E' = E(K) + (\xi - CDF(K))/PDF(K)$. For linear-linear interpolation, the secondary energy is:

$$E' = E(K) + \left\{ \frac{\left[PDF(K)^2 + 2 \cdot \left(\frac{PDF(K+1) - PDF(K)}{E(K+1) - E(K)}\right) \cdot (\xi - CDF(K))\right]^{1/2} - PDF(K)}{\left(\frac{PDF(K+1) - PDF(K)}{E(K+1) - E(K)}\right)} \right\}$$

In either of the last two equations, the random number ξ is the same as that generated initially for the first equation. The histogram equation is used in the linear-linear case when $PDF(K+1) = PDF(K)$.

Law 5 (ENDF law 5): General evaporation spectrum.
The function g(x) is tabulated versus χ and the energy is tabulated versus incident energy E_{in}. The law is then

$$f(E_{in} \rightarrow E_{out}) = g\left(\frac{E_{out}}{T(E_{in})}\right) \quad .$$

This density function is sampled by

$E_{out} = \chi(\xi)\ T(E_{in})$,
where $T(E_{in})$ is a tabulated function of the incident energy
$\chi(\xi)$ is a tabulated function of the random number ξ.

Law 7 (ENDF law 7): Simple Maxwell Fission Spectrum.

$$f(E_{in} \to E_{out}) = C_0 * \sqrt{E_{out}}\ e^{-E_{out}/T(E_{in})} \quad .$$

The nuclear temperature $T(E_{in})$ is a tabulated function of the incident energy. The normalization constant C_0 is given by

$$C_0^{-1} = T^{3/2}\left[(\sqrt{\pi}/2)\ \mathrm{erf}\left(\sqrt{(E_{in}-U)/T}\right) - \sqrt{(E_{in}-U)/T}\ e^{-(E_{in}-U)/T}\right] \quad .$$

U is tabulated (usually by an evaluator) and limits E_{out} to $0 \le E_{out} \le E_{in}-U$. In MCNP this density function is sampled by the rejection scheme

$$E_{out} = -T(E_{in})\left[\frac{\xi_1^2\ \ln \xi_3}{\xi_1^2 + \xi_2^2} + \ln \xi_4\right] \quad .$$

where ξ_1, ξ_2, ξ_3, and ξ_4 are random numbers on the unit interval. ξ_1 and ξ_2 are rejected if $\xi_1^2 + \xi_2^2 > 1$.

Law 9 (ENDF law 9): Evaporation spectrum.

$$f(E_{in} \to E_{out}) = C{*}E_{out}e^{-E_{out}/T(E_{in})},$$

where the nuclear temperature $T(E_{in})$ is a tabulated function of the incident energy. The energy U is assigned so that E_{out} is limited by $0 \le E_{out} \le E_{in}-U$. The normalization constant C is given by

$$C^{-1} = T^2\ [1-e^{-(E_{in}-U)/T}(1+(E_{in}-U/T)].$$

In MCNP this density function is sampled by

$$E_{out} = -T(E_{in})\ \ln(\xi_1\xi_2).$$

where ξ_1 and ξ_2 are random numbers on the unit interval.

Law 11 (ENDF law 11): Energy Dependent Watt Spectrum.

$$f(E_{in} \rightarrow E_{out}) = C_0 \, e^{-E_{out}/a(E_{in})} \sinh \sqrt{b(E_{in})\, E_{out}} \quad .$$

This law is sampled in the same way as Law 10 except that a, b, L, and M now depend upon the incident energy and are looked up in a table. (Law 11 applies to ENDF/B-V.) The normalization constant C_0 is given by

$$C_0^{-1} = \frac{1}{2} \sqrt{\pi a^3 b/4}\; \exp(ab/4) \left[\mathrm{erf}\left(\sqrt{(E_{in}-U)/a} - \sqrt{ab/4}\right) + \mathrm{erf}\left(\sqrt{(E_{in}-U)/a} + \sqrt{ab/4}\right)\right]$$

$$- \, a \exp\left[- \left((E_{in}-U)/a\right) \sinh \sqrt{b(E_{in}-U)}\right] \quad .$$

where the constant U limits the range of outgoing energy so that $0 \le E_{out} \le E_{in}-U$.

Law 22 (UK law 2): Tabular linear functions of incident energy out. Tables of P_{ij}, C_{ij}, and T_{ij} are given at a number of incident energies E_i. If $E_i \le E_{in} < E_{i+1}$ then the i^{th} P_{ij}, C_{ij} and T_{ij} tables are used.

$$E_{out} = C_{ik} \, (E_{in} - T_{ik}).$$

where k is chosen according to

$$\sum_{j=1}^{k} P_{ij} < \xi \le \sum_{j=1}^{k+1} P_{ij} \quad .$$

where ξ is a random number on the unit interval [0,1). Laws 21 and 22 are really just different formats of Law 1.

Law 24 (UK law 6): A table of equiprobable boundaries of parameters T_i. T is sampled in a manner similar to E_{out} of Law 1. Final energy is then given by $E'=T*E_{in}$ where E_{in}

is the incoming neutron energy.

e. Fission Inelastic Scattering: For any fission reaction a number of neutrons, N_p, are emitted according to the value of $\bar{\nu}(E_{in})$. The average number of neutrons per fission, $\bar{\nu}(E_{in})$, is either a tabulated function of energy or a polynomial function of energy. If I is the largest integer less than $\nu(E_{in})$, then

$N_p = I+1$ if $\xi \leq \bar{\nu}(E_{in}) - I$

$N_p = I$ if $\xi > \bar{\nu}(E_{in}) - I$, where ξ is a random number.

The direction of each emitted neutron is sampled independently from the appropriate angular distribution table by the procedure For fissions that use the FISP subroutine,

The energy of each fission neutron is determined from the appropriate (that is, as specified in the evaluation) emission law. The three laws used for fission neutron spectra are 7,9, and 11. These laws are discussed in the preceding section, section d.

Once the energy and direction of the emerging neutrons in the target-at-rest system are determined, the velocity of the target is added to get the velocity in the laboratory system. MCNP then models the transport of the first neutron out after storing all other neutrons in the bank.

6. The $S(\alpha,\beta)$ treatment

The thermal scattering model referred to in this manual by the name $S(\alpha,\beta)$ is, in reality, a complete representation of thermal neutron scattering by molecules and crystalline solids. Two processes are allowed: (1) inelastic scattering with cross section σ_{in} and a coupled energy-angle representation derived from an ENDF/B $S(\alpha,\beta)$ scattering law, and (2) elastic scattering with no change in the outgoing neutron energy for solids with cross section σ_{el} and an angular treatment derived from lattice parameters. The elastic scattering treatment is chosen with probability $\sigma_{el}/(\sigma_{el}+\sigma_{in})$. This thermal scattering treatment also allows the representation of scattering by multiatomic molecules (for example, BeO).

For the inelastic treatment, the distribution of secondary energies is represented by a set of equally probable final energies (typically 16 or 32) for each member of a grid of initial energies from an upper limit of typically 4 eV down to 10^{-5} eV along with a set of angular data for each initial and final energy. The selection of a final energy E′ given an initial energy E may be characterized by sampling from the distribution

$$p(E'|E_i < E < E_{i+1}) = \frac{1}{N} \sum_{j=1}^{N} \delta\left[E'-\rho E_{i,j}-(1-\rho)E_{i+1,j}\right],$$

where E_i and E_{i+1} are adjacent elements on the initial energy grid,

$$\rho = \frac{E_{i+1} - E}{E_{i+1} - E_i} \quad ,$$

N is the number of equally probable final energies, and E_{ij} is the j^{th} discrete final energy for incident energy E_i.

There are two allowed schemes for the selection of a scattering cosine following selection of a final energy and final energy index j. In each case, the $(i,j)^{th}$ set of angular data is associated with the energy transition $E = E_i \rightarrow E' = E_{i,j}$.

(1) The data consist of sets of equally probable discrete cosines $\mu_{i,j,k}$ for $k=1,\ldots,\nu$ with ν typically 4 or 8. An index k is selected with probability $1/\nu$, and μ is obtained by the relation

$$\mu = \rho\mu_{i,j,k} + (1-\rho)\mu_{i+1,j,k} \, .$$

(2) The data consist of bin boundaries of equally probable cosine bins. In this case, random linear interpolation is used to select one set or the other, with ρ being the probability of selecting the set corresponding to incident energy E_i. The subsequent procedure consists of sampling for one of the equally probable bins and then choosing μ uniformly in the bin.

For elastic scattering, the above two angular representations are allowed for data derived by an incoherent approximation. In this case, one set of angular data appears for each incident energy and is used with the interpolation procedures on incident energy described above.

For elastic scattering, when the data have been derived in the coherent approximation, a completely different representation occurs. In this case, the data actually stored are the set of parameters D_k, where

$$\sigma_{el} = D_k/E \qquad \text{for } E_{Bk} \leq E < E_{Bk+1}$$
$$\sigma_{el} = 0 \qquad \text{for } E < E_{B1}$$

and E_{Bk} are Bragg energies derived from the lattice parameters. For incident energy E such that $E_{Bk} \leq E \leq E_{Bk+1}$,

$$P_i = D_i/D_k \text{ for } i = 1,\ldots,k$$

represents a discrete cumulative probability distribution that is sampled to obtain index i, representing scattering from the i^{th} Bragg edge. The scattering cosine is then obtained from the relationship

$$\mu = 1 - 2E_{Bi}/E \, .$$

D. Photon Interactions

The photon portion of MCNP comprises two main treatments: (1) the simple physics treatment that assumes photoelectric absorption is a terminal event but models it by implicit capture, and (2) the detailed physics treatment that accounts for fluorescent photons after photoelectric absorption. The detailed physics treatment is the default, but the user can make a choice on the PHYS=P card. The following discussion is based in part on Ref. 18.

1. Simple Physics Treatment

This treatment is inadequate for high Z nuclides or deep penetration problems. The physical processes treated are photoelectric effect, pair production, and Compton scattering on free electrons. Because this treatment is intended primarily for higher energy photons, the photoelectric effect is regarded as an absorption (without fluorescence), scattering (Compton) is regarded to be on free electrons (without use of form factors), and the highly forward coherent Thomson scattering is ignored. Thus the total cross section σ_t is regarded as the sum of three components:

$$\sigma_t = \sigma_{pe} + \sigma_{pp} + \sigma_s.$$

a. Photoelectric effect: This is treated as a pure absorption by implicit capture with a corresponding reduction in the photon weight WGT, and hence does not result in the loss of a particle history except for Russian roulette played on the weight cutoff. At every collision, the weight $WGT \cdot \sigma_{pe}/\sigma_t$ and energy $E \cdot WGT \cdot \sigma_{pe}/\sigma_t$ are tallied in the appropriate capture bins. The noncaptured weight $WGT(1 - \sigma_{pe}/\sigma_t)$ is then forced to undergo either pair production or Compton scattering.

b. Pair production: In a collision resulting in pair production [probability $\sigma_{pp}/(\sigma_t - \sigma_{pe})$], it is assumed that the kinetic energy WGT(E − 1.022) MeV of the electron-positron pair produced is deposited as thermal energy at the time and point of collision, with isotropic production of one photon of energy 0.511 MeV with weight 2·WGT. This 0.511 MeV photon is then followed to termination. The rare single 1.022-MeV annihilation photon is ignored.

c. Compton scattering: The alternative to pair production is Compton scattering on a free electron, with probability $\sigma_s/(\sigma_t - \sigma_{pe})$. In the event of such a collision, the objective is to determine the energy E′ of the scattered photon, and $\mu = \cos\theta$ for the angle θ of deflection from the

line of flight. This yields at once the energy WGT(E − E′) deposited at the point of collision and the new direction of the scattered photon.

The differential cross section for the process is given by the Klein–Nishina formula

$$K(\alpha,\mu)d\mu = \pi r_o^2\left(\frac{\alpha'}{\alpha}\right)^2\left[\frac{\alpha'}{\alpha} + \frac{\alpha}{\alpha'} + \mu^2 - 1\right]d\mu \quad , \qquad (2.2)$$

where r_o is the classical electron radius 2.817938 x 10^{-13} cm, α and α' are the incident and final photon energies in units of 0.511 MeV [$\alpha = E/(mc^2)$, where m is the mass of the electron and c is the speed of light], and $\alpha' = \alpha/[1 + \alpha(1 - \mu)]$.

The Compton scattering process is sampled exactly by Kahn's method[28] below 1.5 MeV and by Koblinger's method[29] above 1.5 MeV as analyzed and recommended by Blomquist and Gelbard.[30]

2. Detailed Physics Treatment

The detailed physics treatment for photons of energies from 1 keV to 100 MeV provides for fluorescent emission and the modification of Thomson and Klein–Nishina differential cross sections by appropriate form factors that take electron binding effects into account. A photon cross–section library has been prepared, incorporating all constants required by this treatment, for elements Z = 1, ..., 94, in a form designed to expedite computation. The total photon cross section is given by the sum of the photoelectric, pair production, and coherent and incoherent scattering components.

The detailed physics treatment for point detectors is inadvisable, as explained in section g on page 79.

a. Mean free path: MCPLIB has, for each nuclide, a table of the logarithms $L_i(Z) = \ln [E_i(Z)]$ of suitable energies, including the photoelectric edges. It also has a matrix $L_{j,i}(Z) = \ln \sigma_{j,i}(Z)$, containing for j = 1, 2, 3, and 4 the logarithms of corresponding cross sections (when the latter are nonzero) for incoherent scattering, coherent scattering, photoelectric effect, and pair production, respectively. The data in MCPLIB are based mostly on the ENDF compilation. For scattering (j = 1,2), the cited cross sections were obtained by numerical integration, based on the same form factors used in the Monte Carlo treatment of such collisions and referred to below.

In the collision routine, a photon of energy E, starting from a point in a particular medium, has a mean free path

$$\lambda = \frac{1}{\sum_{Z} N(Z) \sum_{j=1}^{4} \sigma_j(Z)} \quad ,$$

where Z runs over all elements present in the medium, N(Z) is the corresponding atomic density, and $\sigma_j(Z)$ is the cross section for process j, each log-log interpolated to energy E. A random number ξ on [0,1) then determines the (infinite medium) distance to collision, $d = -\lambda \ln \xi$.

If collision occurs, two random numbers ξ_1 and ξ_2 serve to sample the element Z hit, and the process j responsible. The former results from a comparison of ξ_1/λ with the partial Z-sums obtained above and present in the memory. The latter process j is determined by a similar comparison of

$$\xi_2 \sum_{j=1}^{4} \sigma_j(Z)$$

with the partial sums involved, the individual $\sigma_j(Z)$ being also retained from the λ computation.

<u>b. Incoherent (Compton) scattering</u>: The objective, if such a process occurs (j = 1), is to determine the angle θ of scattering from the incident line of flight (and thus the new direction), the new energy E′ of the photon, and the local energy deposition E − E′ (the recoil kinetic energy of the electron).

Incoherent scattering is assumed to have the differential cross section $\sigma_1(Z,\alpha,\mu)\, d\mu = I(Z,v)\, K(\alpha,\mu)\, d\mu$, where I(Z,v) is an appropriate scattering factor, modifying the Klein-Nishina cross section in Eq. (2.2).

Qualitatively, the effect of I(Z,v)/Z is to decrease the Klein-Nishina cross section (per electron) more extremely in the forward direction, for low E and for high Z independently. For any Z, I(Z,v) increases from $I(Z,0) = 0$ to $I(Z,\infty) = Z$. The parameter $v = v(\alpha,\mu)$ is a given function of α and μ which, for a particular incident energy α, increases from $v(\alpha,1) = 0$ at $\mu = 1$ to a maximum value $\bar{v} = v(\alpha,-1)$ at $\mu = -1$. The essential features of I(Z,v) are indicated in Fig. 2.4.

The Klein-Nishina formula is sampled exactly as in the simple physics case and the outgoing energy E′ and angle μ are rejected according to the form factors. The energy E−E′ is deposited locally.

For the point detector routine of the general code, one requires, for a given μ (determined by the detector position), the probability of (incoherent) scattering to the angular range $(\mu,\mu + d\mu)$, $p(\mu)d\mu = I(Z,v)K(\alpha,\mu)d\mu/\sigma_1(Z,\alpha)$, where $\sigma_1(Z,\alpha)$ is the integrated

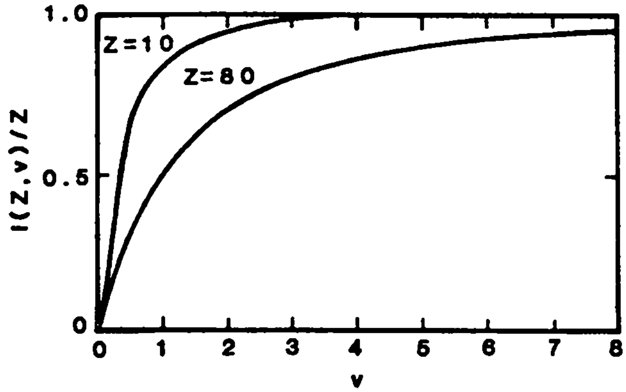


Figure 2.4

incoherent cross section. The values of πr_0^2 and of $\alpha'/\alpha = 1/[1 + \alpha(1 - \mu)]$ are needed in $K(\alpha,\mu)$; $I(Z,v)$ is obtained by linear interpolation at the computed value of $v = v(\alpha,\mu)$; and $\sigma_1\ (Z,\alpha) = \sigma_1(Z)$, at the incident energy E, is recoverable from the free path routine.

c. Coherent (Thomson) scattering: This process ($j = 2$) involves no energy loss, only the scattering angle θ being required.

The differential cross section is now $\sigma_2(Z,\alpha,\mu)d\mu = C^2(Z,v)T(\mu)d\mu$, where $C(Z,v)$ is a form factor modifying the (energy-independent) Thomson cross section $T(\mu) = \pi r_0^2(1 + \mu^2)d\mu$.

The general effect of $C^2(Z,v)/Z^2$ is to decrease the Thomson cross section more extremely for backward scattering, high E, and low Z, being opposite in these respects to the effect of $I(Z,v)/Z$ on $K(\alpha,\mu)$ in Section b above. For a given Z, $C(Z,v)$ decreases from $C(Z,0) = Z$ to $C(Z,\infty) = 0$. The parameter is here the $v = K\alpha\sqrt{1-\mu}$ of that section, with maximum $\bar{v} = \sqrt{2}\,K\alpha$ for given α. The qualitative features of $C(Z,v)$ are shown in Fig. 2.5.

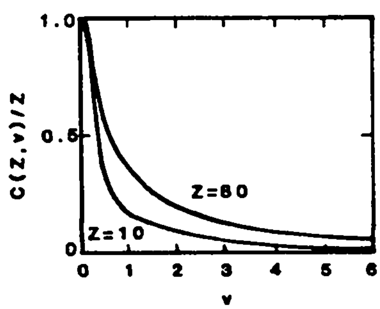


Figure 2.5

For the point detector routine, one must evaluate the density $p(\mu) = \pi r_0^2(1 + \mu^2)\ C^2(Z,v)/\sigma_2\ (Z,\alpha)$ for given μ. Here $\sigma_2\ (Z,\alpha)$ is the integrated coherent cross section. Although $\sigma_2\ (Z,\alpha) = \sigma_2(Z)$ is recoverable from the λ routine, the value of $C^2(Z,v)$ at $v = K\alpha\cdot\sqrt{1 - \mu}$ must be interpolated in the original $C^2(Zv_i)$ tables separately stored on the cross-section library for this purpose.

Note that at high energies, coherent scattering is virtually straight ahead with no energy loss; thus, it appears from a transport viewpoint that no scattering took place. For a point detector to sample this scattering, the point must lie on the original track ($\mu \simeq 1$), which is seldom the case. Thus, photon point detector variances will generally be much greater with detailed photon physics than with simple physics .

<u>d. Photoelectric effect</u>: A collision of this type (j=3) consists of the absorption of the incident photon of energy E with the consequent ejection (or excitation) of an orbital electron of binding energy $e < E$, giving the electron a kinetic energy of E−e. There are a number of physical possibilities, three of which will be considered; zero, one, or two fluorescent photons are emitted.

(1) No photons greater than 1 keV are emitted. In this event the cascade of electrons that fills up the orbital vacancy left by the photoelectric ejection produces electrons and low-energy photons (Auger effect), which deposit energy locally or nearly so. Consequently, the entire incident energy E is tallied as energy deposition and the collision is terminal.

(2) One fluorescent photon of energy greater than 1 keV is emitted. The photon energy E′ is the difference in incident photon energy E, less the ejected electron kinetic energy E−e, less a residual excitation energy e′, which is ultimately dissipated by further Auger processes. This dissipation leads to additional electrons or photons of still lower energy. In general, $E'=E-(E-e)-e'=e-e'$. These primary transactions are taken to have the full fluorescent yield from all possible upper levels e′, but are apportioned among the x-ray lines $K\alpha_1$, ($L_3 \rightarrow K$); $K\alpha_2$,($L_2 \rightarrow K$); $K\beta_1'$, (mean $M \rightarrow K$); and $k\beta_2'$, (mean $N \rightarrow K$).

(3) If the residual excitation e′ of process (2) exceeds 1 keV, a second fluorescence of interest may occur. An electron of binding energy e″ may fill the orbit of binding energy e′, emitting a second fluorescent photon of energy $E''=e'-e''$. As before, the residual excitation e″ is dissipated by further Auger events, which we take to be negligible and do not follow. The energy e″ is thus deposited locally. These secondary transitions come from all upper shells and go to L shells. Thus the primary transitions must be $K\alpha_1$ or $K\alpha_2$ to leave an L shell vacancy.

Each fluorescent photon born as discussed above is taken to be emitted isotropically and is transported, provided that $E',E'' > 1$ keV. The binding energies e, e′, and e″ are very nearly the x-ray absorption edges, because the x-ray absorption cross section takes an abrupt jump as it becomes energetically possible to eject (or excite) the electron of energy first $E \cong e''$, then e′, then e, etc. The jump can be as much as a factor of 20 (for example, K-carbon).

A photoelectric event is regarded as terminal for elements $Z < 12$, since the possible fluorescence energy is below 1 keV. The event can only be a single fluorescence of energy above 1 keV for $31 > Z \geq 12$, but double fluorescence (each above 1 keV) is possible for $Z \geq 31$. For $Z \geq 31$, primary lines $K\alpha_1$, $K\alpha_2$, and $K\beta_1'$ are possible and, in addition, for $Z \geq 37$, the $K\beta_2'$ line is possible.

e. Pair Production: This process ($j = 4$) is considered only in the field of a nucleus. The threshold is $2mc^2[1 + (m/M)] \cong 1.022$ MeV, M being the nuclear mass and m the mass of the electron.

If such a collision occurs, the incident photon -- of energy E -- vanishes; the kinetic energy of the created positron/electron pair, assumed to be $E - 2mc^2$, is deposited locally at the collision point; the positron is considered to be annihilated with an electron at the point of collision, resulting in a single photon of weight twice that of the incoming photon, and an energy of mc^2=0.511 MeV is emitted isotropically. The very rare single-annihilation photon of 1.022 MeV is omitted.

f. Energy Range: If all other effects (bremsstrahlung, etc.) are ignored, nothing prevents using the code to the upper limit of 100 MeV. Below 0.001 MeV, photons are allowed to have only analog capture. They are quickly eliminated because of the very large photoelectric capture cross section.

g. Caution: The use of the detailed photon physics treatment is not recommended for photon point detector calculations. Improved figures of merit (FOMs) for photon point detector calculations can be obtained by using the simple photon physics treatment instead of the detailed physics treatment. The improvement in the FOM for photon lines can be more than a factor of 10 for tallies with small relative errors because the total photon cross section for the detailed physics treatment includes coherent scattering that is highly peaked in the forward direction. For point detectors, this event is difficult to sample because the photon must be traveling directly at the detector point and undergo a coherent scattering event. The simple physics treatment is, therefore, always recommended with point detector calculations of direct photon line tallies.

For spectra calculations, however, detailed physics is recommended when fluorescent radiation is important (for example, photon energy above 0.12 MeV and high-Z materials) or for a problem that is many mean free paths thick for photon energies below 1 MeV. Note that as of now, no bremsstrahlung photon sources from electron slowing are included in MCNP. These photons can dominate the physics for thick, high-Z materials and high-incident photon energies.

C. *Summary Comparison of Simple vs Detailed Photon Physics*

1. Simple Photon Physics Treatment

a. Photoelectric effect treated as pure absorption by implicit capture; no fluorescence.

b. Pair production gives one 0.511008 MeV photon at WGT=2*WGT; isotropic emission.

c. Compton scattering (incoherent) is pure Klein-Nishina; assumes free electrons and no form factors.

d. Thomson scattering (coherent) is ignored because the photon is taken to be scattered nearly straight ahead; thus no form factors or free electrons.

e. Appropriate for: higher energies; point detectors; when fluorescence not important (that is, lower Z).

2. Detailed Photon Physics Treatment (default)

a. Photoelectric ejection of orbital electron results in 0, 1 or 2 fluorescent photons emitted. Track terminates for: no photon emitted; $E_{out} < 1$ keV; $Z < 12$ (when E_{out} would be < 1 keV.)

b. Pair production – same as simple physics treatment.

c. Compton scattering (incoherent) is Klein-Nishina modified for atomic binding of electrons which reduce the cross section for: forward angles; low photon energy; high Z.

d. Thomson scattering (coherent) has no energy loss. Form factors opposite to Compton, namely cross section reduced for: backward angles; high photon energy; low Z.

e. Appropriate for: high Z; deep penetration; when detailed low energy (fluorescence) spectra important. Inappropriate for point detectors.

Bremsstrahlung photon sources from the slowing down of electrons are not included in either treatment. All photon cross sections are assumed to be purely absorbing for energies below 0.001 MeV.

V. *TALLIES*

MCNP provides six standard neutron tallies and five standard photon tallies, but these basic tallies can be easily modified by the user in almost any way desired. All tallies are normalized to be per starting particle.

Tally Mnemonic	Description
F1:N or F1:P	Surface current
F2:N or F2:P	Surface flux
F4:N or F4:P	Track length estimate of cell flux
F5a:N or F5a:P	Flux at a point (point detector)
F6:N or F6:N,P or F6:P	Track length estimate of energy deposition
F7	Track length estimate of fission energy deposition

The above six tally categories represent the basic MCNP tally types. To have many tallies of a given type, add multiples of 10 to the tally number. For example, F1, F11, F21...F981, F991 are all type F1 tallies. Particle type is specified by appending a colon and the particle type. For example, F11:N and F96:N are neutron tallies and F2:P and F25:P are photon tallies. F6 tallies may be for both neutrons and photons -- F16:N,P.

Thought should be given to applying these tallies and to comparing one tally with another. For example, if the flux is varying as $1/R^2$ in a cell, an average flux in the cell determined by the F4 tally will be higher than the flux at a point in the center of the cell determined by a detector. This same consideration applies to the average flux provided by DXTRAN spheres (see page 128).

Standard summary information to give the user a better insight into the physics of his problem and the adequacy of the Monte Carlo simulation includes a complete accounting of the creation and loss of all tracks and their energy; the number of tracks entering and reentering a cell plus the track population in the cell; the number of collisions in a cell; the average weight, mean free path, and energy of tracks in a cell; the activity of each nuclide in and a cell; and a complete weight balance for each cell.

The quantities actually scored in MCNP before the final normalization per starting particle are presented in Table 2.1. Note that adding an asterisk (*Fn) changes the units and multiplies the tally as indicated in the last column of Table 2.1. Table 2.1 also defines much of the notation used in the remainder of this section.

Table 2.1
Tally Quantities Scored

Tally	Fn Quantity	Fn Units	*Fn Multiplier	*Fn Units
F1	W		E	MeV
F2	$W/(\lvert\mu\rvert * A)$	$1/cm^2$	E	MeV/cm^2
F4	$W * T_l / V$	$1/cm^2$	E	MeV/cm^2
F5	$W * p(\mu) * \exp(-\lambda)/(2\pi R^2)$	$1/cm^2$	E	MeV/cm^2
F6	$W * T_l * \sigma_T(E) * H(E) * \rho_a / m$	MeV/gm	1.60219E-22	jerks/gm
F7	$W * T_l * \sigma_f(E) * Q * \rho_a / m$	MeV/gm	1.60219E-22	jerks/gm

W = particle weight
E = particle energy (MeV)
$\lvert\mu\rvert$ = absolute value of cosine of angle between surface normal and particle trajectory. If $\lvert\mu\rvert < .1$, set $\lvert\mu\rvert = .05$.
A = surface area (cm^2)
T_l = track length (cm) = transit time * velocity
V = volume (cm^3)
$p(\mu)$ = probability density function: μ = cosine of angle between particle trajectory and detector
λ = total mean free path to detector
R = distance to detector (cm)
$\sigma_T(E)$ = microscopic total cross section (barns)
$H(E)$ = heating number (MeV/collision)
ρ_a = atom density (atoms/barn-cm)
m = cell mass (gm)
$\sigma_f(E)$ = microscopic fission cross section (barns)
Q = fission heating Q-value (MeV)

The following MCNP definitions of current and flux come from reactor theory but are related to similar quantities in radiative transfer theory. The MCNP particle angular flux multiplied by the particle energy is the same as the intensity in radiative transfer theory. The MCNP particle total flux at energy E multiplied by the particle energy equals the integrated energy density times the speed of light in radiative transfer theory, and the MCNP particle current multiplied by the particle energy is analogous to the radiative flux crossing an area in radiative transfer theory. The MCNP particle current uses $|\mu|$ in the definition, whereas the radiative transfer flux uses μ in its definition. MCNP current is neither net nor positive nor negative current; it is the number of particles crossing a surface in a particular direction. The MCNP particle fluence multiplied by the particle energy is the same as the fluence in radiative transfer theory.

A. Surface Current Tally

The F1 surface current tally estimates the following quantity:

$$F1 = \int_A \int_\mu \int_t \int_E J(r,E,t,\mu)dE\ dt\ d\mu\ dA$$

$$*F1 = \int_A \int_\mu \int_t \int_E E*J(r,E,t,\mu)dE\ dt\ d\mu\ dA \quad .$$

This tally is the number of particles (quantity of energy for *F1) crossing a surface. The scalar current is related to the flux as $J(r,E,t,\mu) = |\mu|\Phi(r,E,t)A$. The range of integration over area, energy, time, and angle (A,E,t,μ) may be controlled by FS, E, T, and C cards, respectively.

B. Flux Tallies

The F2, F4 and F5 flux tallies are estimates of

$$F_{2,4,5} = \int_t \int_E \Phi(r,E,t)dE\ dt$$

$$^{*}F_{2,4,5} = \int_t \int_E E^{*}\Phi(r,E,t)dE\ dt \quad .$$

The range of integration over energy and time may be controlled by E and T cards. The F2 surface flux and F4 cell flux tallies are discussed below in more detail. The F5 detector flux tally, a major topic, is discussed later on page 87.

The units of the flux tally are the units of the source. If the source has units of particles per unit time, the tally is also particles per unit time. When the source has units of particles, this tally represents a fluence tally. A steady-state flux solution can be obtained by having a source with units of particles per unit time and integrating over all time (that is, omitting the Tn card). The flux can be obtained from the fluence tally for a time-dependent source by dividing the tally by the time bin width. These tallies can all be made per unit energy by dividing by the energy bin width.

1. Track Length Estimate of Cell Flux (F4)

The definition of particle flux is $\Phi(r,E,t) = vN(r,E,t)$, where v = particle velocity and N = particle density = particle weight/cell volume. Thus the time integrated flux is

$$\int \Phi(r,E,t)dt = Wv\ dt/V = WT_\ell\ /V.$$

Because of the track length term T_ℓ in the numerator, this tally is known as a track length estimate of the flux. It is generally quite reliable because there are frequently many tracks in a cell (compared to the number of collisions), leading to many contributions to this tally.

2. Surface Flux (F2)

The surface flux may be thought of as the limiting case of the cell flux when the cell becomes infinitely thin as illustrated in Fig. 2.6.

$$F2 = \lim_{\delta \to \infty} WT_\ell/V$$
$$= (W\ \delta/|\cos\ \theta|)\ /\ (A\delta) \ = \ W/(A|\mu|).$$

As the cell thickness δ approaches zero, the volume approaches $A\delta$ and the track length approaches $\delta/|\mu|$, where μ=cos θ, the angle between the surface normal and the particle trajectory. This definition of flux also follows

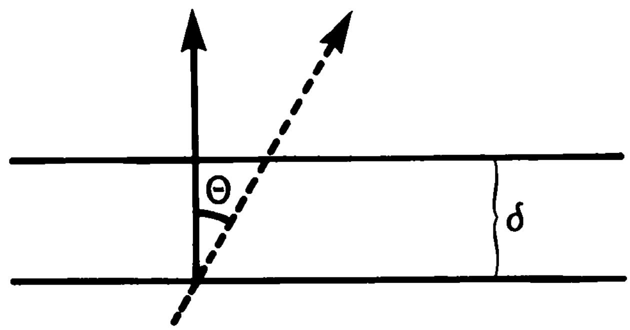


Figure 2.6

directly from the relation between flux and current, $J(r,E,t,\mu) = |\mu|\Phi(r,E,t)A$. MCNP sets $|\mu| = .05$ when $|\mu| < .1$. This construct is essential for stochastic calculation of surface areas when the normal analytic procedure fails.

C. Cell Energy Deposition Tallies

The cell heating and energy deposition tallies are the following track length estimates

$$F_{6,7} = \rho_a/\rho_g \int_t \int_E H(E)\Phi(r,E,t)dE\ dt \ ,$$

where ρ_a = atom density (atoms/barn−cm)
ρ_g = gram density (grams/cm^3)
H(E) = heating response

The units of the heating tally are MeV/gm unless an asterisk is used ($*F_{6,7}$) to change units to jerks/gm (1 jerk = 1.60219E−22 MeV). Note that the asterisk causes the $F_{6,7}$ tally to be modified by a constant rather than by energy as in other tallies. Also note that the heating tallies are merely flux tallies (F4) multiplied by an energy-dependent multiplier (FM card).

Note that the F7 tally includes the gamma heating, whereas the F6 tally does not. Thus for fissionable materials, the F7 result will often be greater than the F6 result even though F7 includes only fission and F6 includes all reactions. The F6 tally does not include gammas because they are assumed to be deposited elsewhere; the F7 tally does include gammas because they are assumed to be deposited locally. The true heating is found by summing the neutron and photon F6 tallies in a coupled neutron/photon calculation. In a neutron-only problem, F6 will give the right heating of light materials only if <u>all</u> photons escape the geometry. F7 will give about the right heating of fissionable materials only if no

photons come from elsewhere, all fission photons are immediately captured, and nonfission reactions can be ignored. The F7 tally cannot be used for photons. Examples of combining the neutron and photon F6 tallies are F6:N,P and F516:P,N

The heating response H(E) has different meanings, depending upon context.

1. F6 Neutrons

$$H(E) = \sigma_T(E)\ H_{avg}(E),$$

where the heating number is

$$H_{avg}(E) = E - \sum_i p_i(E) \left[\bar{E}_{out_i}(E) + Q_i - \bar{E}_{\gamma_i}(E) \right],$$

where σ_T = total neutron cross section
E = incident neutron energy
$p_i(E)$ = probability of reaction i
$\bar{E}_{out_i}$ = average exiting neutron energy for reaction i
Q_i = Q-value of reaction i
$\bar{E}_{\gamma_i}$ = average energy of exiting gammas for reaction i

2. F6 Photons

$$H(E) = \sigma_T(E)H_{avg}(E)$$

$$H_{avg}(E) = \sum_{i=1}^{3} p_i(E)\cdot(E - \bar{E}_{out}).$$

i=1 incoherent (Compton) scattering with form factors
i=2 pair production ($\bar{E}_{out}$=1.022016=$2m_0c^2$)
i=3 photoelectric.

All energy transferred to electrons is assumed to be deposited locally.

3. F7 Neutrons

$$H(E) = \sigma_f(E)Q,$$

where $\sigma_f(E)$ = total fission cross section
Q = fission Q-value (MeV).

The Q-values as tabulated represent the total prompt energy release per

fission and can be found in subroutine TALLY in two data statements for 23 isotopes.

4. F7 Photons

H(E) is undefined because photofission is not included in MCNP.

D. Flux at a Detector

Flux may be estimated at a point in MCNP by one of three methods: point detector, ring detector, or once-more-collided flux estimator. All of these schemes may yield anomalous statistics and must be used with caution. All may take advantage of the MCNP DD card Russian roulette. All require a user-supplied source probability density function whenever a user-supplied source is specified. None works with the $S(\alpha,\beta)$ thermal treatment.

1. Point Detector

A point detector is a deterministic estimate of the flux at a point in space and is made at source and collision events throughout the random walk. Suppose $p(\theta,\varphi)d\Omega$ is the probability of the particle's scattering or being born into the solid angle $d\Omega$ about the direction (θ,φ). If R is the distance to the detector from the collision or source point, then

$$p(\theta,\varphi)d\Omega \cdot e^{-\int_0^R \Sigma_t(s)ds}$$

yields the probability of scattering into $d\Omega$ about (θ,φ) and arriving at the detector point with no further collisions. The attenuation of a beam of monoenergetic particles passing through a material medium is given by $\exp[-\int_0^R \Sigma_t(s)ds]$ where s is measured along the direction from the collision or source point to the detector and $\Sigma_t(s)$ is the macroscopic total cross section at s. If dA is an element of area normal to the scattered line of flight to the detector, $d\Omega = dA/R^2$ and therefore

$$p(\theta,\varphi)\ \frac{dA}{R^2}\ e^{-\int_0^R \Sigma_t(s)ds}$$

is the expression giving the probability of scattering toward the detector and passing through the element of area dA normal to the line of flight to the detector. Because the flux is by definition the number of particles

passing through a unit area normal to the scattered direction, the general expression for the contribution to the flux is given by

$$\frac{p(\theta,\varphi)}{R^2}\, e^{-\int_0^R \Sigma_t(s)ds}.$$

In all the MCNP scattering distributions and in <u>most</u> of the standard sources, the density function $p(\theta,\varphi)$ is uniform in θ and azimuthally symmetric in φ. Another way of saying this is that the angles θ and φ are independent variables so that $p(\theta,\varphi)$ can be written as the product of a probability density function in θ times a probability density function in φ. Illustrating this,

$$\int_\Omega p(\theta,\varphi)d\Omega = \int_0^{2\pi}\int_0^{\pi} p(\theta,\varphi)\ \sin\theta\ d\theta\ d\varphi$$

$$= \int_0^{2\pi}\int_{-1}^{1} p(\mu,\varphi)d\mu\ d\varphi.$$

Because μ and φ are independent, $p(\mu,\varphi) = p(\mu)/2\pi$. Here $p(\mu) = \int_0^{2\pi} p(\mu,\varphi)d\varphi$, the marginal density function in μ, and therefore $\int_{-1}^{1} p(\mu)d\mu = 1$. Clearly $d\varphi/2\pi$ is the uniform density of φ [which is the marginal density of φ, given by $\int_{-1}^{1} p(\mu,\varphi)d\mu = \int_{-1}^{1} p(\mu)\ \Phi(\varphi)\ d\mu = \Phi(\varphi) =$ constant. But $\int_0^{2\pi}\phi(\varphi)d\varphi = \phi(\varphi)\cdot 2\pi = 1$ so $\Phi(\varphi) = 1/2\pi$].

If $p(\theta,\varphi) = p(\mu,\varphi) = p(\mu)/2\pi$ is substituted in the expression for the flux, the expression used in MCNP is arrived at:

$$\Phi(r,E,t,\mu) = Wp(\mu)e^{-\lambda}/(2\pi R^2),$$

where W = particle weight;
$\lambda = \int_0^R \Sigma_t(s)ds$ = total number of mean free paths integrated over the trajectory from the source or collision point to the detector;
R = distance from source or collision event to detector; and
$p(\mu)$ = value of probability density function at μ, the cosine of the angle between the particle trajectory and the direction to the detector. This quantity is discussed later in subsection 4, page 95.

A point detector is also known as a "next event estimator" because it is a tally of the flux at a point if the next event is a trajectory without

further collision directly to the point detector. A contribution to the point detector is made at every source or collision event. The $e^{-\lambda}$ term accounts for attenuation between the present event and the detector point. The $1/2\pi R^2$ term accounts for the solid angle effect. The $p(\mu)$ term accounts for the probability of scattering toward the detector instead of the direction selected in the random walk. For an isotropic source or scatter, $p(\mu) = 0.5$ and the solid angle terms reduce to the expected $1/4\pi R^2$. (Note that $p(\mu)$ may be larger than unity, because it is the value of a density function and not a probability.) Each contribution to the detector may be thought of as the transport of a pseudoparticle to the detector.

The R^2 term in the denominator of the point detector causes a singularity that makes the theoretical variance of this estimator infinite. That is, if a collision event occurs near the detector point, R approaches zero and the flux approaches infinity. The technique is still valid and unbiased, but convergence is slower and often impractical. If the detector is not in a scattering medium, a collision close to the detector is impossible. For problems where there are many scattering events near the detector, a cell or surface estimator should be used instead of a point detector tally. If there are so few scattering events near the detector that cell and surface tallies are impossible, a point detector may still be used with a specified average flux region close to the detector. This region is specified by a radius R_o of a fictitious sphere surrounding the point detector. R_o may be specified either in centimeters or in mean free paths. If R_o is specified in centimeters and if $R < R_o$, the point detector estimation inside R_o is assumed to be the average flux uniformly distributed in volume.

$$\Phi(R < R_o) = \int \Phi dV \Big/ \int dV$$

$$= Wp(\mu) \frac{\int_0^{R_0} e^{(-\Sigma_t r)} 4\pi r^2 dr}{\frac{4}{3}\pi R_o^3}$$

$$= \frac{Wp(\mu)(1 - e^{-\Sigma_t R_o})}{\frac{2}{3}\pi R_0^3 \Sigma_t} \quad .$$

If $\Sigma_t=0$, the detector is not in a scattering medium, so no collision can occur.

$$\Phi(R < R_o, \Sigma_t=0) = \frac{Wp(\mu)R_o}{\frac{2}{3}\pi R_0^3} \quad .$$

If R_o is specified in mean free paths, then $\lambda_0=R_o=\Sigma_t R$ and

$$\Phi(\lambda < \lambda_0) = \frac{Wp(\mu)(1 - e^{-\lambda_0})\Sigma_t^2}{\frac{2}{3}\pi\lambda^3} \quad .$$

The choice of R_o may require some experimentation. For a detector in a void region or a region with very few collisions (such as air), R_o may be set to zero. For a typical problem, setting R_o to a mean free path or some fraction thereof is usually adequate. If R_o is in centimeters, it should correspond to the mean free path for some average energy in the sphere. Be certain when defining R_o that the sphere it defines does not encompass more than one material unless you understand the consequences. This is especially true when using R_o in terms of mean free path because R_o becomes a function of energy and can vary widely. In particular, if R_o is defined in terms of mean free paths and if a detector is on a surface that bounds a void on one side and a material on the other, the contribution to the detector from the direction of the void will be zero even though the importance of the void is nonzero. The reason is simply that the volume of the artificial sphere is infinite in a void. Contributions to the detector from the other direction (that is, across the material) will be accounted for.

Whereas detectors differing only in R_o are coincident detectors (see page 99), there is little cost incurred by experimenting with several detectors that differ only by R_o in a single problem.

2. Ring Detector[31]

A ring detector tally is a point detector tally in which the point detector location is not fixed but rather sampled from some location on a ring. Most of the previous section on point detectors applies to ring detectors as well. In MCNP three ring detector tallies FX, FY, and FZ correspond to rings located rotationally symmetric about the x, y, and z coordinate axes. A ring detector usually enhances the efficiency of point detectors for problems that are rotationally symmetric about a coordinate axis. Ring detectors may also be used for problems where the user is interested in the average flux at a point on a ring about a coordinate axis.

Although the ring detector is based on the point detector, which has a $1/R^2$ singularity and an unbounded variance, the ring detector has a finite variance and only a $1/R_{min}$ singularity, where R_{min} is the minimum distance between the contributing point and the detector ring.[32]

In a cylindrically symmetric system, the flux is constant on a ring about the axis of symmetry. Hence, one may sample uniformly for positions on the ring to determine the flux at any point on the ring. The efficiency of the ring detector is enhanced in MCNP by biasing the selection of point detector locations to favor those near the contributing collision or source point. This results in the same total number of detector contributions, but the large contributions are sampled more frequently, resulting in a reduced relative error.

The theoretically "perfect" biasing function to select a detector position on a ring is the range of detector contributions from a single collision. For isotropic scattering in the lab system, this is proportional to $e^{-P}R^{-2}$, where P is the number of mean free paths and R is the distance from the collision point to the detector point. For most practical applications, using a biasing function involving P presents prohibitive computational complexity except for homogeneous medium problems. For air transport problems, a biasing function resembling e^{-P} has been used with good results. A biasing function was desired that would be applicable to problems involving dissimilar scattering media and would be effective in reducing variance. The function R^{-2} meets these requirements.

In Fig. 2.7, consider a collision point, (x_o,y_o,z_o) at a distance R from a point detector location (x,y,z). The point (x,y,z) is to be selected from points on a ring of radius r. The ring is symmetric about the y-axis in the discussion.

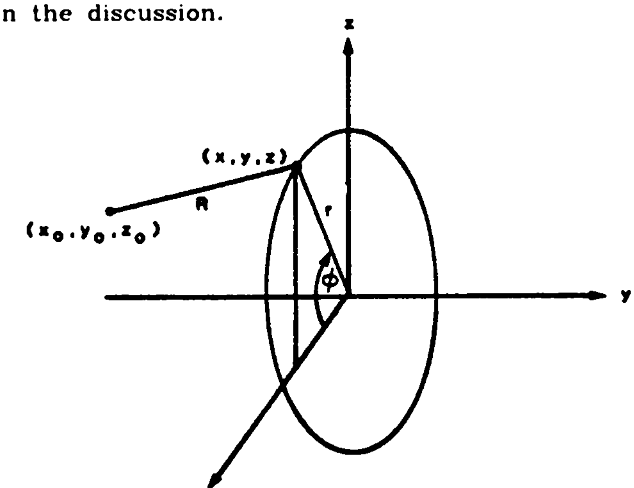


Figure 2.7

To sample a position (x,y,z) on the ring with a $1/R^2$ bias, we pick φ from the density function $p(\varphi)=C/(2\pi R^2)$, where C is a normalization constant. To pick φ from $p(\varphi)$, let ξ be a random number on the unit interval. Then

$$\xi = \frac{C}{2\pi}\int_{-\pi}^{\varphi} \frac{d\varphi}{R^2}$$

$$= \frac{C}{2\pi}\int_{-\pi}^{\varphi} \frac{d\varphi}{(x_o-r\cos\varphi)^2 + (y_o-y)^2 + (z_o-r\sin\varphi)^2}$$

$$= \frac{C}{2\pi}\int_{-\pi}^{\varphi} \frac{d\varphi}{a + b\cos\varphi + c\sin\varphi}$$

$$= \frac{1}{\pi}\tan^{-1}\left\{ \frac{1}{C}\left[(a-b)\tan\frac{\varphi}{2} + c\right] + \frac{\pi}{2}\right\} .$$

where $a = r^2 + x_o^2 + (y-y_o)^2 + z_o^2$
$b = -2rx_o$
$c = -2rz_o$
$C = (a^2 + b^2 + c^2)^{1/2}$.
Solving for $\tan\frac{\varphi}{2}$,

$$\tan\frac{\varphi}{2} = \frac{C}{a-b}\tan\left[\pi\left(\xi - \frac{1}{2}\right)\right] - c .$$

Letting $t = \tan\varphi/2$,
then $x = r\cos\varphi = (1-t^2)/(1+t^2)$
$y = y$ (fixed)
$z = r\sin\varphi = 2t/(1+t^2)$.

For ring detectors, the $1/R^2$ biasing has been supplemented when it is weak to include a biasing based on angle to select the point on the ring. This angle is in the plane of the ring and is relative to the shortest line from the collision point to the detector ring. The angle that would probably be selected would pick the same point on the ring as a straight

line through the axis of the problem, the collision point, and the ring. The angle least likely to be picked would choose the point on the opposite side of the ring. This approach will thus make scores with smaller attenuations more often. This supplemental biasing is achieved by requiring that $a \geqslant 3/2\ (b^2+c^2)^{1/2}$ in the above equation.

If the radius of the ring is very large compared to the dimensions of the scattering media (such that the detector sees essentially a point source in a vacuum), the ring detector is still more efficient than a point detector. The reason for this unexpected behavior is that the individual scores to the ring detector for a specific history have a mean closer to the true mean than to the regular point detector contributions. That is, the point detector contributions from one history will tend to cluster about the wrong mean because the history will not have collisions uniformly in volume throughout the problem, whereas the ring detector will sample many paths through the problem geometry to get to different points on the ring.

3. Once-More-Collided Flux Estimator (OMCFE)[33]

To use the once-more-collided flux estimator in Version 3a, you must call X-6 in Los Alamos to learn a special procedure. We plan to remove the OMCFE from MCNP in Version 3b and will use these calls to find out how much the OMCFE is needed.

The OMCFE is a point detector designed to work well in a scattering medium. Most of the discussion about point detectors applies equally to OMCFE detectors. The procedure involves the sampling of an imaginary intermediate collision point from a real collision. By choosing the probability density function of this intermediate point in a suitable way, an estimator with only a $1/R$ singularity can be derived, compared to the analog flux estimator that has a $1/R^2$ singularity, where R is the distance from collision point to detector. The variance is proportional to

$$\int R^2 dR \left(\frac{1}{4\pi R}\right)^2$$

and is therefore finite.

To speed up the calculation, the sampling of an intermediate collision point is not done at every real collision. An imaginary sphere of specified radius R_o is drawn around the detector. The once-more-collided game is played if the real collision occurs within this sphere. It is also played if the collision is outside the sphere but the direction after collision is inside the cone defined by the collision point and the sphere. Otherwise, an analog contribution is made from the subsequent collisions.

This procedure may be thought of as four separate cases, illustrated in Figs. 2.8 a, b, c and d.

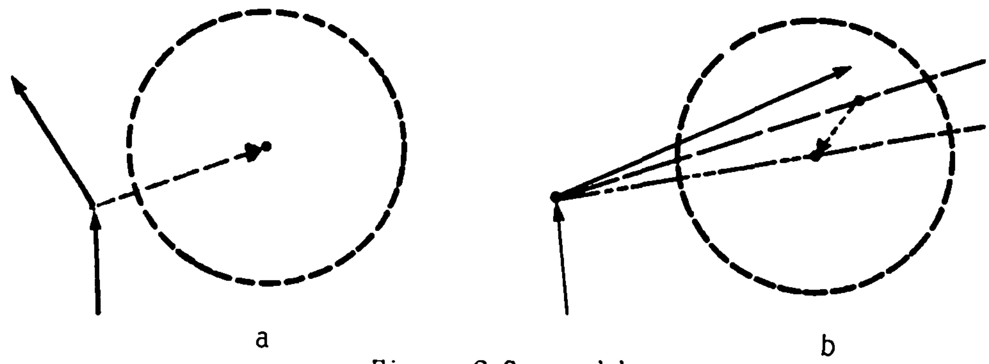


Figure 2.8 a and b

Case 1 (Figure 2.8 a). For all source events and at all collisions not preceded by cases 2, 3, or 4, an analog point detector tally is made. That is, an analog tally is made if no OMCFE tally was made at the previous collision.

Case 2 (Figure 2.8 b). If the track after a collision intersects the sphere, reselect the direction and sample an intermediate collision point. The reselected direction is biased to be more toward the center of the sphere although occasionally it will be sampled farther away than the actual trajectory. The intermediate collision point is also biased to be near the sphere center, but it may occasionally be far from the center, even outside the sphere. An analog point detector tally is then made from this intermediate collision point.

Case 3 (Figure 2.8 c). If the collision is inside the sphere and the angle between the exiting trajectory and a line drawn to the sphere center is greater than 90°, sample the intermediate collision point on the exiting trajectory preferentially close to the collision point. An analog point detector tally is then made from this intermediate collision point.

Case 4 (Figure 2.8 d). If the collision is inside the sphere and the angle between the exiting trajectory and a line drawn to the sphere center is less than 90°, reselect the direction and sample an intermediate collision point. This is the same as case 2 except that the collision is inside the sphere.

The effectiveness of the OMCFE is very sensitive to the radius R_0 of the sphere. If R_0 is chosen too small, this in effect reduces the

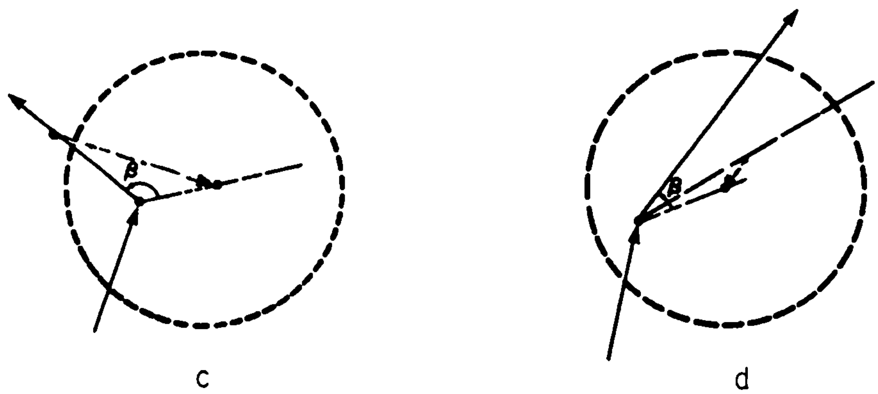


Figure 2.8 c and d

technique to the analog case. If R_0 is too large, the running time increases considerably. The scheme is also relatively ineffective if the collision point is a long way from the detector. A sphere radius of about 1.5 to 3 mean free paths at the source energy seems most effective.

The OMCFE is interesting because of its finite variance; however, it is of little practical use. In problems where there are so many collisions near the detector point that the analog point detector has difficulty, the number of particle tracks is sufficient to make use of cell or surface tallies.

4. General Considerations of Point Detector Estimators

a. Pseudoparticles and detector reliability: Point, ring and OMCFE detectors are all hybrid Monte Carlo methods that employ a deterministic shortcut. The process of simulating particle transport from one place to another is short-circuited. Transport from the source or collision point to the detector is replaced by a deterministic estimate of the potential contribution to the detector. This transport between the source or collision point and the detector may be thought of as being via "pseudoparticles." Pseudoparticles undergo no further collisions. (The OMCFE case may be thought of as two successive pseudoparticles.) These particles do not reduce the weight or otherwise affect the random walk of the particles that produced them. They are merely estimates of a potential contribution. The only resemblance to Monte Carlo particles is that the quantity they estimate requires an attenuation term that must be summed over the trajectory from the source or collision to the detector. Thus most of the machinery for transporting particles can also be used for the pseudoparticles. No records (for example, tracks entering) are kept about pseudoparticle passage.

Because detectors rely on pseudoparticles rather than particle simulation by random walk, they should be considered only as a very useful last resort. Detectors are unbiased estimators, but their use can be tricky, misleading, and occasionally unreliable. Consider the problem illustrated in Fig. 2.9.

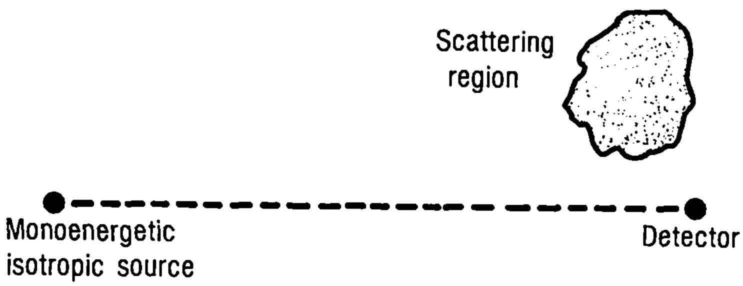


Figure 2.9

The monoenergetic isotropic point source will always make the same contribution to the detector, so the variance of that contribution will be zero. If no particles have yet collided in the scattering region, the detector tally will be converged to the source contribution, an event that is wrong and misleading. But as soon as a particle collides in the scattering region, the detector tally and its variance will jump. Then the detector tally and variance will steadily decrease until the next particle collides in the scattering region, at which time there will be another jump. These jumps in the detector score and variance are characteristic of the detector method. They can be eliminated only by better sampling of the scattering region that caused them. If the user had biased Monte Carlo particles toward the tally region instead of using pseudoparticles, surely the scattering region would have been sampled better, thus eliminating the jump problem. Consequently, it is recommended that detectors be used as a last resort. When they are used, the tally fluctuation charts printed at the end of every MCNP run should be examined closely to see the degree of the fluctuations. Also the detector diagnostic print tables should be examined to see if any one pseudoparticle trajectory made an unusually large contribution to the tally. Detector results should be viewed suspiciously if the relative error is greater than 5%.

b. Detectors and reflecting surfaces: Detectors used with reflecting surfaces usually give wrong answers because pseudoparticles travel only in straight lines. Consider Fig. 2.10, with a point detector and eight source cells. The imaginary cells and point detector are also shown on the other side of the mirror. The solid line shows the source contribution from the indicated cell. MCNP does not allow for the

dashed-line contribution on the other side of the reflecting surface. The result is that contributions to the detector will always be from the solid path instead of from a mixture of solid and dashed contributions. This same effect will also occur at every collision. Therefore, the detector tally will be lower (with the same starting weight) than the correct answer and thus should not be used. This effect is even worse for problems with more than one reflecting plane.

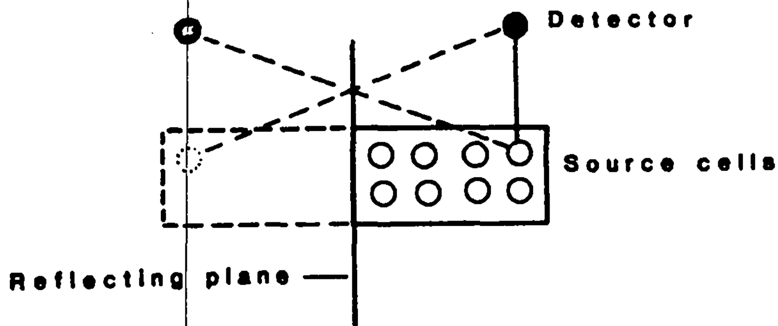


Figure 2.10

c. Variance reduction schemes for detectors: Pseudoparticles of point detectors are not subject to the variance reduction schemes of analog particles of the random walk. They do not split according to importances, weight windows, etc., although they are terminated by entering zero importance cells. However, two Russian roulette games are available specifically for detector pseudoparticles.

The MCNP PD card may be used to specify the pseudoparticle generation probability for each cell. The entry for each cell i is p_i where $0 \leq p_i \leq 1$. Pseudoparticles are created with probability p_i and weight $1/p_i$. If $p_i=1$; which is the default; then every source or collision event produces a pseudoparticle. If $p_i=0$, no pseudoparticle will be produced. Setting $p_i=0$ in a cell that can actually contribute to a detector erroneously biases the detector tally by eliminating such contributions. Thus $p_i=0$ should be used only if the true probability of scoring is zero or if the score from cell i is unwanted for some legitimate reason such as problem diagnostics. Fractional entries of p_i are also discouraged because the PD card applies equally to all pseudoparticles. Usually it is better to use the DD card to Russian roulette just the unimportant pseudoparticles.

The MCNP DD card controls both the detector diagnostic printing and a Russian roulette game played on pseudoparticles in transit to detectors. The Russian roulette game is governed by the input parameter k, which controls a comparison weight w_c internal to MCNP:

$w_c = -k$ if $k < 0$

$w_c = 0$ if $k=0$

$w_c = 0$ if $k > 0$ and $N \leq 200$

$w_c = (k/N) \sum_i^I \varphi_i$ if $k > 0$ and $N > 200$,

where N = number of histories run so far

I = number of pseudoparticles started so far

$\varphi_i = Wp(\mu)e^{-\lambda}/(2\pi R^2)$

= contribution of the ith pseudoparticle to the detector tally.

When each pseudoparticle is generated, W, $p(\mu)$, and R are already known before the expensive tracking process is undertaken to determine λ. If $Wp(\mu)/(2\pi R^2) < w_c$, clearly the pseudoparticle contribution to the detector φ_i will be less than the comparison weight. Playing Russian roulette on all pseudoparticles with $\varphi_i < w_c$ avoids the expensive tracking of unimportant pseudoparticles. Most are never started. Some are started but are rouletted as soon as λ has increased to the point where $Wp(\mu)e^{-\lambda}/(2\pi R^2) < w_c$. Rouletting pseudoparticles whose expected detector contribution is small also has the added benefit that those pseudoparticles surviving Russian roulette now have larger weights, so the disparity in particle weights reaching the detector is reduced. Typically, using the DD card will enhance the efficiency of detector problems by a factor of ten. This Russian roulette is so powerful that it is one of two MCNP variance reduction options (the other being implicit capture) that is turned on by default. The default value of k is 0.1.

The DD card Russian roulette game is almost foolproof. Performance is relatively insensitive to the input value of k. For most applications the default value of k=0.1 is adequate. Usually, k should be chosen so that there are 1–5 transmissions (pseudoparticle contributions) per source history. If k is too large, too few pseudoparticles are sampled; thus $k \geq 1$ is a fatal error. Because a random number is used for the Russian roulette game invoked by $k > 0$, the addition of a detector tally can affect the random walk tracking processes. Detectors are the only tallies that can affect results (not answers, since everything will eventually converge to the same solution.) Therefore, it is recommended that $k < 0$ be used once a good guess at w_c can be made. This is especially important if a problem is to be debugged by starting at some history past the first one. Also, $k < 0$ makes the first 200 histories run faster.

There are two cases when it is beneficial to turn off the DD card Russian roulette game by setting k=0. First, when looking at the tail of a spectrum or some other low probability event, the DD card roulette game will preferentially eliminate small scores and thus wipe out the very phenomenon of interest. For example, if energy bias is used to preferentially produce high energy particles, these biased particles will have a lower weight and thus preferentially will be rouletted by the DD card game. Second, in very deep penetration problems, pseudoparticles will sometimes go a long way before being rouletted. In this rare case it is

wasteful to roulette a pseudoparticle after a great deal of time has been spent following it and perhaps a fractional PD card should be used or, better yet, a cell or surface tally.

d. Coincident detectors: Because tracking pseudoparticles is very expensive, MCNP will use a single pseudoparticle for more than one detector if possible. Such detectors are known as coincident detectors because they must be identical in the following respects:

same geometric location
same particle type (that is, neutron or photon)
same DD card Russian Roulette control parameter, k
same PD card entries, if any, and
same fictitious sphere radius R_0 for OMCFE.

Note that energy bins, time bins, tally multipliers, response functions, fictitious sphere radii, user-supplied modifications (TALLYX), etc., may all be different.

Coincident detectors are "free" in that most detector cost is spent in tracking a pseudoparticle. Subsequent detectors using the same pseudoparticle require only inconsequential additional computational effort.

e. Direct vs total contribution: Unless specifically turned off by the user, MCNP automatically prints out both the total and direct detector contribution. As mentioned earlier, pseudoparticles are generated at source and collision events. The direct contribution is that portion of the tally from pseudoparticles born at source events. The total contribution is the total tally from both source and collision events. For Mode N P problems with photon detectors, the direct contribution is from pseudophotons born in neutron collisions. The direct contributions for detailed photon physics will be smaller than the simple physics direct results because coherent scattering is included in the detailed physics total cross section and omitted in the simple physics treatment.

f. Angular distribution functions for point detectors: All detector estimates require knowledge of the $p(\mu)$ term. This is the value of the probability density function at an angle θ, where $\mu = \cos\theta$. This quantity is available to MCNP for the standard source and for all kinds of collisions except $S(\alpha,\beta)$ collisions. For user-supplied source subroutines, MCNP assumes an isotropic distribution

$$p(\mu)d\mu = \frac{d\Omega}{4\pi} = \int_0^{2\pi} \frac{d\mu d\varphi}{4\pi} = \frac{1}{2}\, d\mu \quad .$$

Therefore, $p(\mu)=1/2$ (=PSC in the MCNP code). If the source distribution is not isotropic in a user-supplied source subroutine, the user must also supply a subroutine SRCDX if there are any detectors or DXTRAN spheres in the problem. In subroutine SRCDX, the parameter PSC, which is the value of $p(\mu)$, must be set for each detector and DXTRAN sphere. An example of how this is done and also a description of several other source angular distribution functions is in Chapter 4.

g. Incompatability with $S(\alpha,\beta)$ thermal treatment: Because the angular distribution probability density function $p(\mu)$ is exceedingly difficult to compute in $S(\alpha,\beta)$ thermal collisions, detectors and DXTRAN are presently incompatible with the $S(\alpha,\beta)$ thermal treatment.

E. Additional Tally Features

The six standard MCNP tally types may be controlled, modified, and beautified by a number of additional tally features. These features are described in detail in Chapter 3; an overview is given here.

Bin limit control: The integration limits of the various tally types may be controlled by E, T, C, and FS cards. The E card establshes energy bin ranges; the T card establishes time bin ranges; the C card establishes cosine bin ranges; and the FS card segments the surface or cell of a tally into subsurface or subcell bins.

Flagging: Cell and surface flagging (CF and SF cards) may be used to determine which portions of a tally come from where. For example, if flux tally F4 in cell 1 is flagged by cells 2, 3, and 4 on the CF4 card, the tally for cell 1 will be output twice: first, the total tally, which is the total flux in cell 1; second, the flagged tally, which is that portion of the flux caused by particles having passed through cells 2, 3, or 4.

Multipliers and modification: MCNP tallies may be modified in many different ways. The EM, TM, and CM cards allow the quantities in each energy, time, or cosine bin to be multiplied by a different constant. This capability is useful for modeling response functions or changing units. For example, a surface current tally may have its units changed to per steradian by entering the inverse steradian bin sizes on the CM card.

The DE and DF cards allow modeling of an energy-dependent dose function that is a continuous function of energy from a table whose data points need not coincide with the tally energy bin structure (E card). An example of such a dose function is the flux-to-radiation dose conversion factor given in Appendix H.

Any tally except F6:N,P can be multiplied by any continuous-energy quantity known to MCNP and its data libraries by use of an FM card. For example, average heating numbers $H_{avg}(E)$ and total cross section $\sigma_T(E)$ are stored on the MCNP data libraries. Thus an F4 tally can be multiplied by $\sigma_T H_{avg}(E)\rho_a/\rho_g$ to convert it to an F6 tally, or an F5 detector tally can be multiplied by the same quantity to calculate heating at a point. The FM card can modify any flux or current tally of the form $\int\varphi(E)dE$ into $\int R(E)\varphi(E)dE$, where R(E) is any combination of sums and products of energy-dependent quantities known to MCNP.

The FM card may also be used to model attenuation. Here the tally is converted to $\int\varphi(E)e^{-\sigma_t(E)\rho_a x}dE$, where x is the thickness of the attenuator, ρ_a is its atom density, and Σ_t is its total cross section. Double parentheses allow the calculation of $\int\varphi(E)e^{-\sigma_t(E)\rho_a x}R(E)dE$. More complex expressions of $\Sigma_t(E)\rho_a x$ are allowed so that many attenuators may be stacked. This is useful for calculating attenuation in line-of-sight pipes and through thin foils and detector coatings, particularly when done in conjunction with point and ring detector tallies. Beware, however, that attenuation assumes that the attenuated portion of the tally is lost from the system by capture or escape and cannot be scattered back in.

User modification: If all the above capabilities still do not provide the user with exactly what is desired, tallies may be relatively easily modified by a user-supplied TALLYX subroutine (FU card). As with a user-supplied SOURCE subroutine, which lets the user provide his own specialized source, the TALLYX subroutine lets the user modify any tally in a straightforward manner with all the programming changes conveniently located in a single subroutine.

Tally output format: Not only can users change the contents of MCNP tallies, but the output format can be modified as well. Any desired descriptive comment may be added to the tally title by the tally comment (FC) card. The printing order may be changed (FQ card) so that instead of, for instance, getting the default output blocks in terms of time vs energy, they could be printed in blocks of segment vs cosine. The tally bin, which is monitored for the tally fluctuation chart printed at the problem end, may also be reselected (TF card). Detector tally diagnostic prints are controlled with the DD card. Finally, most diagnostic summary information and the detail with which the tallies themselves are described can be controlled with the PRINT card.

F. Time-Dependent Sources and Tallies

Because the geometry and material compositions are independent of time, the expected tally, $T(t,t+\tau)$, at time $t+\tau$ from a source particle

emitted at time t is identical to the expected tally, $T(0,\tau)$, from a source particle emitted at time 0. Thus, if a calculation is performed with all source particles started at t=0, one has an estimate of $T(0,\tau)$ and the tallies T_{Q_i} from a number of time-distributed sources $Q_i(t)$ can be calculated at time η as

$$T_{Q_i}(\eta) = \int Q_i(t)\ T(t,\eta)\ dt = \int Q_i(t)\ T(0,\eta-t)\ dt.$$

This last integral can be done either directly in a TALLYX subroutine by sampling t from $Q_i(t)$ and recording each particle's tally (shifted by t) or after the calculation by integrating $Q_i(t)$ multiplied by the histogram estimate of $T(0,\eta-t)$.

VI. ESTIMATION OF THE MONTE CARLO PRECISION

Monte Carlo results represent an average of the contributions from many histories sampled during the course of the problem. An important quantity equal in stature to the Monte Carlo answer (or tally) itself is the statistical error or uncertainty associated with results. The importance of this error and its behavior vs the number of histories cannot be overemphasized because the user not only gains insight into the quality of the result, but also can determine if a tally is well behaved. If a tally is not well behaved, the estimated error associated with the result generally will not reflect the true confidence interval of the result and, thus, these answers could be completely erroneous.

The purpose of this section is to educate MCNP users about the proper interpretation of the MCNP estimated means and relative errors and caution them to carefully check the results and the associated FOM tables in the tally fluctuation charts to ensure a well-behaved and meaningful tally.

A. Monte Carlo Means, Variances, and Standard Deviations

Monte Carlo results are obtained by sampling possible random walks and assigning a score x_i (for example, x_i=energy deposited by the i^{th} random walk) to each random walk. The scores assigned to various random walks will generally vary. Suppose p(x) is the probability density function for selecting a random walk that scores x to the tally being estimated. The true answer (or mean) is the expected value of x, E(x), where $E(x) = \int xp(x)dx$ = true mean. The quantity E(x) is seldom known because p(x) is not known directly but can be estimated by Monte Carlo through the random walk process as $\bar{x}$, which is given by

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{2.3}$$

where x_i is the value of x selected from p(x) for the i^{th} history and N is the number of histories calculated in the problem. The Monte Carlo mean $\bar{x}$ is the average value of the scores x_i for all the histories calculated in the problem. The relationship between E(x) and $\bar{x}$ is given by the Strong Law of Large Numbers, which states that if E(x) is finite, $\bar{x}$ tends to the limit E(x) as N approaches infinity.

The variance of the population of x values is a measure of the spread in these values and is given by

$$\sigma^2 = \int (x - E(x))^2\, p(x)dx = E(x^2) - (E(x))^2.$$

The square root of the variance is σ, which is called the standard deviation of the population of scores. As with E(x), σ is seldom known but can be estimated by Monte Carlo as S, given by (for large N)

$$S^2 = \frac{\sum_{i=1}^{N} (x_i - \bar{x})^2}{N-1} \approx \overline{x^2} - \bar{x}^2 \tag{2.4a}$$

and

$$\overline{x^2} = \frac{1}{N}\sum_{i=1}^{N} x_i^2. \tag{2.4b}$$

The quantity S is the estimated standard deviation of the population of x based on the values of x_i that were actually sampled.

The estimated variance of $\bar{x}$ is given by

$$S_{\bar{x}}^2 = \frac{S^2}{N}. \tag{2.5}$$

These formulae do not depend on any restriction on the distribution of x or $\bar{x}$ (such as normality) beyond requiring that E(x) and σ^2 exist and are finite. The estimated standard deviation of the mean $\bar{x}$ is given by $S_{\bar{x}}$.

It is important to note that $S_{\bar{x}}$ is proportional to $1/\sqrt{N}$, which is the inherent drawback to the Monte Carlo method. To halve $S_{\bar{x}}$, four times the

original number of histories must be calculated, a calculation that can be computationally expensive. The quantity $S_{\bar{x}}$ can also be reduced for a specified N by making S smaller, reducing the inherent spread of the tally results. This can be accomplished by using variance reduction techniques such as those discussed in section VII of this chapter.

B. Precision and Accuracy

There is an extremely important difference between precision and accuracy of a Monte Carlo calculation. As illustrated in Fig. 2.11, precision is the uncertainty in $\bar{x}$ caused by the statistical fluctuations of the x_i's for the portion of physical phase space sampled by the Monte Carlo process. Important portions of physical phase space might not be sampled because of problem cutoffs in time or energy, inappropriate use of variance reduction techniques, or an insufficient sampling of important low-probability events. Accuracy is a measure of how close the expected value of $\bar{x}$, E(x), is to the true physical quantity being estimated. The difference between this true value and E(x) is called the systematic error, which is seldom known. Error or uncertainty estimates for the results of Monte Carlo calculations refer only to the precision of the result and not to the accuracy. It is quite possible to calculate a highly precise result that is far from the physical truth because nature has not been modeled faithfully.

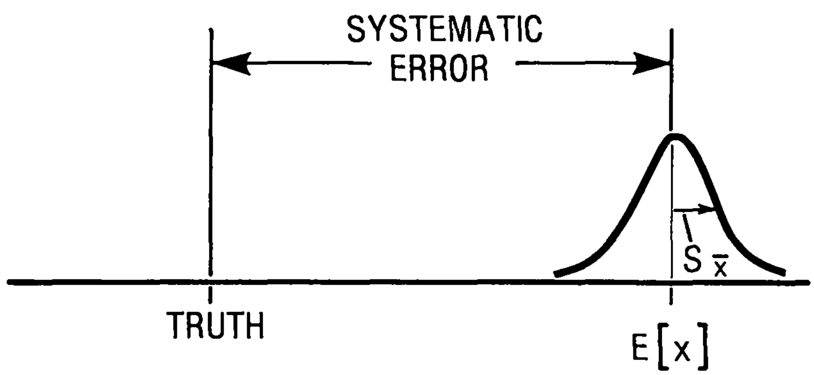


Figure 2.11

Three factors affect the accuracy of a Monte Carlo result: (1) the code, (2) problem modeling, and (3) the user. Code factors encompass: the physics features included in a calculation as well as the mathematical models used; uncertainties in the data, such as the transport and reaction cross sections, Avogadro's number, atomic weights, etc.; the quality of the representation of the differential cross sections in energy and angle; and

coding errors (bugs). All of the applicable physics must be included in a calculation to produce accurate results. Even though the evaluations are not perfect, more faithful representation of the evaluator's data should produce more accurate results. The descending order of preference for Monte Carlo data for calculations is continuous energy, thinned continuous energy, discrete reaction, and multigroup. Coding errors can always be a problem because no large code is bug-free. MCNP, however, is a very mature, heavily used production code. With steadily increasing use over the years, the likelihood of a serious coding error continues to diminish.

The second area, problem-modeling factors, can quite often contribute to a decrease in the accuracy of a calculation. Many calculations produce seemingly poor results because the model of the energy and angular distribution of the radiation source is not adequate. Two other problem-modeling factors affecting accuracy are the geometrical description and the physical characteristics of the materials in the problem.

The third general area affecting calculational accuracy involves user errors in the problem input or in user-supplied subroutines and patches to MCNP. The user can also abuse variance reduction techniques such that portions of the physical phase space are not allowed to contribute to the results. Checking the input and output carefully can help alleviate these difficulties. A last item that is often overlooked is a user's thorough understanding of any measured quantities being calculated. Factors such as detector efficiencies, data reduction and interpretation, etc., must be completely understood and included in the calculation, or the comparison is not meaningful.

The precision of a Monte Carlo result is affected by four user-controlled choices: (1) forward vs adjoint calculation, (2) tally type, (3) variance reduction techniques, and (4) number of histories run.

The choice of a forward vs adjoint calculation depends mostly on the relative sizes of the source and detector regions. Starting particles from a small region is easy to do, whereas transporting particles to a small region is generally hard to do. Because forward calculations transport particles from source to detector regions, forward calculations are preferable when the detector (or tally) region is large and the source region is small. Conversely, because adjoint calculations transport particles backward from the detector region to the source region, adjoint calculations are preferable when the source (or tally) region is large and the detector region is small. X-6 supports a multigroup adjoint version of MCNP called MCMG. A continuous-energy adjoint version of MCNP does not exist.

As alluded to above, the smaller the tally region, the harder it becomes to get good tally estimates. An efficient tally will average over as large a region of phase space as practical. In this connection, tally dimensionality is extremely important. A one-dimensional tally is

typically 10 to 100 times easier to estimate than a two-dimensional tally, which is 10 to 100 times easier than a three-dimensional tally. This fact is illustrated in Fig. 2.15 later in this section.

Variance reduction techniques can be used to improve the precision of a given tally by increasing the nonzero tallying efficiency and by decreasing the spread of the nonzero history scores. These two components are depicted in Fig. 2.12. A calculation will be more precise when the history-scoring efficiency is high and the variance of the nonzero scores is low. The user should strive for these conditions in every Monte Carlo calculation. Examples of these two components of precision are given on page 112.

As a <u>last</u> resort, more histories can be run to improve precision (see section C following). Since the precision is proportional to $1/\sqrt{N}$, running more particles is often costly in computer time and therefore is viewed as the method of last resort.

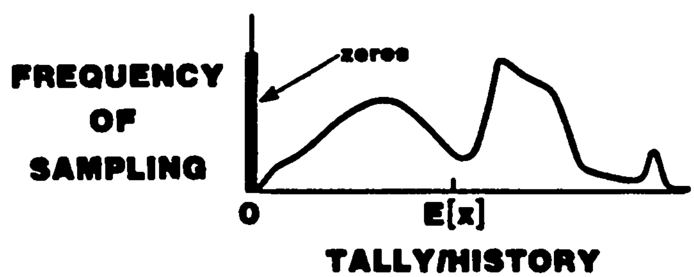


Figure 2.12

C. The Central Limit Theorem

To define confidence intervals for the precision of a Monte Carlo result, the Central Limit Theorem[10] of probability theory is used, stating that

$$\lim_{N\to\infty} \text{Prob}\left[E(x) + \alpha \frac{\sigma}{N^{1/2}} < \bar{x} < E(x) + \beta \frac{\sigma}{N^{1/2}}\right] = \frac{1}{\sqrt{2\pi}} \int_{\alpha}^{\beta} e^{-t^2/2}\, dt ,$$

where α and β can be any arbitrary values. In terms of the estimated standard deviation of $\bar{x}$, $S_{\bar{x}}$, this may be rewritten in the following approximation for large N:

$$\text{Prob}\left[\alpha S_{\bar{x}} < \bar{x} - E(x) < \beta S_{\bar{x}}\right] \sim \frac{1}{\sqrt{2\pi}} \int_{\alpha}^{\beta} e^{-t^2/2}\, dt\, .$$

This crucial theorem states that for large values of N (that is, as N tends to infinity) and identically distributed independent random variables x_i with finite means and variances, the distribution of the $\bar{x}$'s approaches a normal distribution. Therefore, for any distribution of tallies (an example is shown in Fig. 2.12), the distribution of resulting $\bar{x}$'s will be approximately normally distributed, as shown in Fig. 2.11, with a mean of E(x). If S is approximately equal to σ, which is valid for a statistically significant sampling of a tally, then

$$\bar{x} - S_{\bar{x}} < E(x) < \bar{x} + S_{\bar{x}}, \sim 68\% \text{ of the time} \qquad (2.6a)$$

and

$$\bar{x} - 2S_{\bar{x}} < E(x) < \bar{x} + 2S_{\bar{x}}, \sim 95\% \text{ of the time} \qquad (2.6b)$$

from standard tables for the normal distribution function.

The key point about the validity of these confidence intervals is that the physical phase space must be adequately sampled by the Monte Carlo process. If an important path in the geometry or a window in the cross sections, for example, has not been well sampled, both E(x) and $S_{\bar{x}}$ will be unknowingly incorrect and the results will be wrong, usually tending to be too small. The user must take great care to be certain that adequate sampling of the source, transport, and any tally response functions have indeed taken place.

D. Estimated Relative Errors in MCNP

All standard MCNP tallies are normalized to be per starting particle and are printed in the output with a second number, which is the estimated relative error defined as

$$R \equiv S_{\bar{x}}/\bar{x}. \qquad (2.7a)$$

The relative error is a convenient number because it represents statistical precision as a fractional result.

Combining Eqs. (2.3), (2.4), and (2.5), R can be written (for large N) as

$$R = \left[\frac{1}{N}\left(\frac{\overline{x^2}}{\bar{x}^2} - 1\right)\right]^{1/2} = \left[\frac{\sum_{i=1}^{N} x_i^2}{\left(\sum_{i=1}^{N} x_i\right)^2} - \frac{1}{N}\right]^{1/2} . \qquad (2.7b)$$

Several important observations about the relative error can be made from Eq. (2.7b). First, if all the x_i's are nonzero and equal, R is zero. Thus, low-variance solutions should strive to reduce the spread in the x_i's. If the x_i's are all zero, R is defined as zero. If only one nonzero score is made, R approaches unity as N becomes large. For nonnegative x_i's, it can be shown that by using the definition of R above, $S_{\bar{x}}$ can never be greater than $\bar{x}$. For positive and negative x_i's, this need not be true. The range of R values for nonnegative x_i's is therefore between zero and unity.

To determine what values of R lead to results that can be stated with confidence using Eqs. (2.6), consider Eq. (2.7b) for a difficult problem in which nonzero scores occur very infrequently. In this case,

$$\frac{1}{N} << \frac{\sum_{i=1}^{N} x_i^2}{\left(\sum_{i=1}^{N} x_i\right)^2}. \qquad (2.8a)$$

For clarity, assume that there are n out of N (n << N) nonzero scores that are identical and equal to x. With these two assumptions, R for "difficult problems" becomes

$$R_{D.P.} \sim \left[\frac{nx^2}{n^2x^2}\right]^{1/2} = \frac{1}{\sqrt{n}},\ n << N \quad . \qquad (2.8b)$$

This is expected because the limiting form of a binomial distribution with infrequent nonzero scores and large N is the Poisson distribution, which is the form used in detector "counting statistics."

Table 2.2
Estimated Relative Error R vs Number of Identical Tallies n for Large N

n	1	4	16	25	100	
R	1.0	0.5	0.25	0.20	0.10	0.05

Through use of Eqs. (2.8), a table of R values versus the number of tallies or "counts" can be generated as shown in Table 2.2. A relative error of 0.5 is the equivalent of four counts, which is hardly adequate for a statistically significant answer. Sixteen counts is an improvement, reducing R to 0.25, but still is not a large number of tallies. The same is true for n equals 25. When n is 100, R is 0.10, so the results should

be much improved. With 400 tallies, an R of 0.05 should be quite good indeed.

Based on this qualitative analysis, Table 2.3 presents the recommended interpretation of the 1σ confidence interval $\bar{x}(1 \pm R)$ for various values of R associated with an MCNP tally. These guidelines were determined empirically, based on years of experience using MCNP on a wide variety of problems.

Table 2.3
Guidelines for Interpreting the Relative Error R[a]

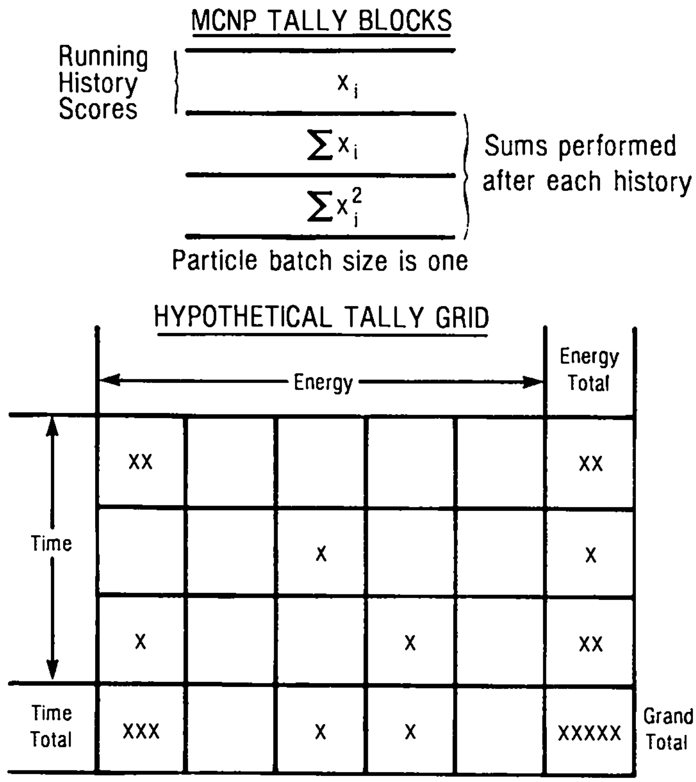
Range of R	Quality of the Tally
0.5 to 1	Garbage
0.2 to 0.5	Factor of a few
0.1 to 0.2	Questionable
< 0.10	Generally reliable except for point detector
< 0.05	Generally reliable for point detector

[a]$R = S_{\bar{x}}/\bar{x}$ and represents the estimated statistical relative error at the 1σ level. These interpretations of R assume that all portions of the problem phase space are being well sampled by the Monte Carlo process.

Point detector tallies generally require a smaller value of R for valid confidence interval statements because contributions near the detector point are usually extremely important and may be difficult to sample well. Experience has shown that for R less than 0.05, point detector results are generally reliable. For an R of 0.10, point detector tallies may only be known within a factor of a few.

MCNP calculates the relative error for each tally bin in the problem using Eq. (2.7b). Each x_i is defined as the total contribution from the i^{th} starting particle and all resulting progeny. This definition is important in many variance reduction methods, multiplying physical processes such as fission or (n,xn) neutron reactions that create additional neutrons, and coupled neutron/photon problems. The i^{th} source particle and its offspring may thus contribute many times to a tally and all of these contributions are correlated because they are from the same source particle.

Figure 2.13 represents the MCNP process of calculating the first and second moments of each tally bin and relevant totals using three tally



x = Score from the present history

Figure 2.13

storage blocks of equal length for each tally bin. The hypothetical grid of tally bins in the bottom half of Fig. 2.13 has 24 tally bins including the time and energy totals. During the course of the i^{th} history, sums are performed in the first MCNP tally storage block. Some of the tally bins receive no tallies and others receive one or more tallies. At the conclusion of the i^{th} history, the sums are added to the second MCNP tally storage block. The sums in the first MCNP tally storage block are squared and added to the third tally storage block. The first tally storage block is then filled with zeros and the i+1 history begins. After the last history N, the estimated tally means are computed using the second MCNP tally storage block and Eq. (2.3). The estimated relative errors are calculated using the second and third MCNP tally storage blocks and Eq. (2.7b). This method of estimating the error of the result produces the best estimate because the batch size is one, which minimizes the variance of the variance.[34]

Note that there is no guarantee that the estimated relative error will decrease inversely proportional to the $\sqrt{N}$ because of the statistical nature of the tallies. Early in the problem, R will generally have large statistical fluctuations. Later, infrequent large contributions may cause

fluctuations in $S_{\bar{x}}$ and to a lesser extent in $\bar{x}$ and therefore in R. MCNP calculates a FOM for one bin of each numbered tally to aid the user in determining the quality and efficiency of the tally. A variance error estimator has been used to examine in more detail the quality of various MCNP tallies,[35] but the work is considered too premature for inclusion into MCNP at this time.

E. MCNP FOM

The estimated relative error squared R^2 should be proportional to 1/N, as shown by Eq. (2.7a). The computer time T used in an MCNP problem should be directly proportional to N; therefore, R^2T should be approximately a constant. It is convenient to define a FOM of a tally to be

$$\mathrm{FOM} \equiv 1/R^2T \quad . \tag{2.9a}$$

The numerical value of the FOM, which should be approximately a constant, will increase as the problem becomes more efficient because R^2T will decrease. MCNP prints the FOM for one bin of each numbered tally as a function of N, where the unit of computer time T is minutes. The table is printed in particle increments of 1000 up to 20 000 histories. Between 20 000 and 40 000 histories, the increment is doubled to 2000. This trend continues, producing a table of up to 20 entries.

The FOM is a very important statistic about a tally bin and provides important information that should be studied by the user. It is a tally reliability indicator in the sense that if the tally is well behaved, the FOM should be approximately a constant with the possible exception of statistical fluctuations very early in the problem. An order-of-magnitude estimate of the expected fractional statistical fluctuations in the FOM is 2R. This result assumes that both the relative statistical uncertainty in the relative error is of the order of the relative error itself and the relative error is small compared to unity. The user should <u>always</u> examine the tally fluctuation charts at the end of the problem to check that the FOMs are approximately constant as a function of the number of histories for each tally.

The numerical value of the FOM can be better appreciated by considering the relation

$$R = 1/\sqrt{\mathrm{FOM} \bullet T} \tag{2.9b}$$

Table 2.4 shows the expected value of R that would be produced in a one-minute problem (T=1) as a function of the value of the FOM. It is clearly advantageous to have a large FOM for a problem because the computer time required to reach a desired level of precision is proportionally reduced. Examination of Eq. (2.9b) shows that doubling the FOM for a

problem will reduce the computer time required to achieve the same R by a factor of two.

Table 2.4
R Values as a Function of the FOM for T=1 Minute

FOM	1	10	100	1000	10000
R	1.0	0.32	0.10	0.032	0.010

In summary, the FOM has three possible uses. The most important use is as a tally reliability indicator. If the FOM is not approximately a constant (except for statistical fluctuations early in the problem), the confidence intervals may not overlap the precise answer the expected fraction of the time. A second use for the FOM is to optimize the efficiency of the Monte Carlo calculation by making several short test runs with different variance reduction parameters and then selecting the problem with the largest FOM. Remember that the statistical behavior of the FOM for a small number of histories may cloud the selection of techniques competing at the same level of efficiency. A third use for the FOM is to estimate the computer time required to reach a desired value of R by using $T \sim 1/R^2FOM$.

Note: Calculating the FOM requires that the computer time used is available to MCNP. Subroutine SECND may need to be modified at your installation to obtain this information.

F. Factors Affecting Problem Efficiency

Three factors that affect the efficiency of a Monte Carlo problem are (1) history-scoring efficiency, (2) dispersions in nonzero history scores, and (3) computer time per history. All three factors are included in the FOM. The first two factors control the value of R; the third is T.

The relative error can be separated into two components: the nonzero history-scoring efficiency component R^2_{eff} and the intrinsic spread of the nonzero x_i scores R^2_{int}. It therefore follows that if f is defined to be the fraction of histories producing nonzero x_i's,

$$R^2 = R^2_{eff} + R^2_{int} \; , \tag{2.10a}$$

where

$$R^2_{int} = \frac{\sum_{i=1}^{N} x_i^2}{\left(\sum_{i=1}^{N} x_i\right)^2} - \frac{1}{fN} \tag{2.10b}$$

and $$R^2_{eff} = (1-f)/(fN). \qquad (2.10c)$$

For identical nonzero x_i's, R^2_{int} is zero and for a 100% scoring efficiency, R^2_{eff} is zero. It is usually possible to increase f for most problems using one or more of the MCNP variance reduction techniques. These techniques alter the random walk sampling to favor those particles that produce a nonzero tally. The weights are then adjusted appropriately so that the expected tally is preserved. This topic is described in Sec. VII (Variance Reduction). The sum of the two terms of Eq. (2.10a) produces the same result as Eq. (2.7b).

These equations can be used to better understand the effects of scoring inefficiency; that is, those histories that do not contribute to a tally. Table 2.5 shows the expected values of R_{eff} as a function of f and the number of histories N. This table is appropriate for identical nonzero scores and represents the theoretical minimum relative error possible for a specified f and N. It is no surprise that small values of f require a compensatingly large number of particles to produce precise results.

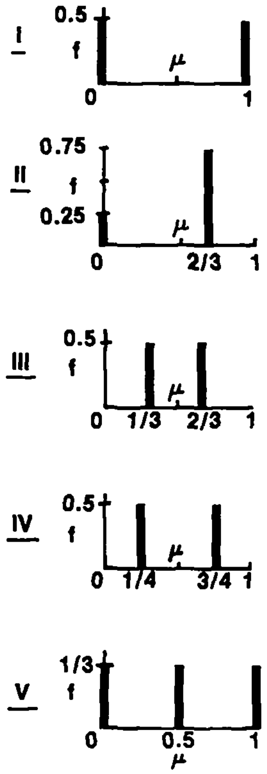
Table 2.5
Expected Values of R_{eff} as a Function of f and N

N \ f	0.001	0.01	0.1	0.5
10^3	0.999	0.315	0.095	0.032
10^4	0.316	0.099	0.030	0.010
10^5	0.100	0.031	0.009	0.003
10^6	0.032	0.010	0.003	0.001

A practical example of scoring inefficiency would be the case of infrequent high-energy particles in a down-scattering-only problem. If only a small fraction of all source particles have an energy in the highest energy tally bin, the dominant component of the relative error will probably be the scoring efficiency because only the high-energy source particles have a nonzero probability of contributing to the highest energy bin. For problems of this kind, it is often useful to run a separate problem starting only high-energy particles from the source and to raise the energy cutoff. The much-improved scoring efficiency will result in a much larger FOM for the high-energy tally bins.

To further illustrate the components of the relative error, consider the five examples of selected discrete probability density functions shown in Fig. 2.14. Cases I and II have no dispersion in the nonzero scores, cases III and IV have 100% scoring efficiency, and case V contains both elements contributing to R. The most efficient problem is case III. Note

FIVE CASES WITH A MEAN OF 0.5



I: $E[x] = 0.5(0+1) = 0.5$; $R = R_{eff} = 1/\sqrt{N}$; $R_{int} = 0$

II: $E[x] = 0 \times 1/4 + 2/3 \times 3/4 = 1/2$; $R = R_{eff} = 0.58/\sqrt{N}$; $R_{int} = 0$

III: $E[x] = 1/2 \times 1/3 + 1/2 \times 2/3 = 0.5$; $R = R_{int} = 0.33/\sqrt{N}$; $R_{eff} = 0$

IV: $E[x] = 1/2 \times 1/4 + 1/2 \times 3/4 = 0.5$; $R = R_{int} = 0.5/\sqrt{N}$; $R_{eff} = 0$

V: $E[x] = 0 \times 1/3 + 1/3 \times 1/2 + 1/3 \times 1 = 0.5$; $R = 0.82/\sqrt{N}$; $R_{int} = 0.41/\sqrt{N}$ 25%; $R_{eff} = 0.71/\sqrt{N}$ 75%

Figure 2.14

that the scoring inefficiency contributes 75% to R in case V, the second worst case of the five.

G. *An Example of Poorly Behaved Tallies*

As an example of poorly behaved tallies, consider Fig. 2.15, which shows plots made by the MCNP plotting code MCPLOT for a particularly nasty problem for point detectors imbedded in a concrete shell with a 14-MeV point isotropic source at the center with no directional biasing. The plots are of the mean of the neutron flux above 12 MeV, relative error, and FOM as a function of the number of particles started (NPS) for a spherical surface flux estimator and the three MCNP types of point detectors at the same radius for both 20 000 and 4×10^6 history problems. The relative error for the surface flux is well behaved in both problems because it decreases as $1/\sqrt{N}$. The same is true for the current tally on the same surface. The FOM for these tallies is approximately a constant, so there is every reason for confidence in the surface tallies.

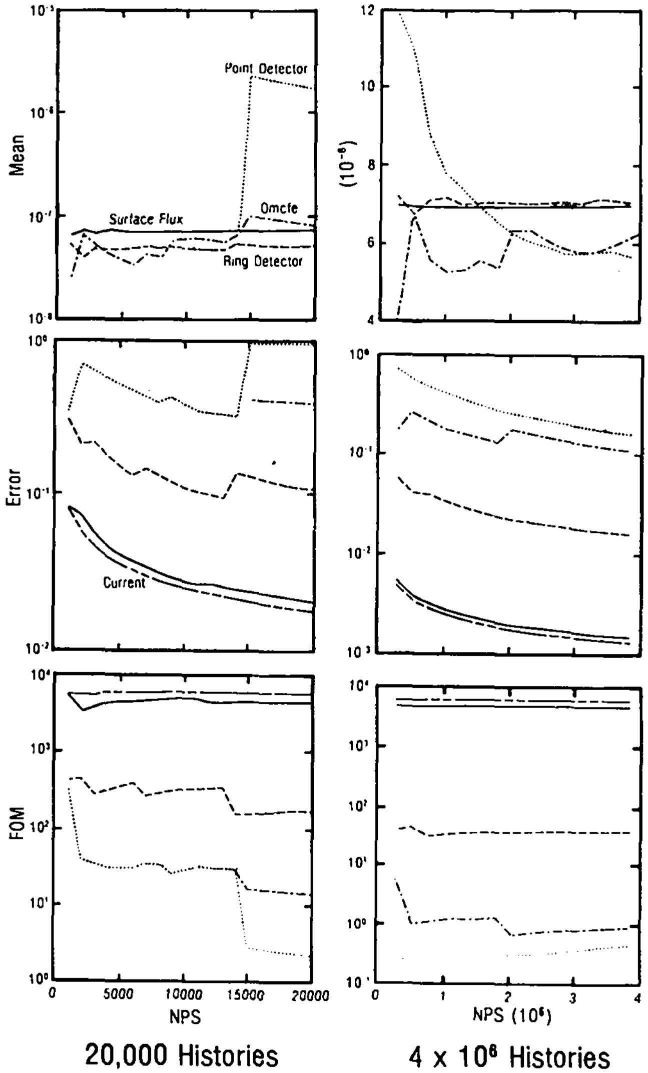


Figure 2.15

The results for the point detectors are quite another matter, however. First consider the 20 000 history problem. Both the standard and OMCFE point flux estimators with appropriate neighborhoods have severe difficulties with the behavior of the tally, relative error, and FOM. The results from these tallies are worthless, as an examination of the relative errors shows. The close tally for the OMCFE is nothing more than good luck. The point detector tally increased by a factor of 30 because of one history that had six collisions near the detector point. Even the ring detector is having difficulties, and the user should be suspicious of the answer based on the factor of 2 drop in the FOM. For 20 000 histories, the ring detector result is about four estimated standard deviations from the correct result. When the relative error for the ring detector reaches

0.05, the results are consistent with the correct answer as the guidelines in Table 2.3 recommend. It is also interesting to observe the large decreases in the FOM as the tally changes from 1D (surface flux) to 2D (ring detector) to 3D (standard and OMCFE point detectors). The FOM decreases by about a factor of 100 for each increase in the problem dimensionality.

The results of the 4 million history problem exhibit some very interesting characteristics. First, the ring detector results for the relative error and FOM have settled down and are finally well behaved. The confidence intervals for the ring detector results appear reasonable for relative errors of less than 0.05 as recommended in Table 2.3. The FOM for this tally for the long problem is a factor of 4 smaller than in the short problem.

The tallies, relative errors, and FOMs for the OMCFE and point detector are not well behaved even though 4 million histories have been run. The FOM is slowly increasing because the relative error is decreasing faster than $1/\sqrt{N}$. This behavior is caused by one large tally contribution and many smaller contributions, transforming R from a constant at small N to its normal $1/\sqrt{N}$ behavior for large N. In between, R will decrease faster than $1/\sqrt{N}$, as can be deduced from Eq. (2.7b) for this case. Based on the recommendations of Table 2.3 and the poor FOM behavior, there is no reason to have confidence in the OMCFE or point detector results.

This ill behaved problem is intended to make the user cautious when using detectors or DXTRAN-related tallies by carefully examining the FOMs in the tally fluctuation charts. This type of behavior can also occur in other types of tallies such as a track length estimator in a long, skinny cell where very infrequently the track length may be large compared to the cell thickness. The user is encouraged to always check the FOMs of all tallies in the tally fluctuation charts at the end of the problem output.

VII. VARIANCE REDUCTION

A. General Considerations

1. Variance Reduction and Accuracy

Variance-reducing techniques in Monte Carlo calculations reduce the computer time required to obtain results of sufficient precision. Note that precision is only one requirement for a good Monte Carlo calculation. Even a zero variance calculation cannot accurately predict natural behavior if other sources of error are not minimized. Factors affecting accuracy are discussed in Chapter 2.VI.

2. Two Choices That Affect Efficiency

The efficiency of a Monte Carlo calculation is affected by two choices, tally type and random walk sampling. The tally choice (for example, point detector flux tally vs surface crossing flux tally) amounts to trying to obtain the best results from the random walks sampled. The chosen random walk sampling amounts to preferentially sampling "important" random walks at the expense of "unimportant" random walks. (A random walk is important if it has a large affect on a tally.) These two choices usually affect the time per history and the history variance as described in Sec. 3 below.

MCNP estimates tallies of the form

$$\langle T\rangle = \int d\vec{r} \int d\vec{v} \int dt \; N(\vec{r},\vec{v},t) \; T(\vec{r},\vec{v},t)$$

by sampling particle histories that statistically produce the correct particle density $N(\vec{r},\vec{v},t)$. The tally function $T(\vec{r},\vec{v},t)$ is zero except where a tally is required. For example, for a surface crossing tally (F1), T will be one on the surface and zero elsewhere. MCNP variance reduction techniques allow the user to try to produce better statistical estimates of N where T is large, usually at the expense of poorer estimates where T is zero or small.

There are many ways to statistically produce $N(\vec{r},\vec{v},t)$. Analog Monte Carlo simply samples the events according to their natural physical probabilities. In this way, an analog Monte Carlo calculation estimates the number of physical particles executing any given random walk. Nonanalog techniques do not directly simulate nature. Instead, nonanalog techniques are free to do anything if N, hence $\langle T\rangle$, is preserved. This preservation is accomplished by adjusting the weight of the particles. The weight can be thought of as the number of physical particles represented by the MCNP particle. Every time a decision is made, the nonanalog techniques require that the expected weight associated with each outcome be the same as in the analog game. In this way, the expected number of physical particles executing any given random walk is the same as in the analog game.

For example, if an outcome "A" is made q times as likely as in the analog game, when a particle chooses outcome "A," its weight must be multiplied by q^{-1} to preserve the expected weight for outcome "A." Let p be the analog probability for outcome "A"; then pq is the nonanalog probability for outcome "A." If w_0 is the current weight of the particle, then the expected weight for outcome "A" in the analog game is $w_0 * p$ and the expected weight for outcome "A" in the nonanalog game is $(w_0/q) * pq$.

MCNP uses three basic types of nonanalog games: (1) splitting, (2) Russian roulette, and (3) sampling from nonanalog probability density

functions. The previous paragraph discusses type 3. Splitting refers to dividing the particle's weight among two or more daughter particles and following the daughter particles independently. Usually the weight is simply divided evenly among k identical daughter particles whose characteristics are identical to the parent except for a factor 1/k in weight (for example, splitting in the weight window) In this case the expected weight is clearly conserved because the analog technique has one particle of weight w_0 at $(\vec{r},\vec{v},t)$, whereas the splitting results in k particles of weight w_0/k at $(\vec{r},\vec{v},t)$. In both cases the outcome is weight w_0 at $(\vec{r},\vec{v},t)$.

Other splitting techniques split the parent particle into k, typically two, differing daughter particles. The weight of the j^{th} daughter represents the expected number of physical particles that would select outcome j from a set of k mutually exclusive outcomes. For example, the MCNP forced collision technique considers two outcomes: (1) the particle reaches a cell boundary before collision, or (2) the particle collides before reaching a cell boundary. The forced collision technique divides the parent particle representing w_0 physical particles into two daughter particles, representing w_1 physical particles that are uncollided and w_2 physical particles that collide. The uncollided particle of weight w_1 is then put on the cell boundary. The collision site of the collided particle of weight w_2 is selected from a conditional distance-to-collision probability density, the condition being that the particle must collide in the cell. This technique preserves the expected weight colliding at any point in the cell as well as the expected weight not colliding. A little simple mathematics is required to demonstrate this.

Russian roulette takes a particle at $(\vec{r},\vec{v},t)$ of weight w_0 and turns it into a particle of weight $w_1 > w_0$ with probability w_0/w_1 and kills it (that is, weight=0) with probability $(1 - (w_0/w_1)$. The expected weight at $(\vec{r},\vec{v},t)$ is $w_1*(w_0/w_1) + (1-(w_0/w_1))*0 = w_0$, the same as in the analog game.

Some techniques use a combination of these basic games and DXTRAN uses all three.

3. Efficiency, Time per History, and History Variance

Recall from Sec. VI that the measure of efficiency for MCNP calculations is the FOM: $FOM = 1/(R^2T)$, where

R^2 = sample relative standard deviation of the mean and

T = computer time for the calculation (in minutes).

Recall further that $R = (S/\sqrt{N})/\bar{x}$, where

S^2 = sample history variance

N = number of particles

$\bar{x}$ = sample mean.

Generally we are interested in obtaining the smallest R in a given time T. The equation above indicates that to decrease R it is desirable to: 1) decrease S and 2) increase N; that is, decrease the time per particle history. Unfortunately, these two goals usually conflict. Decreasing S normally requires more time because better information is required. Increasing N normally increases S because there is less time per history to obtain information. However, the situation is not hopeless. It is often possible either to decrease S substantially without decreasing N too much or to increase N substantially without increasing S too much, so that R decreases.

Many variance reduction techniques in MCNP attempt to decrease R by either producing or destroying particles. Some techniques do both. In general, techniques that produce tracks work by decreasing S (we hope much faster than N decreases) and techniques that destroy tracks work by increasing N (we hope much faster than S increases).

4. Strategy

Successful use of MCNP variance reduction techniques is often difficult, tending to be more art than science. The introduction of the weight window generator has improved things, but the user is still fundamentally responsible for the choice and proper use of variance reducing techniques. Each variance reduction technique has its own advantages, problems, and pecularities. However, there are some general principles to keep in mind while developing a variance reduction strategy.

Not surprisingly, the general principles all have to do with understanding both the physical problem and the variance reduction techniques available to solve the problem. If an analog calculation will not suffice to calculate the tally, there must be something special about the particles that tally. The user should understand the special nature of those particles that tally. Perhaps, for example, only particles that scatter in particular directions can tally. After the user understands why the tallying particles are special, he can select MCNP techniques (or develop his own) that will enhance the number of special particles followed.

After the MCNP techniques are selected the user typically has to supply appropriate parameters to the variance reduction techniques. This is probably more difficult than is the selection of techniques. The first guess at appropriate parameters typically comes either from experience with similar problems or from experience with an analog calculation of the current problem. It is usually better to err on the conservative side; that is, too little biasing rather than too much biasing. After the user has supplied parameters for the variance reduction techniques, a short

Monte Carlo run is done so that the effectiveness of the techniques and parameters can be monitored with the MCNP output.

The MCNP output contains much information to help the user understand the sampling. This information should be examined to ensure that

(1) the variance reduction techniques are improving the sampling of the particles that tally;
(2) the variance reduction techniques are working cooperatively; that is, one is not destructively interfering with another;
(3) the FOM table is not erratic, which would indicate poor sampling; and
(4) there is nothing that looks obviously ridiculous.

Unfortunately, analyzing the output information requires considerable thought and experience. References 1 (tapes 12-15) and 2 show in detail strategies and analysis for a particular problem.

After ascertaining that the techniques are improving the calculation, the user makes a few more short runs to refine the parameters until the sampling no longer improves. The weight window generator can also be turned on to supply information about the importance function in different regions of the phase space. This rather complex subject is described on page 142.

5. Erratic Error Estimates

Erratic error estimates are sometimes observed in MCNP calculations. In fact, the primary reason for the FOM table in the MCNP output is to allow the user to monitor the FOM and the relative error as a function of the number of histories. Except for an analog point detector embedded in a scattering medium with $R_0=0$ (a practice highly discouraged), all MCNP tallies are finite variance tallies. For finite variance tallies the relative error should drop roughly as $\sqrt{N}$, so the FOM should be roughly constant. If it is not, the error estimates should be considered erratic and unreliable, no matter how small the relative error estimate is.

Erratic error estimates occur typically because a high-weight particle tallies from an important region of phase space that has not been well sampled. A high-weight particle in a given region of phase space is a particle whose weight is some nontrivial fraction of <u>all</u> the weight that has tallied from that region because of all previous histories. A good example is a particle that collides very close to a point or ring detector. If not much particle weight has previously collided that close to the detector, the relative error estimate will exhibit a jump for that history.

To avoid high-weight particles in important regions, the user should try to ensure that these regions are well sampled by many particles and try to minimize the weight fluctuation among these particles. Thus the user should try to use biasing techniques that preferentially push particles

into important regions without introducing large weight fluctuations in these regions. The weight window can often be very useful in minimizing weight fluctuations caused by other variance reduction techniques.

If, despite a user's efforts, an erratic error estimate occurs, the user should obtain event logs for those particles causing the estimate to be erratic. The event logs should be studied to learn what is special about these particles. When the special nature of these particles is understood, the user can adjust the variance reduction techniques to sample these particles more often. Thus their weight will be smaller and they will not be as likely to cause erratic estimates. *Under absolutely no circumstances should these particles be discarded or ignored*! The fact that these particles contribute very heavily to the tally indicates that they are important to the calculation and the user should try to sample more of them.

6. Biasing Against Random Walks of Presumed Low Importance

It was mentioned earlier that one should be cautious and conservative when applying variance reduction techniques. Many more people get into trouble by overbiasing than by underbiasing. Note that preferentially sampling some random walks means that some walks will be sampled (for a given computer time) less frequently than they would have been in an analog calculation. Sometimes these random walks are so heavily biased against that very few, or even none, are ever sampled in an actual calculation because not enough particles are run.

That is, suppose that (on average) for every million histories only one track enters cell 23. Further suppose that a typical run is 100 000 histories. On any given run it is unlikely that a track enters cell 23. Now suppose that tracks entering cell 23 turn out to be much more important than a user thought. Maybe 10% of the answer should come from tracks entering cell 23. The user could run 100 000 particles and get 90% of the true tally with an estimated error of 1%, with absolutely no indication that anything is amiss. However, suppose the biasing had been set such that (on average) for every 10 000 particles one track entered cell 23. Then about 10 tracks enter cell 23 and the tally will probably be clobbered by at least one high weight particle and will hover closer to the true tally with a larger and perhaps erratic error estimate. The essential point is this: following ten tracks into cell 23 does not cost much computer time and it helps ensure that the estimated error cannot be low when the tally is seriously in error. Always make sure that all regions of the problem are sampled enough to be certain that they are unimportant.

B. Variance Reduction Techniques

1. Energy Cutoff

The energy cutoff in MCNP is a single user-supplied, problem-wide energy level. Particles are terminated when their energy falls below the energy cutoff. The energy cutoff terminates tracks and thus decreases the time per history. The energy cutoff should be used only when it is known that low-energy particles are either of zero or almost zero importance. An energy cutoff is a Russian roulette game with zero survival probability. A number of pitfalls exist.

a. Remember that low-energy particles can often produce high-energy particles (for example, fission or low-energy neutrons inducing high-energy photons). Thus, even if a detector is not sensitive to low-energy particles, the low-energy particles may be important to the tally.
b. The energy cutoff is the same throughout the entire problem. Often low-energy particles have zero importance in some regions and high importance in others.
c. The answer will be biased (low) if the energy cutoff is killing particles that might otherwise have contributed. Furthermore, as $N \to \infty$ the apparent error will go to zero and therefore mislead the unwary. Serious consideration should be given to two techniques (discussed later), energy roulette and space-energy weight window, that are always unbiased.

One advantage in using the energy cutoff is not directly related to variance reduction but needs to be mentioned. A lower energy cutoff requires more cross sections so that computer memory requirements go up and interactive computing with a timesharing system is degraded.

2. Time Cutoff

The time cutoff in MCNP is a single user-supplied, problem-wide time value. Particles are terminated when their time exceeds the time cutoff. The time cutoff terminates tracks and thus decreases the computer time per history. A time cutoff is a Russian roulette game with zero survival probability. The time cutoff should only be used in time-dependent problems where the last time bin will be earlier than the cutoff.

Although the energy and time cutoffs are similar, more caution must be exercised with the energy cutoff because low energy particles can produce high energy particles, whereas a late time particle cannot produce an early time particle.

3. Geometry Splitting with Russian Roulette

Geometry splitting/Russian roulette is one of the oldest and most widely used variance-reducing techniques in general geometry codes. When used judiciously, it can save substantial machine time. Basically, as particles migrate in an important direction, they are increased in number to provide better sampling, but if they head in the opposite direction, they are killed in an unbiased manner to avoid wasting time on them. Care must be taken, however, not to over-split; this can lead to a substantial wasting of machine time.

Each cell in the problem geometry setup is assigned an importance I by the user on the IMP input card, a number which should be proportional to the estimated value that particles in the cell have for the quantity being scored. When a particle of weight WGT passes from a cell of importance I to one of higher importance I′, the particle is split into a number of identical particles of lower weight according to the following recipe. If I′/I is an integer n ($n \geq 2$), the particle is split into n identical particles, each weighing WGT/n. Weight is preserved in the integer splitting process. If I′/I is not an integer but still greater than 1, splitting is done probabilistically so that the expected number of splits is equal to the importance ratio. Denoting $n = [I'/I]$ to be the largest integer in I′/I, $p = I'/I - n$ is defined. Then with probability p, n+1 particles are used, and with probability $1 - p$, n particles are used. For example, if I′/I is 2.75, 75% of the time split 3 for 1 and 25% of the time split 2 for 1. The weight assigned to each particle is WGT·I/I′, which is the expected weight, to minimize dispersion of weights.

On the other hand, if a particle of weight WGT passes from a cell of importance I to one of lower importance I′, so that $I'/I < 1$, Russian roulette is played and the particle is killed with probability $1-(I'/I)$, or followed further with probability I′/I and weight WGT·I/I′.

Geometry splitting with Russian roulette is very reliable because the weights of all particle tracks are the same in a cell no matter which geometrical path the tracks have taken to get to the cell. (This assumes that no other biasing techniques are used.) The variance of any tally will be reduced when the possible contributors all have the same weight.

The assigned cell importances can have any value – they are not limited to integers. However, adjacent cells with greatly different importances place a greater burden on reliable sampling. Once a sample track population has deteriorated and lost some of its information, large splitting ratios (like 20 to 1) can build the population back up, but nothing can regain the lost information. It is generally better to keep the ratio of adjacent importances small (for example, a factor of a few) and have cells with optical thicknesses in the penetration direction less than about two mean free paths.

Generally, in a deep penetration shielding problem the sample size (that is, number of particles) diminishes to almost nothing in an analog simulation, but splitting helps keep the size built up. A good rule is to keep the population of tracks traveling in the desired direction more or less constant – that is, approximately equal to the number of particles started from the source. A good initial recipe is to split the particles 2 for 1 wherever the track population drops by a factor–of 2. Near–optimum splitting can usually be achieved with only a few iterations, and conversely, additional iterations show strongly diminishing returns. Note that in a combined neutron/photon problem, importances will probably have to be set individually for neutrons and for photons.

One other small facet deserves mention. MCNP never splits into a void although Russian roulette may be played entering a void. Splitting into a void accomplishes nothing except extra tracking because all the split particles must be tracked across the void and they all make it to the next surface. The split should be done according to the importance ratio of the last nonvoid cell departed and the first nonvoid cell entered.

Finally, splitting generally decreases the history variance but increases the time per history, whereas Russian roulette generally increases the history variance but decreases the time per history.

Note four more items:

a. Geometry splitting/Russian roulette works well only in problems that do not have extreme angular dependence. In the extreme case, splitting/Russian roulette can be useless if no particles ever enter an important cell where the particles can be split.
b. Geometry splitting/Russian roulette will preserve weight variations. The technique is "dumb" in that it never looks at the particle weight before deciding appropriate action. An example is geometry splitting/Russian roulette used with source biasing.
c. Geometry splitting/Russian roulette are turned on or off <u>together</u>.
d. Particles are killed immediately upon entering a zero importance cell. This acts as a geometry cutoff.

4. Energy Splitting/Roulette

Energy splitting and roulette are typically used together, but the user can specify only one if desired. Energy splitting/roulette is independent of spatial cell. If the problem has a space–energy dependence, the space–energy dependent weight window is normally a better choice.

<u>a. Splitting</u>: In some cases, particles are more important in some energy ranges than in others. For example, it may be difficult to calculate the number of ^{235}U fissions because the thermal neutrons are also being captured and not enough thermal neutrons are available for a reliable

sample. In this case, once a neutron falls below a certain energy level it can be split into several neutrons with an appropriate weight adjustment. A second example involves the effect of fluorescent emission after photoelectric absorption. With energy splitting, the low-energy photon track population can be built up rather than rapidly depleted, as would occur naturally with the high photoelectric absorption cross section.

Particles can be split as they move up or down in energy at up to five different energy levels.

Energy splitting may increase as well as decrease tally variances. Currently, the MCNP weight cutoff game does not take into account whether a particle has undergone energy splitting or not.

With only a minor modification to MCNP, the mechanics for energy splitting may be used for time splitting.

b. Russian roulette: In many cases the number of tracks increases with decreasing energy, especially neutrons near the thermal energy range. These tracks can have many collisions requiring appreciable computer time. They may be important to the problem and cannot be completely eliminated with an energy cutoff, but their number can be reduced by playing a Russian roulette game to reduce their number and computer time.

If a track's energy drops through a prescribed energy level, the roulette game is played, based on the input value of the survival probability. If the game is won, the track's history is continued, but its weight is increased by the reciprocal of the survival probability to conserve weight.

5. Implicit Capture and Weight Cutoff

a. Implicit Capture: "Implicit capture," "survival biasing," and "absorption by weight reduction" are synonymous. Implicit capture is a variance reduction technique applied in MCNP after the collision nuclide has been selected. Let

σ_{ti} = total microscopic cross section for nuclide i and

σ_{ai} = microscopic absorption cross section for nuclide i.

When implicit capture is used rather than sampling for absorption with probability σ_{ai}/σ_{ti}, the particle always survives the collision and is followed with new weight: WGT * $(1 - \sigma_{ai}/\sigma_{ti})$. Implicit capture is a splitting process where the particle is split into absorbed weight (which need not be followed further) and surviving weight.

Implicit capture can also be done along a flight path rather than at collisions when a special form of the exponential transform is used. See page 144 for details.

Two advantages of implicit capture are

1. a particle that has finally, against considerable odds, reached the tally region is not absorbed just before a tally is made, and

2. the history variance, in general, decreases when the surviving weight (that is, 0 or WGT) is not sampled, but an expected surviving weight is used instead (but see weight cutoff discussion).

Two disadvantages are

1. a fluctuation in particle weight is introduced, and

2. the time per history is increased (but see weight cutoff discussion, next section).

b. Weight Cutoff: In weight cutoff, Russian roulette is played if a particle's weight drops below a user-specified weight cutoff. The particle is either killed or its weight is increased to a user-specified level. The weight cutoff was originally envisioned for use with geometry splitting/Russian roulette and implicit capture. Because of this intent,

1. The weight cutoffs in cell j depend not only on WC1 and WC2 on the CUT card, but also on the cell importances.

2. Implicit capture is always turned on (except in detailed photon physics) whenever a nonzero WC1 is specified.

Referring to item 1 above, the weight cutoff is applied when the particle's weight falls below R_j*WC2, where R_j is the ratio of the source cell importance (IMP card) to cell j's importance. With probability WGT/(WC1*R_j) the particle survives with new weight WC1*R_j; otherwise the particle is killed.

As mentioned earlier, the weight cutoff game was originally envisioned for use with geometry splitting and implicit capture. To illustrate the need for a weight cutoff when using implicit capture, consider what can happen without a weight cutoff. Suppose a particle is in the interior of a very large medium and there are no time nor energy cutoffs. The particle will go from collision to collision, losing a fraction of its weight at each collision. Without a weight cutoff, a particle's weight would eventually be too small to be representable in the computer, at which time an error would occur. If there are other loss mechanisms (for example, escape, time cutoff, or energy cutoff), the particle's weight will not decrease indefinitely, but the particle may take an unduly long time to terminate.

Weight cutoff's dependence on the importance ratio can be easily understood if one remembers that the weight cutoff game was originally designed to solve the low-weight problem sometimes produced by implicit capture. In a high-importance region, the weights are low by design, so it makes no sense to play the same weight cutoff game in high- and low-importance regions.

c. Comments: Many techniques in MCNP cause weight change. The weight cutoff was really designed with geometry splitting and implicit capture in mind. Care should be taken in the use of other techniques.

Weight cutoff games are unlike time and energy cutoffs. In time and energy cutoffs, the random walk is always terminated when the threshold is crossed. Potential bias may result if the particle's importance was not zero. A weight cutoff (weight roulette would be a better name) does not bias the game because the weight is increased for those particles that survive.

Setting the weight cutoff is not typically an easy task and requires thought and experimentation. Essentially the user must guess what weight is worth following and start experimenting with weight cutoffs in that vicinity.

6. Forced Collisions

The forced collision method is a variance reduction scheme that increases sampling of collisions in specified cells. Because detector contributions and DXTRAN particles arise only from collisions and the source, it is often useful in certain cells to increase the number of collisions that may produce large detector contributions or large weight DXTRAN particles. Sometimes it is desired to sample collisions in a relatively thin cell (a fraction of a mean free path) to improve the estimate of quantities like a reaction rate or energy deposition or to cause collisions that are important to some other part of the problem.

The forced collision method splits particles into collided and uncollided parts. The collided part is forced to collide within the current cell. The uncollided part exits the current cell without collision and is stored in the bank until later when its track is continued at the cell boundary. Its weight is W_o*exp(-DLS*QPL), where W_o = current particle weight before forced collision, DLS = distance to cell surface in the particle's direction, and QPL = macroscopic cross section of the cell material. That is, the uncollided part is the current particle weight multiplied by the probability of exiting the cell without collision.

The collided part has weight W_o*(1-exp(-DLS*QPL)), which is the current particle weight multiplied by the probability of colliding in the cell. The uncollided part is always produced. The collided part may be produced only a fraction f of the time, in which case the collided weight is W_o*(1-exp(-DLS*QPL))/f. This is useful when several forced collision cells are adjacent or when too much time is spent producing and following forced collision particles.

The collision distance is sampled as follows. If P(x) is the unconditional probability of colliding within a distance x, P(x)/P(DLS) is the conditional probability of colliding within a distance x given that a

collision is known to occur within a distance DLS. Thus the position PMF of the collision must be sampled on the interval 0 < PMF < DLS within the cell according to ξ = P(PMF)/P(DLS), where P(PMF) = 1-exp(-PMF*QPL) and ξ is a random number. Solving for PMF, one obtains

$$\mathrm{PMF} = -\frac{1}{\mathrm{QPL}} \ln\left\{1-\xi\left[1-\exp(-\mathrm{DLS}*\mathrm{QPL})\right]\right\}.$$

Because a forced collision usually yields a collided particle having a relatively small weight, care must be taken with the weight-cutoff game (page 126), the weight-window game (page 140), and subsequent collisions of the particle within the cell. The weight window game is not played on the surface of a forced collision cell that the particle is entering. For collisions inside the cell the user has two options.

Option 1: (negative entry for the cell on the forced collision card.) After the forced collision, subsequent collisions of the particle are sampled normally. The weight cutoff game is turned off and detector contributions and DXTRAN particles are made before the weight window game is played. If weight windows are used, they should be set to the weight of the collided particle weight or set to zero if detector contributions or DXTRAN particles are desired.

Option 2: (positive entry for the cell on the forced collision card.) After the forced collision, detector contributions or DXTRAN particles are made and either the weight cutoff or weight window game is played. Surviving collided particles undergo subsequent forced collisions. If weight windows are used, they should bracket the weight of particles entering the cell.

7. DXTRAN

DXTRAN is typically used when a small region is being inadequately sampled because particles have a very small probability of scattering toward that region. To ameliorate this situation, the user can specify a DXTRAN sphere (in the input file) that encloses the small region. Upon collision (or exiting the source) outside the sphere, DXTRAN creates a special "DXTRAN particle" and deterministically scatters it toward the DXTRAN sphere and deterministically transports it, without collision, to the surface of the DXTRAN sphere. The collision itself is otherwise treated normally, producing a non-DXTRAN particle that is sampled in the normal way, with no reduction in weight. However, the non-DXTRAN particle is killed if it tries to enter the DXTRAN sphere. DXTRAN uses a combination of splitting, Russian roulette, and sampling from a nonanalog probability density function.

The subtlety about DXTRAN is how the extra weight created for the DXTRAN particles is balanced by the weight killed as non-DXTRAN particles cross the DXTRAN sphere. The non-DXTRAN particle is followed without any weight correction, so if the DXTRAN technique is to be unbiased, the extra weight put on the DXTRAN sphere by DXTRAN particles must somehow (on average) balance the weight of non-DXTRAN particles killed on the sphere.

a. DXTRAN Viewpoint 1: One can view DXTRAN as a splitting process (much like the forced collision technique) wherein each particle is split upon departing a collision (or source point) into two distinct pieces:

1. the weight that does not enter the DXTRAN sphere on the next flight either because the particle is not pointed toward the DXTRAN sphere or because the particle collides before reaching the DXTRAN sphere, and
2. the weight that enters the DXTRAN sphere on the next flight.

Let w_o be the weight of the particle before exiting the collision, let p_1 be the analog probability that the particle does not enter the DXTRAN sphere on its next flight, and let p_2 be the analog probability that the particle does enter the DXTRAN sphere on its next flight. The particle must undergo one of these mutually exclusive events, thus $p_1 + p_2 = 1$. The expected weight not entering the DXTRAN sphere is $w_1 = w_o p_1$, and the expected weight entering the DXTRAN sphere is $w_2 = w_o p_2$. Think of DXTRAN as deterministically splitting the original particle with weight w_o into two particles, a non-DXTRAN (particle 1) particle of weight w_1 and a DXTRAN (particle 2) particle of weight w_2. Unfortunately, things are not quite that simple.

Recall that the non-DXTRAN particle is followed with unreduced weight w_o rather than weight $w_1 = w_o p_1$. The reason for this apparent discrepancy is that the non-DXTRAN particle (#1) plays a Russian roulette game. Particle 1's weight is increased from w_1 to w_o by playing a Russian roulette game with survival probability $p_1 = w_1/w_o$. The reason for playing this Russian roulette game is simply that p_1 is not known, so assigning weight $w_1 = p_1 w_o$ to particle 1 is impossible. However, it is possible to play the Russian roulette game without explicitly knowing p_1. It is not magic, just slightly subtle.

The Russian roulette game is played by sampling particle 1 normally and keeping it only if it does not enter (on its next flight) the DXTRAN sphere; that is, particle 1 survives (by definition of p_1) with probability p_1. Similarly, the Russian roulette game is lost if particle 1 enters (on its next flight) the DXTRAN sphere; that is, particle 1 loses the roulette with probability p_2. To restate this idea, with probability p_1, particle 1 has weight w_o and does not enter the DXTRAN sphere and with probability p_2, the particle enters the DXTRAN sphere and is killed. Thus, the expected weight not entering the DXTRAN sphere is $w_o p_1 + 0 * p_2 = w_1$, as desired.

So far, this discussion has concentrated on the non-DXTRAN particle and ignored exactly what happens to the DXTRAN particle. The sampling of the DXTRAN particle will be discussed after a second viewpoint on the non-DXTRAN particle.

b. DXTRAN Viewpoint 2: This second way of viewing DXTRAN does not see DXTRAN as a splitting process but as an accounting process in which weight is both created and destroyed on the surface of the DXTRAN sphere. In this view, DXTRAN estimates the weight that should go to the DXTRAN sphere upon collision and creates this weight on the sphere as DXTRAN particles. If the non-DXTRAN particle does not enter the sphere, its next flight will proceed exactly as it would have without DXTRAN, producing the same tally contributions and so forth. However, if the non-DXTRAN particle's next flight attempts to enter the sphere, the particle must be killed or there would be (on average) twice as much weight crossing the DXTRAN sphere as there should be because the weight crossing the sphere has already been accounted for by the DXTRAN particle.

c. The DXTRAN Particle: Although the DXTRAN particle does not confuse people nearly as much as the non-DXTRAN particle, the DXTRAN particle is nonetheless subtle.

The most natural approach for scattering particles toward the DXTRAN sphere would be to sample the scattering angle (μ=cos θ) proportional to the analog density. This is not done because it is too much work to sample proportional to the analog density and because it is sometimes useful to bias the sampling.

To sample μ in an unbiased fashion when it is known that $\eta_o \le \mu \le 1$, one samples the conditional density

$$P_{con}(\mu) = P(\mu)/\int_1^{\eta_o} P(\mu)d\mu \qquad \text{(where } \eta_o \text{ defines the cone)}$$

and multiplies the weight by $\int_1^{\eta_o} P(\mu)d\mu)$, the probability of scattering into the cone (see Fig. 2.16). However, it is too much work to calculate the above integral for each collision. Instead, an arbitrary density function $P_{arb}(\mu)$ is sampled and the weight is multiplied by

$$\frac{P_{con}(\mu)}{P_{arb}(\mu)} = \frac{P(\mu)}{P_{arb}(\mu)\int_1^{\eta_o} P(\mu)d\mu} \quad .$$

The total weight multiplication is the product of the fraction of the weight scattering into the cone, $\int_1^{\eta_o} P(\mu)d\mu$, and the weight correction for sampling $P_{arb}(\mu)$ instead of $P_{con}(\mu)$. Thus, the weight correction on

scattering is $P(\mu)/P_{arb}(\mu)$. This result can be obtained more directly, but the other derivation does not shed light on why $P_{con}(\mu)$ is not sampled.

Because $P_{arb}(\mu)$ is arbitrary, MCNP can choose a scheme that samples μ from a two-step density that favors particles within the larger μ interval. In fact, the inner DXTRAN sphere has to do only with this arbitrary density and is not essential to the DXTRAN concept. The DXTRAN particles are always created on the outside DXTRAN sphere, with the inner DXTRAN sphere defining only the boundary between the two steps in the density function.

After $\mu = \cos\theta$ has been chosen, the azimuthal angle φ is sampled uniformly on $[0,2\pi]$; this completes the scattering. Recall, however, that the DXTRAN particle arrives at the DXTRAN sphere without collision. Thus the DXTRAN particle also has its weight multiplied by the negative exponential of the optical path between the collision site and the sphere.

d. Inside the DXTRAN Sphere: So far, only collisions outside the DXTRAN sphere have been discussed. At collisions inside the DXTRAN sphere, the DXTRAN game is not played because first, the particle is already in the desired region, and second, it is impossible to define the angular cone of Fig. 2.16. If there are several DXTRAN spheres and the collision occurs in sphere i, DXTRAN will be played for all spheres except sphere i.

e. Terminology - Real particle and Pseudoparticle: In some X-6 documentation, the DXTRAN particle is called a pseudoparticle and the non-DXTRAN particle is called the original or real particle. The terms "real particle" and "pseudoparticle" are potentially misleading. Both particles are equally real: both execute random walks, both carry nonzero weight, and both contribute to tallies. The only sense in which the DXTRAN particle should be considered "pseudo" or "not real" is during creation. A DXTRAN particle is created on the DXTRAN sphere, but creation involves determining what weight the DXTRAN particle should have upon creation. Part of this weight determination requires calculating the optical path between the collision site and the DXTRAN sphere. MCNP determines the optical path by tracking a pseudoparticle from the collision site to the DXTRAN sphere. This pseudoparticle is deterministically tracked to the DXTRAN sphere simply to determine the optical path. No distance to collision is sampled, no tallies are made, and no records of the pseudoparticle's passage are kept (for example, tracks entering). In contrast, once the DXTRAN particle is created at the sphere's surface, the particle is no longer a pseudoparticle. The particle has real weight, executes random walks, and contributes to tallies.

f. DXTRAN Details: To explain how the scheme works, consider the neighborhood of interest to be a spherical region surrounding a

designated point in space. In fact, consider two spheres of arbitrary radii about the point $P_o = (x_o, y_o, z_o)$. Further, assume that the particle having direction (u,v,w) collides at the point $P_1 = (x,y,z)$, as shown in Fig. 2.16.

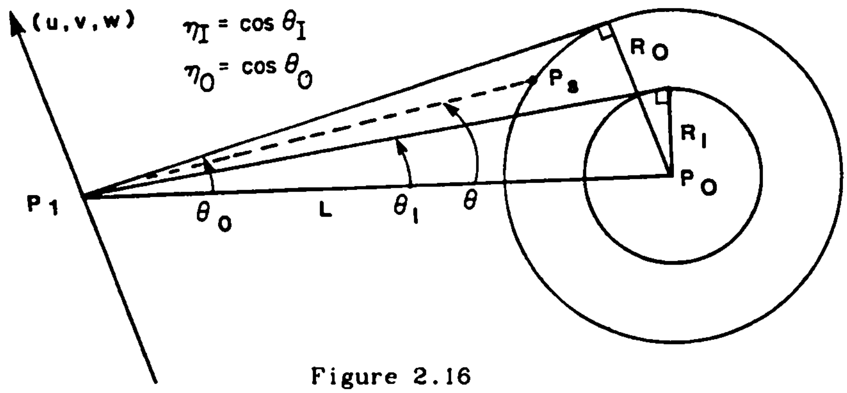


Figure 2.16

The quantities θ_I, θ_O, η_I, η_O, R_I, and R_O are defined in the figure. Thus L, the distance between the collision point and center of the spheres, is

$$L = [(x-x_o)^2 + (y-y_o)^2 + (z-z_o)^2]^{1/2} \quad .$$

On collision, a DXTRAN particle is placed at a point on the outer sphere of radius R_O as described below. Provision is made for biasing the contributions of these DXTRAN particles on the outer sphere within the cone defined by the inner sphere. The weight of the DXTRAN particle is adjusted to account for the probability of scattering in the direction of the point on the outer sphere and traversing the distance with no further collision.

The steps in sampling the DXTRAN particles are outlined:

$$\eta_I = \cos\,\theta_I = (L^2 - R_I^2)^{1/2}/L$$

$$\eta_O = \cos\,\theta_O = (L^2 - R_O^2)^{1/2}/L$$

Sample $\eta = \eta_I + \xi(1 - \eta_I)$ uniformly in $(\eta_I, 1)$ with probability

$$Q(1 - \eta_I)/[Q(1 - \eta_I) + \eta_I - \eta_O]$$

and with probability

$$(\eta_I - \eta_O)/[Q(1 - \eta_I) + \eta_I - \eta_O]$$

sample $\eta = \eta_O + \xi(\eta_I - \eta_O)$ uniformly in (η_O, η_I). The quantity Q (equal to 5 in MCNP) is a factor that measures the importance assigned to scattering in the inner cone relative to the outer cone. Therefore, Q is also the ratio of weights for particles put in the two different cones.

With $\eta = \cos\,\theta$ chosen, a new direction (u′,v′,w′) is computed by

considering the rotation through the polar angle θ (and a uniform azimuthal angle φ) from the reference direction

$$\left(\frac{x_o - x}{L}, \frac{y_o - y}{L}, \frac{z_o - z}{L} \right).$$

The particle is advanced in the direction (u′,v′,w′) to the surface of the sphere of radius R_0. The new DXTRAN particle with appropriate direction and coordinates is banked. The weight of the DXTRAN particle is determined by multiplying the weight of the particle at collision by

$$\nu \cdot \frac{P(\mu)\ \{Q(1-\eta_1)+\eta_1-\eta_0\}\ e^{-\int_{P_1}^{P_s} \Sigma_t(s)ds}}{Q}, \quad \eta_1 \le \eta \le 1$$

and

$$\nu \cdot P(\mu)\ \{Q(1-\eta_1)+\eta_1-\eta_0\}\ e^{-\int_{P_1}^{P_s} \Sigma_t(s)ds}, \quad \eta_0 \le \eta \le \eta_1,$$

where

μ = uu′ + vv′ + ww′

$P(\mu)$ = scattering probability density function for scattering through the angle $\cos^{-1} \mu$ in the lab system for the event sampled at (x,y,z)

ν = number of neutrons emitted from the event

$e^{-\int_{P_1}^{P_s} \Sigma_t(s)ds}$ = the attenuation along the line between P_1(x,y,z) and P_s, the point on the sphere where the particle is placed.

In arriving at the weight factor, note that the density function for sampling η is given by

$$\begin{array}{ll} Q/[Q(1-\eta_1)+\eta_1-\eta_0], & \eta_1 < \eta \le 1 \\ 1/[Q(1-\eta_1)+\eta_1-\eta_0], & \eta_0 \le \eta \le \eta_1 \end{array} .$$

Thus the weight of the DXTRAN particle is the weight of the incoming particle at P_1 modified by the ratio of the probability density function for actually scattering from P_1 and arriving at P_s without collision to the density function actually sampled in choosing P_s. Therefore, particles in

the outer cone have weights Q=5 times higher than the weights of similar particles in the inner cone.

The attenuation is calculated at the energy obtained by scattering through the angle μ. The energy is uniquely determined from μ in elastic scattering (and also in level scattering), whereas for other nonelastic events, the energy is sampled from the corresponding probability density function for energy, and may not depend on μ.

g. Auxiliary Games for DXTRAN: The major disadvantage to DXTRAN is the extra time consumed following DXTRAN particles with low weights. Three special games can control this problem:

1. DXTRAN weight cutoffs,
2. DXC games, and
3. DD game.

Particles inside a DXTRAN sphere are not subject to the normal MCNP weight cutoff or weight window game. Instead DXTRAN spheres have their own weight cutoffs, which allow the user to roulette DXTRAN particles that, for one reason or another, do not have enough weight to be worth following.

Sometimes low-weighted DXTRAN particles occur because of collisions many free paths from the DXTRAN sphere. The exponential attenuation causes these particles to have extremely small weights. The DXTRAN weight cutoff will roulette these particles only after much effort has been spent producing them. The DXC cards are cell dependent and allow DXTRAN contributions to be taken only some fraction of the time. That is, the user specifies a probability p_i that a DXTRAN particle will be produced at a given collision or source sampling in cell i. The DXTRAN result remains unbiased because when a DXTRAN particle is produced its weight is multiplied by p_i^{-1}. (The non-DXTRAN particle is treated exactly as before, unaffected unless it enters the DXTRAN sphere, whereupon it is killed.) To see the utility, suppose that the DXTRAN weight cutoff was immediately killing 99% of the DXTRAN particles from cell i. Only 1% of the DXTRAN particles survive anyway, so it might be appropriate to produce only 1% (p_i=.01) and have these not be killed immediately by the DXTRAN weight cutoff. Or the p_i's can often be set such that all DXTRAN particles from all cells are created on the DXTRAN sphere with roughly the same weight. Choosing the p_i's is often difficult and the method works well typically when the material exponential attenuation is the major source of the weight fluctuation.

Often the weight fluctuation arises because the probability ($P(\mu)$) of scattering toward the DXTRAN sphere varies greatly, depending on what nuclide is hit and what the collision orientation is with respect to the DXTRAN sphere. For example, consider a highly forward-peaked scattering probability density. If the DXTRAN sphere were close to the particle's precollision direction, $P(\mu)$ will be large; if the DXTRAN sphere were at

105° to the precollision direction, P(μ) will be small. The DD game can be used to eliminate the weight fluctuation on the DXTRAN sphere caused by these geometry effects as well as the material exponential attenuation effects.

The DD game works by selectively rouletting the DXTRAN pseudoparticles during creation, depending on the DXTRAN particles' weight compared to some reference weight. This is the same game that is played on detector contributions, and is described on page 97. The reference weight can be either a fraction of the average of previous DXTRAN particle weights or a user input reference weight. Recall that a DXTRAN particle's weight is computed by multiplying the exit weight of the non-DXTRAN particle by a weight factor having to do with the scattering probability and the negative exponential of the optical path between collision site and DXTRAN sphere. The optical path is computed by tracking a pseudoparticle from collision to DXTRAN sphere. The weight of the pseudoparticle is monotonically decreasing, so the DD game compares the pseudoparticle's weight at the collision site and upon exiting each cell against the reference weight. A roulette game is played when the pseudoparticle's weight falls below the reference weight. In this way the DD card stops tracking a pseudoparticle as soon as the weight becomes inconsequential, saving time by eliminating subsequent tracking.

h. Final Comments:

1. DXTRAN should be used carefully in optically thick problems. Do not rely on DXTRAN to do penetration.
2. If the source is user supplied, some provision must be made for obtaining the source contribution to particles on the DXTRAN sphere.
3. Extreme care must be taken when more than one DXTRAN sphere is in a problem. Cross-talk between spheres can result in extremely low weights and an explosion in particle tracks.
4. Never put a zero on the DXC card. This will bias the calculation by not creating DXTRAN particles but still killing the non-DXTRAN particle if it enters the DXTRAN sphere.
5. Usually there should be a rough balance of weights in the Ledger of Particle Creation and Loss.
6. DXTRAN cannot be used with reflecting surfaces for the same reasons that point detectors cannot be used with reflecting surfaces. See page 96 for further explanation.
7. Both DXTRAN and point detectors track pseudoparticles to a point. Therefore, most of the discussion about detectors applies to DXTRAN. Refer to the section on detectors, page 87, for more information.

8. Source Variable Biasing

Provision is made for biasing the MCNP sources in any or all of the source variables specified. This allows the production of more source particles, with suitably reduced weights, in the more important regimes of each variable. For example, one may start more "tracks" at high energies and in strategic directions in a shielding problem while correcting the distribution by altering the weights assigned to these tracks. Sizable variance reductions may result from such biasing of the source. Source biasing samples from a nonanalog probabilty density function.

If negative weight cutoff values are used on the CUT card, the weight cutoff is made relative to the lowest value of source particle weight generated by the biasing schemes.

a. Biasing by Specifying Explicit Sampling Frequencies: The SB input card determines source biasing for a particular variable by specifying the frequency at which source particles will be produced in the variable regime. If this fictitious frequency does not correspond to the fraction of actual source particles in a variable bin, the corrected weight of the source particles in a particular bin is determined by the ratio of the actual frequency (defined on the SP card) divided by the fictitious frequency (defined on the SB card) except for the lin–lin interpolation where it is defined to be the ratio of the actual to fictitious frequency evaluated at the exact value of the interpolated variable. The total weight of particles started in a given energy interval is thus conserved.

b. Biasing by Standard Prescription: Source biasing may also be performed by using certain built–in prescriptions similar in principle to built–in analytic source distributions. These biasing options are detailed in the sections below for the appropriate source variables. The SB card input is analogous to that of an SP card for an analytic source distribution; that is, the first entry is a negative prescription number for the type of biasing required, followed by one or more optional user–specified parameters, which are discussed in the following sections.

1) *Direction Biasing*: The source may be biased directionally by sampling from a continuous exponential function and by using cones of fixed size and starting a fixed fraction of particles within each cone. The user can bias particles in any arbitrary direction or combination of directions.

In general, continuous biasing is preferable to fixed cone biasing because cone biasing can cause problems from the discontinuities of source track weight at the cone boundaries. However, if the cone parameters (cone size and fraction of particles starting in the cone) are optimized through

a parameter study and you understand the paths that tracks take to contribute to tallies, fixed cone biasing can sometimes outperform continuous biasing. Unfortunately, it is usually time consuming (both human and computer) and difficult to arrive at the necessary optimization.

Source directional biasing can be sampled from an exponential probability density function $p(\mu) = Ce^{K\mu}$, where C is a norming constant equal to $K/(e^{K}-e^{-K})$ and $\mu = \cos\theta$, where θ is an angle relative to the biasing direction. K is typically about 1; K = 3.5 defines the ratio of weight of tracks starting in the biasing direction to tracks starting in the opposite direction to be 1097. This ratio is equal to e^{2K}.

Table 2.6 may help to give you a feel for the biasing parameter K.

Table 2.6
Exponential Biasing Parameter

K	Cumulative Probability	Theta	Weight
.01	0	0	0.990
	.25	60	0.995
	.50	90	1.000
	.75	120	1.005
	1.00	180	1.010
1.0	0	0	.432
	.25	42	.552
	.50	64	.762
	.75	93	1.230
	1.00	180	3.195
2.0	0	0	.245
	.25	31	.325
	.50	48	.482
	.75	70	.931
	1.00	180	13.40
3.5	0	0	.143
	.25	23	.190
	.50	37	.285
	.75	53	.569
	1.00	180	156.5

From this table for K=1, we see that half the tracks start in a cone of 64° opening about the axis, and the weight of tracks at 64° is 0.762 times the unbiased weight of source particles. K = 0.01 is almost equivalent to no biasing, and K = 3.5 is very strong.

Cone directional biasing can be invoked by specifying cone cosines on the SI card and the desired biasing probabilities on the SB card. An isotropic distribution is assumed unless some other distribution is specified on an SP card. Both histogram and linear interpolation can be used. For example, consider the following case:

SIn -1 ν 1

SBn 0 p_1 p_2 The direction cosine relative to the reference direction, say v, is sampled uniformly within the cone $\nu < v < 1$ with probability p_2 and within $-1 < v < \nu$ with the complementary probability p_1. The weights assigned are $W(1 - \nu)/(2p_2)$ and $W(1 + \nu)/(2p_1)$, respectively. Note that for a very small cone defined by ν and a high probability $p_2 \gg p_1$ for being within the cone, the few source particles generated outside the cone will have a very high weight which can severely perturb a tally. The sampling of the direction cosines azimuthal to the reference axis is not biased.

2) *Covering Cylinder Extent Biasing*: This biasing prescription allows the automatic spatial biasing of source particles in a cylindrical-source-covering-volume along the axis of the cylinder. Such biasing can aid in the escape of source particles from optically thick source regions and thus represents a variance reduction technique.

3) *Covering Sphere Radial Biasing*: This biasing prescription allows for the radial spatial biasing of source particles in either a spherical or cylindrical source covering volume. Like the previous example of extent biasing, this biasing can be used to aid in the escape of source particles from optically thick source regions.

c. Biasing Standard Analytic Source Functions:[36] The preceding examples discuss the biasing of source variables by either input of specific sampling frequencies corresponding to SP card entries or by standard analytic biasing functions. A third biasing category can be used in conjunction with standard analytic source probability functions (for example, a Watt fission spectrum).

A negative entry on an SP card, that is,

SPn $-$i a b

causes MCNP to sample source distribution n from probability function i

with input variables a,b,... . Sampling schemes are typically unbiasable. For example, for

SPn -5 a

the evaporation spectrum f(E) = C E exp(-E/a) is sampled according to the sampling prescription E = -a log (r1*r2), where r1 and r2 are random numbers. Biasing this sampling scheme is usually very difficult or impossible. Fortunately, there is an approximate method available in MCNP for biasing any arbitrary probability function.[36] The code approximates the function as a table, then uses the usual SB card biasing scheme to bias this approximate table function. The user merely inputs a coarse bin structure to govern the bias and the code adds up to 300 additional equiprobable bins to assure accuracy. For example, suppose we wish to sample the function

f(E) = C E exp(-E/a)

and suppose that we want half the source to be in the range .005 < E < .1 and the other half to be in the range .1 < E < 20. Then the input is

```
SPn -5 a
SIn .005 .1 20
SBn C 0 .5 1 .
```

MCNP breaks up the function into 150 equiprobable bins below E = .1 and 150 more equiprobable bins above E = .1. Half the time E is chosen from the upper set of bins and half the time it is chosen from the lower set. Particles starting from the upper bins have a different weight than that of particles starting from the lower bins to adjust for the bias, and a detailed summary of all this is printed when the PRINT option is used.

Note that in the above example the probability distribution function is truncated below E = .005 and above E = 20. MCNP prints out how much of the distribution is lost in this manner and reduces the weight accordingly.

Note also that it is possible for the user to choose a foolish biasing scheme. For example,

```
SPn -5 a
SIn .005 297I .1 20
SBn 0 1 298R
```

causes each of the 299 bins to be chosen with equal probability. This would be all right except that since there are never more than 300 equiprobable bins, this allocates only 1 equiprobable bin per user-supplied

bin. The single equiprobable bin for $.1 < E < 20$ is inadequate to describe the distribution function over this range. Thus the table no longer approximates the function and the source will be sampled erroneously. MCNP issues an error message whenever too much of the source distribution is allocated to a single equiprobable bin in order to alert users to a poor choice of binning which might inadequately represent the function. The coarse bins used for biasing should be chosen so that the probability function is roughly equally distributed among them.

9. Weight Window

The weight window (Fig. 2.17) is a space–energy–dependent splitting and Russian roulette technique. For each space–energy phase space cell, the user supplies a lower weight bound. The upper weight bound is a user–specified multiple of the lower weight bound. These weight bounds define a window of acceptable weights. If a particle is below the lower weight bound, Russian roulette is played and the particle's weight is either increased to a value within the window or the particle is terminated. If a particle is above the upper weight bound, it is split so that all the split particles are within the window. No action is taken for particles within the window.

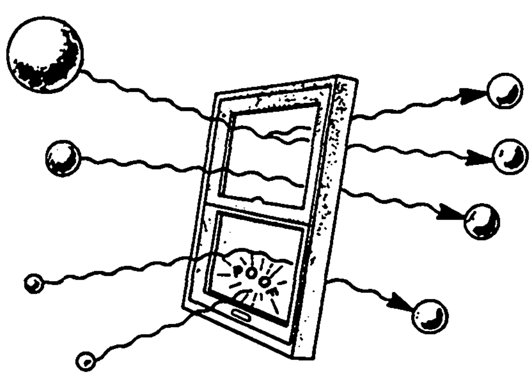


Figure 2.17

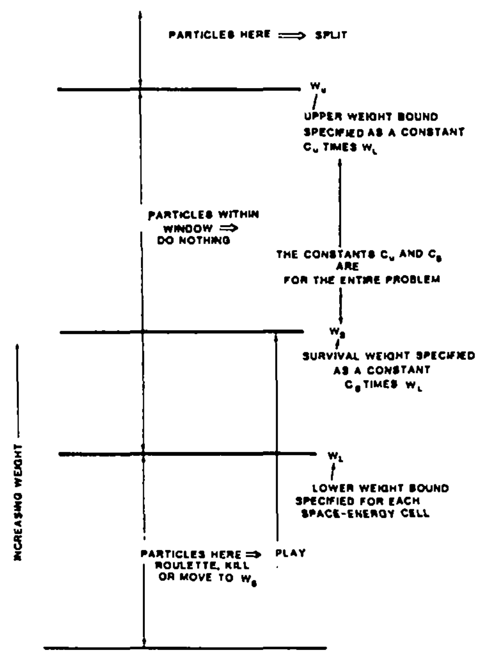


Figure 2.18

Figure 2.18 is a more detailed picture of the weight window. Three important weights define the weight window in a space-energy cell

1. W_L, the lower weight bound,
2. W_S, the survival weight for particles playing roulette, and
3. W_U, the upper weight bound.

The user specifies W_L for each space-energy cell on WFN cards. W_S and W_U are calculated using two problem-wide constants, C_S and C_U (entries on the WDWN card), as $W_S = C_S W_L$ and $W_U = C_U W_L$. Thus all cells have an upper weight bound C_U times the lower weight bound and a survival weight C_S times the lower weight bound.

Although the weight window can be effective when used alone, it was designed for use with other biasing techniques that introduce a large variation in particle weight. In particular, a particle may have several "unpreferred" samplings, each of which will cause the particle weight to be multiplied by a weight factor substantially larger than one. Any of these weight multiplications by itself is usually not serious, but the cumulative weight multiplications can seriously degrade calculational efficiency. Worse, the error estimates may be misleading until enough extremely high-weight particles have been sampled. Monte Carlo novices are prone to be mislead because they do not have enough experience reading and interpreting the summary information on the sampling supplied by MCNP. Hence, a novice may put more faith in an answer than is justified.

Although it is impossible to eliminate all pathologies in Monte Carlo calculations, a properly specified weight window goes far toward eliminating the pathology referred to in the preceding paragraph. As soon as the weight gets above the weight window, the particle is split and subsequent weight multiplications will thus be multiplying only a fraction of the particle's weight (before splitting). Thus, it is hard for the tally to be clobbered by a particle of extremely large weight. In addition, low-weight particles are rouletted, so time is not wasted following particles of trivial weight.

One cannot insist that every history contribute the same score (a zero variance solution), but by using a window inversely proportional to the importance, one can insist that the mean score from any track in the problem be roughly constant. (A weight window generator exists to estimate these importance reciprocals; see page 142.) In other words, the window is chosen so that the track weight times the mean score (for unit track weight) is approximately constant. Under these conditions, the variance is due mostly to the variation in the number of contributing tracks rather than the variation in track score.

Thus far, two things remain unspecified about the weight window: the constant of inverse proportionality and the width of the window. It has been observed empirically that an upper weight bound five times the lower weight bound works well, but the results are reasonably insensitive to this

choice anyway. The constant of inverse proportionality is chosen so that the lower weight bound in some reference cell is chosen appropriately. In most instances the constant should be chosen so that the source particles start within the window.

a. Weight Window Compared to Geometry Splitting: Although both techniques employ splitting and Russian roulette, there are some important differences:

1. the weight window is space–energy dependent, whereas geometry splitting is only space dependent;

2. the weight window discriminates on particle weight before deciding appropriate action, whereas geometry splitting is done regardless of particle weight;

3. the weight window works with absolute weight bounds, whereas geometry splitting is done on the ratio of the importance across a surface;

4. the weight window can be applied at surfaces, collision sites, or both, whereas geometry splitting is applied only at surfaces;

5. the weight window can control weight fluctuations introduced by other biasing techniques by requiring all particles in a cell to have weight $W_L < W < W_U$, whereas the geometry splitting will preserve any weight fluctuations because it is weight independent;

6. in the rare case where no other weight modification schemes are present, importances will cause all particles in a given cell to have the same weight, whereas weight windows will merely bound the weight; and

7. the weight windows can be turned off for a given cell or energy regime by specifying a zero lower bound. This is useful in long or large regions where no single importance function applies. Care should be used because when the weight window is turned off at collisions, the weight cutoff game is turned on, sometimes causing too many particles to be killed.

b. The Weight Window Generator: The weight window generator is a method that automatically generates weight window importance functions.[37] This simplifies the task of choosing importances by guessing, intuition, experience, and trial and error and also can provide insight into the Monte Carlo calculation.

Although the window generator has proved very useful, two caveats are appropriate. The generator is by no means a panacea for all importance sampling problems and certainly is not a substitute for thinking on the user's part. In fact, in most instances, the user will have to decide when the generator's results look reasonable and when they do not. After these disclaimers, one might wonder what use to make of a generator that produces both good and bad results. To use the generator effectively, it is necessary to remember that the generated parameters are only statistical

estimates and that these estimates can be subject to considerable error. Nonetheless, practical experience indicates that a user can learn to use the generator effectively to solve some very difficult transport problems.

Examples of the weight window generator are given in references 2 and 37 and should be examined before using the generator. Note that this importance estimation scheme works regardless of what other variance reduction techniques (except DXTRAN) are used in the calculation. A generator useful with DXTRAN is under development.

c. Theory: The importance of a particle at a point P in phase space equals the expected score a unit weight particle will generate. Imagine dividing the phase space into a number of phase space "cells" or regions. The importance of a cell can then be defined as the expected score generated by a unit weight particle after entering the cell. Thus, with a little bookkeeping, the cell's importance can be estimated as

$$\text{Importance (expected score)} = \frac{\text{total score because of particles (and their progeny) entering the cell}}{\text{total weight entering the cell}}$$

After the importances have been generated, MCNP assigns weight windows inversely proportional to the importances. Then MCNP supplies card images of the weight windows to be used in the input file of a subsequent calculation. The phase space division is defined on the WWG card. In other words, the WWG card defines the energy ranges used to generate the weight windows. In addition, the constant of proportionality is specified on the WWG card.

d. Limitations of the Weight-Window Generator: The principal problem encountered when using the generator is bad estimates of the importance function because of the statistical nature of the generator. In particular, unless a phase space region is sampled adequately, there will be either no generator importance estimate or an unreliable one. The generator often needs a very crude importance guess just to get any tally; that is, the generator needs an initial importance function to estimate a (we hope) better one for subsequent calculations.

Fortunately, in most problems the user can guess some crude importance function sufficient to get enough tallies for the generator to estimate a new set of weight windows. Because the weight windows are statistical, several iterations are usually required before the optimum importance function is found for a given tally; that is, the first set of generated

weight windows should be used in a subsequent calculation, which generates a better set of windows, etc.

In addition to iterating on the generated weight windows, our experience is that the user must exercise some degree of judgment. Specifically, in a typical generator calculation, some generated windows will look suspicious and will have to be reset. In MCNP, this task is simplified by an algorithm that automatically scrutinizes importance functions, either input by the user or generated by a generator. By flagging the generated windows that are more than a factor of 4 different from those in adjacent spatial regions, it is often easy for the user to determine which generated weight windows are likely to be statistical flukes that should be revised before the next generator iteration. For example, suppose the lower weight bounds in adjacent cells were 0.5, 0.3, 0.9, 0.05, 0.03, 0.02, etc.; here the user would probably want to change the 0.9 to something like 0.1 to fit the pattern and to reduce the 18:1 ratio between cells 3 and 4.

The weight window generator will also fail when phase space is not sufficiently subdivided. In this case no single set of weight window bounds is representative of the whole region. It is necessary to turn off the weight windows (by setting a lower bound of zero) or to further subdivide the geometry or energy phase space.

10. Exponential Transform

The exponential transform samples the distance to collision from a nonanalog probability density function. Although many impressive results are claimed for the exponential transform, it should be remembered that these results are usually obtained for one-dimensional geometries and quite often for energy-independent problems. A review article by Clark[38] gives theoretical background and sample results for the exponential transform. Sarkar and Prasad[39] have done a purely analytical analysis for the optimum transform parameter for an infinite slab and one energy group.

The exponential transform allows particle walks to move in a preferred direction by artifically reducing the macroscopic cross section in the preferred direction and increasing the cross section in the opposite direction according to

$$\Sigma_t^* = \Sigma_t \,(1 - p\mu),$$

where Σ_t^* = fictitious transformed cross section
Σ_t = true total cross section
p = the exponential transform parameter used to vary the degree of biasing $|p| < 1$. May be a constant or $p = \Sigma_a/\Sigma_t$, in which case $\Sigma_t^* = \Sigma_s$.

μ = cosine of the angle between the preferred direction and the particle's velocity. $|\mu| \leq 1$. The preferred direction can be specified on a VECT card.

At a collision a particle's weight is multiplied by a factor w_c (derived below) so that the expected weight colliding at any point is preserved; that is, the particle's weight is adjusted such that the weight multiplied by the probability that the next collision is in ds about s remains constant.

The probability of colliding in ds about s is

$$\Sigma e^{-\Sigma s} ds$$

(where Σ is either Σ_t or Σ_t^*) so that preserving the expected collided weight requires

$$\Sigma_t e^{-\Sigma_t s} ds = w_c \Sigma_t^* e^{-\Sigma_t^* s} ds,$$

or

$$w_c = \frac{\Sigma_t e^{-\Sigma_t s}}{\Sigma_t^* e^{-\Sigma_t^* s}} = \frac{e^{-p\Sigma_t \mu s}}{1 - p\mu}$$

If the particle reaches a cell surface, time cutoff, DXTRAN sphere, or tally segment instead of colliding, the particle's weight is adjusted so that the weight, multiplied by the probability that the particle travels a distance s to the surface, remains constant. The probability of traveling a distance s without collision is

$$e^{-\Sigma s} ,$$

so that preserving the expected uncollided weight requires

$$e^{-\Sigma_t s} = w_s e^{-\Sigma_t^* s},$$

or

$$w_s = \frac{e^{-\Sigma_t s}}{e^{-\Sigma_t^* s}} = e^{-p\Sigma_t \mu s}$$

For one-dimensional, deep penetration through highly absorbing media, the variance will typically decrease as p goes from zero to some p′,

and then increase as p goes from p′ to one. For $p < p'$, the solution is "underbiased" and for $p > p'$, the solution is "overbiased."

Choosing p′ is usually a matter of experience, although some insight may be gleaned by understanding what happens in severely underbiased and severely overbiased calculations. For illustration, apply the variance analysis of page 112 to a deep penetration problem when the exponential transform is the only nonanalog technique used. In a severely underbiased calculation ($p \to 0$), very few particles will score, but those that do will all contribute unity. Thus the variance in an underbiased system is due to a low scoring efficiency rather than a large dispersion in the weights of the penetrating particles. In a severely overbiased system ($p \to 1$) particles will score, but there will be a large dispersion in the weights of the penetrating particles with a resulting increase in variance.

<u>Comments</u>: MCNP gives a warning message if the exponential transform is used without a weight window. There are numerous examples where an exponential transform without a weight window gives unreliable means and error estimates. However, with a good weight window both the means and errors are well behaved and the transform need not be considered a "dial-an-answer" technique. The exponential transform works best on highly absorbing media and very poorly on highly scattering media. For neutron penetration of concrete or earth, experience indicates that a transform parameter $p = 0.7$ is about optimal. For photon penetration of high-Z material, even higher values such as $p = 0.9$ are justified.

Perhaps it is worthwhile explaining what happens without a weight window. For simplicity consider a slab of thickness T with constant Σ_t. Let the tally be a simple count (F1) of the weight penetrating the slab and let the exponential transform be the only nonanalog technique used. Suppose for a given penetrating history that there are k flights, m that collide and n that do not collide. The penetrating weight is thus:

$$w_p = \prod_{i=1}^{m} \frac{e^{-p\Sigma_t \mu_i s_i}}{(1 - p\mu_i)} \prod_{j=m+1}^{k} e^{-p\Sigma_t \mu_j s_j} .$$

However, note that the particle's penetration of the slab means that

$$\sum_{\ell=1}^{k} \mu_\ell s_\ell = T \quad \text{and hence}$$

$$w_p = e^{-p\Sigma_t T} \prod_{i=1}^{m} (1 - p\mu_i)^{-1}$$

Note that the only variation in w_p is because of the $(1 - p\mu)^{-1}$ factors that arise only from collisions. For a perfectly absorbing medium, every particle that penetrates scores exactly $e^{-p\Sigma_t T}$. If a particle has only a few collisions, the weight variation will be small compared to a particle that has many collisions. The weight window acts by splitting the particle whenever the weight gets too large, depriving the particle of getting a whole series of weight multiplications upon collision that are substantially greater than one.

11. Correlated Sampling

MCNP provides for correlated sampling to estimate the change in a quantity resulting from a small perturbation of any type in the problem. This technique enables the evaluation of small quantities that would otherwise be masked by the statistical errors of uncorrelated calculations. MCNP correlates a pair of runs by providing each new history in the unperturbed and perturbed problems with the same initial pseudorandom number, thus the same sequence of subsequent numbers, until the perturbation causes the sequences to diverge. This is done by incrementing the random number generator at the beginning of each history by 4297 random numbers, from the beginning (not end) of the previous history, a quantity greater than would be needed by most histories.

Unfortunately, an estimate of the error in the difference is not available unless a postprocessor code is used. There are plans to correct this deficiency and reference 2 shows how the error in the difference between two correlated runs can be estimated with a postprocessor.

12. Point Detector Tally

The point detector is a tally and does not bias random walk sampling. Recall from Section VI, however, that the tally choice affects the efficiency of a Monte Carlo calculation. Thus, a little will be said here in addition to the discussion in the tally section.

Although flux is a point quantity, flux at a point cannot be estimated by either a track-length tally (F4) or a surface flux tally (F2) because the probability of a track entering the volume or crossing the surface of a point is zero. For very small volumes, a point detector tally can provide a good estimate of the flux where it would be almost impossible to get either a track-length or surface-crossing estimate because of the low probability of crossing into the small volume.

It is interesting that a DXTRAN sphere of vanishingly small size with a surface-crossing tally across the diameter normal to the particle's trajectory is equivalent to a point detector. Thus, many of the comments

on DXTRAN are also appropriate and the DXC cards are essentially identical to the PD cards.

For a complete discussion of point detectors, see page 87.

VIII. CRITICALITY CALCULATIONS

Nuclear criticality is characterized by k_{eff}, the eigenfunction to the neutron transport equation. If a fissioning system is noncritical, k_{eff} is the quantity by which $\bar{\nu}$ (the average number of neutrons per fission) must be divided to keep the system exactly critical. In elementary reactor theory, k_{eff} is thought of as the ratio between the numbers of neutrons in successive generations, with the fission process regarded as the birth event which separates generations of neutrons.[40] For critical systems, $k_{eff} = 1$. For subcritical systems, $k_{eff} < 1$. For supercritical systems, $k_{eff} > 1$.

Calculating k_{eff} consists of estimating the mean number of fission neutrons produced in one generation per fission neutron started. A generation is the life of a neutron from birth in fission to death by escape, parasitic capture, or absorption leading to fission. Processes such as (n,2n) and (n,3n) are considered internal to a generation and do not act as termination. Because fission neutrons are killed in each generation to provide the source for the next generation, a single history can be continued from generation to generation. The effect of the delayed neutrons is included by using the total $\bar{\nu}$. The spectrum of delayed neutrons is assumed to be the same as neutrons from prompt fission.

A. Program Flow

Because the calculation of k_{eff} entails running successive generations of particles, criticality calculations have a different program flow than other MCNP problems and require a special criticality source (KCODE) incompatible with the standard source, the surface source and user-supplied sources.

1. Initiation Stage

Initially, the user supplies the following information on the KCODE card in the INP file:

a) the nominal number of source histories, N, per source cycle;

b) an initial guess of k_{eff};

c) the number of source cycles, I_c, to skip before tally accumulation;

d) the total number of cycles, I_t, in the problem;
e) an initial spatial distribution for the source; and
f) a flag for tally normalization.

The initial spatial distribution is a set of triplets of points provided by the user or loaded from a file (SRCTP) from a previous MCNP criticality calculation. Any points in geometric cells where fission cannot occur are rejected. The remaining points are duplicated (or rejected) enough times so the total number of points, M, in the source spatial distribution is approximately the nominal source size, N. The energy of each source particle is selected from a generic thermal fission distribution if it is not available from the SRCTP file.

2. Transport Stage

In each cycle, M source particles are started isotropically from the source. For the first cycle these M points come from the initiation stage. For subsequent cycles, these points are the ones written at collision sites from the previous cycle. The weight of each source particle is N/M, so all normalizations occur as if N rather than M particles started in each cycle.

Source particles are transported through the MCNP geometry by the standard random walk process, except that fission is treated as capture, either analog or implicit. Also, at each collision point the following is done:

a) the removal lifetime by absorption estimate is accumulated;
b) if fission is possible, the k_{eff} estimate is accumulated; and
c) if fission is possible, n fission sites are stored for use as source points in the next cycle, where

where $n = W\bar{\nu}(\sigma_f/\sigma_t)\ (1/k_{eff})$;
W = particle weight (before implicit capture weight reduction or analog capture);
$\bar{\nu}$ = average number of neutrons produced by fission at the incident energy of this collision, with either prompt $\bar{\nu}$ or total $\bar{\nu}$ used;
σ_f = fission cross section;
σ_t = total cross section; and
k_{eff} = estimated k_{eff} from previous cycle or initialization stage
$M = \sum n$ = number of source points to be used in next cycle.

The number of fission sites, n, stored at each collision is rounded up or down to an integer (including zero) with a probability proportional to its closeness to that integer. Note that if the initial guess of k_{eff} is too low or too high, the number of fission sites written as source points for

the next cycle will be too high or too low relative to the desired nominal number, N. This is the only effect of a bad initial guess of k_{eff}.

The fission energy is sampled separately for each source point. It is sampled from the same distributions as fissions would be sampled in the random walk for the incident neutron energy and fissionable isotope. The geometric coordinates of the fission site are also stored.

d) The collision nuclide and reaction are sampled (after steps a, b, and c) but the fission reaction is not allowed to occur because fission is treated as capture.

3. Cycle Termination

At the end of each cycle a new set of M source particles has been written from fissions in that cycle. (The number M varies from cycle to cycle.) These M particles are written to the SRCTP file, which can then be used as the initial source in a subsequent criticality calculation with a similar, though not necessarily identical, geometry. Also, certain tally quantities are manipulated, as will be described below.

The first I_c cycles in a KCODE calculation are "settle" cycles. I_c is a user-provided input parameter for the number of source cycles to be skipped before tally accumulation. After the first I_c cycles, the source spatial distribution is assumed in equilibrium, "active" cycles begin, and tallies are accumulated. Cycle after cycle is run until either a time limit is reached or the user-specified total cycles in the problem are attained.

B. Estimation of Criticality and Removal Lifetimes

The critical eigenvalue, k_{eff}, and removal lifetime are automatically estimated in every criticality calculation in addition to any user-requested tallies. These quantities are estimated for every active cycle as well as averaged over all previous active cycles. K_{eff} is estimated in three ways; the removal lifetime is estimated in two ways. These estimates are combined and correlated so the user can select the combined tally with the lowest variance, which is also estimated.

1. Collision Estimators

The collision estimate for k_{eff} for any active cycle is:

$$k^C_{eff} = \frac{1}{N} \sum_i W_i \left[\frac{\sum_k f_k \bar{\nu}_k \sigma_{F_k}}{\sum_k f_k \sigma_{T_k}} \right] .$$

where i is summed over all collisions where fission is possible;
k is summed over all isotopes of the material involved in the i^{th} collision;

σ_{T_k} = total microscopic cross section;
σ_{F_k} = microscopic fission cross section;
$\bar{\nu}_k$ = average number of prompt or total neutrons produced per fission by the collision isotope at the incident energy;
f_k = atomic fraction for nuclide k;
N = nominal source size for cycle; and
W_i = weight of particle entering collision.

Because W_i represents the number of neutrons entering the i^{th} collision,

$$W_i \left[\frac{\sum_k f_k \bar{\nu}_k \sigma_{F_k}}{\sum_k f_k \sigma_{T_k}} \right]$$

is the expected number of neutrons to be produced from all fission processes in the collision. Thus k^C_{eff} is the mean number of fission neutrons produced per neutron generation.

The collision estimate of the removal lifetime for any active cycle is

$$R_c = \frac{1}{N} \left[\sum_i W_i T_i \left[\frac{\sum_k f_k (\sigma_{A_k} + \sigma_{F_k})}{\sum_k f_k \sigma_{T_k}} \right] + \sum W_e T_e \right] \ ,$$

where σ_{A_k} is the microscopic absorption cross section excluding fission, T_i is the time from birth of the particle to the collision, and W_e and T_e are the weight and elapsed time from particle birth to escape. The removal lifetime is simply the average time required for a fission source neutron to be removed from the system by either absorption, fission, or escape.

2. Absorption Estimators

The absorption estimators for k_{eff} differs for analog capture and implicit capture. For analog capture,

$$k^A_{eff} = (1/N) \sum_i W_i \bar{\nu}_k \sigma_{F_k} / (\sigma_{A_k} + \sigma_{F_k}) \ ,$$

where i is summed over each analog capture event in the k^{th} isotope. Note that in analog capture, the weight is the same both before and after the collision. Because analog capture includes fission in criticality

calculations, the frequency of analog capture at each collision is $(\sigma_{A_k} + \sigma_{F_k})/\sigma_{T_k}$. The product of the analog absorption k_{eff} estimate and its frequency at each collision is very similar to the collision estimator of k_{eff} except that the k^{th} nuclide selected in the collision is used rather than quantities averaged over all nuclides.

For implicit capture, the following is accumulated at each collision

$$k^A_{eff} = (1/N) \sum_i W'_i \, \bar{\nu}_k \, \sigma_{F_k} \, /(\sigma_{A_k} + \sigma_{F_k}) \ ,$$

where i is summed over all collisions in which fission is possible and where $W'_i = W_i(\sigma_{A_k} + \sigma_{F_k}) / \sigma_{T_k}$ is the weight adjustment for implicit capture. The difference between the absorption estimator, k^A_{eff}, and the collision estimator, k^C_{eff}, is that again the nuclide involved in the collision is used for absorption rather than an average of all nuclides in the material for the collision estimator.

For analog capture, the removal lifetime is accumulated as $R_A = (1/N) \sum_i W_i \, T_i$ at analog capture events and system leakage. For implicit capture, the removal lifetime is accumulated as $R_A = (1/N) \sum_i W'_i \, T_i$ at all collisions and system leakage.

3. Track Length Estimator of k_{eff}

The track length estimator of k_{eff} is accumulated every time the neutron traverses a distance d:

$$k^{TL}_{eff} = (1/N) \sum_i W_i \, \rho d \sum_k f_k \bar{\nu}_k \sigma_{F_k},$$

where i is summed over all neutron trajectories, ρ is the atomic density in the cell, and d is the trajectory track length from the last event. Because $\rho d \sum_k f_k \bar{\nu}_k \sigma_{F_k}$ is the expected number of fission neutrons produced along trajectory d, k^{TL}_{eff} is a third estimate of the mean number of fission neutrons produced in a cycle per nominal (fission) source neutron.

There is no track length estimator of the removal lifetime.

4. Error Estimation and Estimator Combination

After the first I_c settle cycles (during which the source distribution is allowed to come into equilibrium), MCNP begins to combine the estimates of k_{eff} and the removal lifetimes with those estimates from previous active (after the settle) cycles. The relative error of each quantity, R, is estimated in the usual way as

$$R = \frac{1}{\bar{x}} \sqrt{\frac{\overline{x^2} - \bar{x}^2}{M - 1}}$$

where M = the number of active cycles and

$$\bar{x} = \frac{1}{M} \sum_m x_m, \qquad \overline{x^2} = \frac{1}{M} \sum_m x_m^2,$$

where x_m = a quantity, such as k_{eff}^C, from cycle m.

MCNP also combines the three estimators in all possible ways and determines the covariance and correlations. The simple average of two estimators is defined as $x^{ij} = (1/2)(x^i + x^j)$, where, for example, x^i may be the collision estimator, k_{eff}^C, and x^j may be the absorption estimator, k_{eff}^A.

The "combined average" of two estimators is weighted by the covariances as

$$x^{ij} = x^i - \frac{(x^i - x^j)(C_{ii} - C_{ij})}{(C_{ii} + C_{jj} - 2C_{ij})} \quad .$$

where the covariance, C_{ij} is

$$C_{ij} = \frac{1}{M}\sum_m x_m^i x_m^j - \left(\frac{1}{M} \sum_m x_m^i\right) \left(\frac{1}{M} \sum_m x_m^j\right).$$

Note that $C_{ii} = \overline{x^2} - \bar{x}^2$ for estimator i.

The "correlation" between two estimators is a function of their covariances and is given by

$$\text{correlation} = \frac{C_{ij}}{\sqrt{|C_{ii}C_{jj}|}} \quad .$$

The combined average and uncertainty of all three k_{eff} estimators is quite complicated. Details are given in a paper by Halperin.[41]

5. Interpreting Estimators and Their Combinations

The advantages of each estimator vary with problem specifications; no one estimator will be advantageous for all problems. All the estimators are valid. By providing them all, the user can choose the one with the lowest variance. Furthermore, a combined estimate usually has a lower variance than does a separate estimator. For example, if two estimators of

approximately equal variance have negligible correlation, their simple average will have about half the variance of either estimator alone. If the estimators are anticorrelated, the potential variance reduction may exceed a factor of two. If, however, the two estimators are significantly correlated or have radically differing variances, little or nothing will be gained by averaging them.

As an empirical rule, the absorption estimator defined above will frequently be only weakly correlated with either the collision or the track length estimator, and – at least for some fast reactor compositions – may be significantly anticorrelated. Under these conditions, the optimum estimator will be a simple average of the absorption estimator with one of the other two. Except in the most heterogeneous systems, the collision and track length estimators are likely to be so strongly correlated that little gain can be expected by averaging them. A simple average of all three estimators may be advantageous under some circumstances but is not likely to produce an additional large gain, if any. When a weight reduction rather than an analog capture treatment is used, the correlation between the absorption estimator and the others will be increased and the advantages of averaging estimators correspondingly reduced.

In addition to the simple averaging of estimators, the combined estimators of the previous section are available. The combined estimators are linear combinations of estimators computed with the properties of the sample covariance matrix following a paper by Halperin.[41] This technique is a generalization of the concept of $1/\sigma^2$ weighting when the correlation between the estimators is considered an unknown not equal to zero. The combined estimator is only rarely more useful than simple averaging and may frequently be counterproductive. However, when the correlation between estimators is small or negative but the variances are significantly unequal, the method of combined estimators may produce a modest gain. The method of combined estimators also exhibits one feature that may occasionally be disconcerting. Sometimes the combined estimate will lie <u>outside</u> the interval defined by the two individual estimates.

When constructing a confidence interval using any of the single k_{eff} or removal lifetime estimators, or any of the <u>simple</u> averages, the number of degrees of freedom is N−1, where N is the number of cycles. However, when using the <u>combined</u> average of two estimators, the number of degrees of freedom is N−2, while for a combined average of three estimators, it is N−3.

6. Normalization of Standard Tallies

Track length fluxes, surface currents, surface fluxes, heating and detectors – all the standard MCNP tallies – may be tallied during a criticality calculation. These tallied quantities are accumulated only

after the I_c settle cycles are finished. The tally normalization is per active source particle, n, where $n = N*(I_t-I_c)$, in which N is the nominal source size (from the KCODE card); I_t is the total number of cycles in the problem; and I_c is the number of settle cycles (from KCODE card). The number n is appropriately adjusted if the last cycle is only partially completed. If the tally normalization flag (on the KCODE card) is turned on, the tally normalization is the actual number of starting particles during the active cycles rather than the nominal number above. Bear in mind, however, that the source particle weights are all set to $W = N/M$ so that the source normalization is based upon the nominal source size N for each cycle.

7. Neutron Tallies and the Multiplication Factor

The MCNP multiplication factor M printed out on the Creation and Loss Ledger page differs from the k_{eff} from the criticality code. We will examine a simple model to illustrate the approximate relationship between these quantities and compare the tallies between standard and criticality calculations.

Assume we run a standard MCNP calculation using an external neutron source distribution identical in space and energy to the source distribution obtained from the solution of an eigenvalue problem with $k_{eff} < 1$. Each generation will usually have the same space and energy distribution as the source. Then the contribution to an estimate of any quantity from one generation is reduced by a factor of k_{eff} from the contribution in the preceding generation. The estimate E_k of a tally quantity obtained in an eigenvalue calculation is the contribution for one generation produced by a unit source. An estimate for a standard MCNP fixed source calculation, E_s, is the sum of contributions for all generations starting from a unit source.

$$E_s = E_k + k_{eff}E_k + k_{eff}^2E_k + k_{eff}^3E_k + \dots = E_k/(1 - k_{eff}). \qquad (2.11)$$

Note that $1/(1 - k_{eff})$ is the true system multiplication. The above result depends on our assumptions about the unit source used in the standard MCNP run. Usually, E_s will vary considerably from the above result, depending on the difference between the actual source and the eigenmode source generated in the eigenvalue problem. It will be a fairly good estimate if the actual source is a distributed source roughly approximating the eigenmode source. Tallies from a criticality calculation are appropriate only for a critical system and the tally results can be scaled to a desired fission neutron source (power) level.

In a standard MCNP problem, the net multiplication M is defined to be unity plus the gain G_f in neutrons from fission plus the gain G_x from

nonfission multiplicative reactions. Using neutron weight balance (creation equals loss),

$$M = 1 + G_f + G_x = W_e + W_c \quad , \tag{2.12}$$

where W_e is the weight of neutrons escaped per source neutron and W_c is the weight of neutrons captured per source neutron. In a criticality calculation, fission is treated as an absorptive process; the corresponding relationship for the net multiplication is then

$$M^o = 1 + G_x^o = W_e^o + W_c^o + W_f^o \quad , \tag{2.13}$$

where the superscript o designates results from the criticality calculation and W_f^o is the weight of neutrons causing fission per source neutron. Because k_{eff} is the number of fission neutrons produced in a generation per source neutron, we can also write

$$k_{eff} = \bar{\nu} W_f^o \quad , \tag{2.14}$$

where $\bar{\nu}$ is the average number of neutrons emitted per fission for the entire problem. Making the same assumptions as above for the fixed source used in the standard MCNP calculation and using equations (2.11), (2.12), and (2.13), we obtain

$$M = W_e + W_c = \frac{W_e^o + W_c^o}{1 - k_{eff}} = \frac{M^o - W_f^o}{1 - k_{eff}}$$

or, by using (2.13) and (2.14),

$$M = \frac{M^o - \dfrac{k_{eff}}{\bar{\nu}}}{1 - k_{eff}} = \frac{1 - \dfrac{k_{eff}}{\bar{\nu}} + G_x^o}{1 - k_{eff}} \quad .$$

Often, the nonfission multiplicative reactions $G_x^o \ll 1$. This implies that k_{eff} can be estimated by $(M-1)/(M-1/\bar{\nu})$ in a fixed source problem when the source distributions are nearly the same. The average value of $\bar{\nu}$ in a problem can be calculated by dividing the fission neutrons gained by the fission neutrons lost as given in the totals of the neutron weight balance for physical events. Note, however, that the above estimate is subject to the same limitations as described in Eq. 2.11.

C. *Criticality Problem Recommendations*

Criticality problems suffer from two potential problems. The first is the failure to sufficiently converge the source from its initial guess to a distribution fluctuating around the eigenmode solution. It is recommended that you make an initial run with a relatively small number of source particles per generation (perhaps 500) and generously allow a large enough number of cycles so that the eigenvalue appears to be fluctuating about a constant value. You should examine the results and continue the calculation if any trends in the eigenvalue are noticeable. The SRCTP file from the last generation of the initial run can then be used as the source for the final production run to be made with a large number of histories per generation.

The second potential problem arises from the fact that the criticality algorithm produces a bias in the estimated eigenvalue.[42] How large the bias may be is generally unknown, but has historically been assumed to be small and/or unimportant. What is known is that the bias varies as 1/N, where N is the number of source particles per generation. Thus it is desirable to make N as large as possible. If I_t is the number of cycles contributing to the eigenvalue estimation, then the total running time is proportional to NI_t, and the standard deviation in the estimated eigenvalue is proportional to $1/\sqrt{NI_t}$. By looking at the results of the convergence run, the total number of histories necessary to achieve the desired variance may be estimated. Because nothing is to be gained by making I_t greater than about 20 (and somewhat less may be satisfactory), N can be chosen to give the desired number of histories for I_t cycles.

The above described convergence procedure can be extended for very slowly convergent systems, typically large, thermal low-leakage systems, where a convergence run might be made with 250 histories per cycle. Then a second convergence run would be made with 1000 histories per cycle, using the SRCTP file from the first run as an initial guess. If the results from the second run appear satisfactory, then a final run might be made using 4000 particles per cycle with SRCTP from the second run as an initial guess. In the final run, only a new cycles need be skipped to reduce correlation from the initial 1000 histories on SRCTP.

IX. VOLUMES AND AREAS[43]

The particle flux in Monte Carlo transport problems is often estimated as the track length per unit volume or the number of particles crossing a surface per unit area. Therefore, knowing the volumes and surface areas of various geometric regions in a Monte Carlo problem is very important.

Knowing volumes is also useful in calculating the masses and densities of problem cells and thus in calculating volumetric or mass heating. Further, calculation of the mass of a geometry is frequently a good check on the accuracy of the geometry setup when the mass is known by other means.

Calculating volumes and surface areas in modern Monte Carlo transport codes is nontrivial. In particular MCNP allows for constructing cells from the unions and/or intersections of any regions defined by an arbitrary combination of second-degree surfaces, toroidal fourth-degree surfaces, or both. These surfaces may have different orientations. They may be segmented for tallying purposes, or the cells they compose may even consist of several disjoint subcells. Although such generality greatly increases the flexibility of MCNP, computing cell volumes and surface areas understandably requires increasingly elaborate computational methods.

MCNP automatically calculates volumes and areas if they have been generated by surfaces of revolution about any axis, even a skew axis. MCNP also calculates the volumes and surface areas of polyhedral cells. If a tally is segmented, the segment volumes or areas are also computed. For nonrotationally symmetric or nonpolyhedral cells, a stochastic volume and surface area method which uses ray tracing is available.

A. *Rotationally Symmetric Volumes and Areas*

The procedure for computing volumes and surface areas of rotationally symmetric bodies is as follows:

1. Determine the common axis of symmetry of the cell.[40] If there is none and if the cell is not a polyhedron, MCNP cannot compute the volume (except stochastically) and the area of each bounding surface cannot be computed on the side of the asymmetric cell.

2. Convert the bounding surfaces to q-form:

$$ar^2 + br + cs^2 + ds + e = 0,$$

where s is the axis of rotational symmetry in the r-s coordinate system. All MCNP surfaces except tori are quadratic surfaces and can therefore be put into q-form.

3. Determine all intersections of the bounding surfaces with each other in the r-s coordinate system. This generally requires the solution of a quartic equation.[20] For spheres, ellipses, and tori, extra intersection points are added so that these surfaces are not infinite. Put the list of intersections in order of increasing s-coordinate. If no intersection is found, the surface is infinite; its volume and the surface area on one side cannot be computed.

4. Integrate over each bounding surface segment between intersections:

$$V = \pi \int r^2 \, ds \qquad \text{for volumes;}$$

$$A = 2\pi \int r \sqrt{1 + \left(\frac{dr}{ds}\right)^2} \, ds \qquad \text{for surface areas.}$$

A bounding surface segment lies between two intersections that bound the cell of interest.

A numerical integration is required for the area of a torroidal surface; all other integrals are quickly solved by integration formulas. The sense of a bounding surface to a cell determines the sign of V. Note that the area of each surface is determined cell-by-cell twice, once for each side of the surface. A surface area will be calculated unless bounded on both sides by asymmetric or infinite cells.

B. Polyhedron Volumes and Areas

A polyhedron is a body bounded only by planes (the planes may have an arbitrary orientation). The procedure for calculating the volumes and surface areas of polyhedra is as follows:

1. For each facet side (planar surface), determine the intersections (r_i,s_i) of the other bounding planes in the r–s coordinate system. The r–s coordinate system is redefined for each facet to be an arbitrary coordinate system in the plane of the facet.

2. Determine the area of the facet:

$$a = 1/2 \sum (s_{i+1} - s_i)(r_{i+1} + r_i)$$

and the coordinates of its centroid, r_c, s_c:

$$r_c = 1/(6A) \sum (s_{i+1} - s_i)(r_{i+1}^2 + r_{i+1}r_i + r_i^2)$$

$$s_c = 1/(6A) \sum (r_{i+1} - r_i)(s_{i+1}^2 + s_{i+1}s_i + s_i^2).$$

The sums are over all bounding edges of the facet where i and i+1 are the ends of the bounding edge such that in going from i to i+1 the facet is on the right side. As with rotationally symmetric cells, the area of a surface is determined cell-by-cell twice, once for each side. The area of a surface on one side is the sum over all facets on that side.

3. The volume of a polyhedron is computed by using an arbitrary reference plane. Prisms are projected from each facet normal to the reference plane, and the volume of each prism is $V = da \cos \theta$ where

d = distance from reference plane to facet centroid;

a = facet area; and
θ = angle between the external normal of the facet and the positive normal of the reference plane.

The sum of the prism volumes is the polyhedron cell volume.

C. *Stochastic Volume and Area Calculation*

MCNP cannot calculate the volumes and areas of asymmetric, nonpolyhedron, or infinite cells. Also, in very rare cases, the volume and area calculation may fail because of roundoff errors. For these cases a stochastic estimation is possible by ray tracing. The procedure is as follows:

1. Void out all materials in the problem (VOID card).
2. Set all nonzero importances to one and all positive weight windows to zero.
3. Use a planar source with a source weight equal to the surface area to flood the geometry with particles. This will cause the particle flux throughout the geometry to statistically approach unity. Perhaps the best way for a stochastic volume estimation is to use an inward-directed, biased cosine source on a spherical surface with weight equal to πr^2.
4. Use the cell flux tally (F4) to tabulate volumes and the surface flux tally (F2) to tabulate areas. The cell flux tally is inversely proportional to cell volume. Thus in cells whose volumes are known, the unit flux will result in a tally of unity and in cells whose volume is uncalculated, the unit flux will result in a tally of volumes. Similarly, the surface flux tally is inversely proportional to area so that the unit flux will result in a tally of unity wherever the area is known and a tally of area wherever it is unknown.

X. *PLOTTER*

The MCNP plotter draws cross sections of the problem geometry in response to commands entered by the user. See Appendix B for the command vocabulary and examples of use. The pictures may be drawn on the screen of a terminal or on some local or remote hard copy graphics device, as directed by the user. The pictures are drawn in a square viewport on the graphics device. The mapping between the viewport and the portion of the problem space to be plotted, which is called the window, is defined by the user. A plane in problem space, the plot plane, is defined by specifying an origin $\vec{r}_o$ and two perpendicular basis vectors $\vec{a}$ and $\vec{b}$. The size of the window in the plot plane is defined by specifying two extents. The picture appears in the viewport with the origin at the center, the first basis

vector pointing to the right and the second basis vector pointing up. The width of the picture is twice the first extent and the height is twice the second extent. If the extents are unequal, the picture is distorted. The central task of the plotter is to plot curves representing the intersections of the surfaces of the geometry with the plot plane within the window.

All plotted curves are conics, defined here to include straight lines. The intersection of a plane with any MCNP surface that is not a torus is always a conic. A torus is plotted only if the plot plane contains the torus axis or is perpendicular to it, in which cases the intersection curves are conics. The first step in plotting the curves is to find equations for them, starting from the equations for the surfaces of the problem. Equations are needed in two forms for each curve: a quadratic equation and a pair of parametric equations. The quadratic equations are needed to solve for the intersections of the curves. The parametric equations are needed for defining the points on the portions of the curves that are actually plotted.

The equation of a conic is

$$As^2 + 2Hst + Bt^2 + 2Gs + 2Ft + C = 0,$$

where s and t are coordinates in the plot plane. They are related to problem coordinates (x,y,z) by

$$\vec{r} = \vec{r}_o + s\vec{a} + t\vec{b}$$

or in matrix form

$$\begin{bmatrix}1\\x\\y\\z\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\\x_o & a_x & b_x\\y_o & a_y & b_y\\z_o & a_z & b_z\end{bmatrix}\begin{bmatrix}1\\s\\t\end{bmatrix} \quad \text{or} \quad \begin{bmatrix}1\\x\\y\\z\end{bmatrix} = PL\begin{bmatrix}1\\s\\t\end{bmatrix} .$$

In matrix form the conic equation is

$$[1\ s\ t]\begin{bmatrix}C & G & F\\G & A & H\\F & H & B\end{bmatrix}\begin{bmatrix}1\\s\\t\end{bmatrix} = 0 \quad \text{or} \quad [1\ s\ t]\ QM\begin{bmatrix}1\\s\\t\end{bmatrix} .$$

Thus, finding the equation of a curve to be plotted is a matter of finding the QM matrix, given the PL matrix and the coefficients of the surface.

CHAPTER 2
Plotter

Any surface in MCNP, if it is not a torus, may be readily written as

$$Ax^2 + By^2 + Cz^2 + Dxy + Eyz + Fzx + Gx + Hy + Jz + K = 0$$

or in matrix form as

$$[1\ x\ y\ z] \begin{bmatrix} K & G/2 & H/2 & J/2 \\ G/2 & A & D/2 & F/2 \\ H/2 & D/2 & B & E/2 \\ J/2 & F/2 & E/2 & C \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \\ z \end{bmatrix} = 0$$

or $$[1\ x\ y\ z]\ AM \begin{bmatrix} 1 \\ x \\ y \\ z \end{bmatrix} = 0$$

The transpose of the transformation between (s,t) and (x,y,z) is

$$[1\ x\ y\ z] = [1\ s\ t]\ PL^T,$$

where PL^T is the transpose of the PL matrix. Substitution in the surface equation gives

$$[1\ s\ t]\ PL^T\ AM\ PL \begin{bmatrix} 1 \\ s \\ t \end{bmatrix} = 0$$

Therefore, $QM = PL^T\ AM\ PL$.

A convenient set of parametric equations for conics is

straight line	$s = C_1 + C_2p$
	$t = C_4 + C_5p$
parabola	$s = C_1 + C_2p + C_3p^2$
	$t = C_4 + C_5p + C_6p^2$
ellipse	$s = C_1 + C_2\sin p + C_3\cos p$
	$t = C_4 + C_5\sin p + C_6\cos p$
hyperbola	$s = C_1 + C_2\sinh p + C_3\cosh p$
	$t = C_4 + C_5\sinh p + C_6\cosh p.$

The type of a conic is determined by examination of the conic invariants,[44] which are simple functions of the elements of QM. Some of the surfaces produce two curves, such as the two branches of a hyperbola or two straight

lines. A separate set of parametric coefficients, C_1 through C_8, is needed for each curve in such cases. The parametric coefficients are found by transforming QM into yet another coordinate system where most of its elements are zero. The parametric coefficients are then simple functions[41] of the remaining elements. Finally, the coefficients are transformed from that coordinate system back to the (s,t) system.

For a plottable torus, the curves are either a pair of identical ellipses or a pair of concentric circles. The parametric coefficients are readily calculated from the surface coefficients and the elements of QM are simple functions of the parametric coefficients.

The next step is to reject all curves that lie entirely outside the window by finding the intersections of each curve with the straight line segments that bound the window, taking into account the possibility that an ellipse may lie entriely inside the window.

The remaining curves are plotted one at a time. The intersections of the current curve with all of the other remaining curves and with the boundaries of the window are found by solving the simultaneous equations

$$[1 \;\; s \;\; t] \; QM_i \begin{bmatrix} 1 \\ s \\ t \end{bmatrix} = 0$$

where i=1 is the current curve and i=2 is one of the other curves. This, in general, requires finding the roots of a quartic. False roots and roots outside the window are rejected and the value of the parameter p for each remaining intersection is found. The intersections are then arranged in order of increasing values of p.

Each segment of the curve – the portion of the curve between two adjacent intersections – is examined to see whether and how it should be plotted. A point near the center of the segment is transformed back to the (x,y,z) coordinate system. All cells immediately adjacent to the surface at that point are found. If there is exactly one cell on each side of the surface and those cells are the same, the segment is not plotted. If there is exactly one cell on each side and those cells are different, the segment is plotted as a solid line. If anything else is found, the segment is plotted as a dotted line, which indicates either that there is an error in the problem geometry or that some other surface of the problem also intersects the plot plane along the segment.

If a curve to be plotted is not a straight line, it is plotted as a sequence of short straight lines between selected points on the curve. The points are selected according to the criterion that the middle of the line drawn between points must not lie farther from the nearest point on the true curve than the nominal resolution of the picture. The nominal resolution is fixed at 1/3000 of a side of the viewport. This is a bit

coarse for the best plotting devices and is quite a bit too fine for the worst ones, but it produces adequate pictures at reasonable cost.

XI. PSEUDORANDOM NUMBERS

MCNP, like any other Monte Carlo program, uses a sequence of pseudorandom numbers to sample from probability distributions. The scheme that Version 3A uses to generate pseudorandom numbers is the same one that MCNP has always used, but it has been made portable. The basic method is the popular congruential scheme of Lehmer.[45] A sequence of integers i_n is generated by $i_1 = 5^{19}$; $i_n = \mathrm{mod}(5^{19} i_{i-1}, 2^{48})$. The pseudorandom number R_n is a number between zero and one that is generated by $R_n = 2^{-48} i_n$. Unless the MCNP user specifies otherwise, the first pseudorandom number used in the first history of a problem is R_{4299}. The history then uses as many more numbers from the sequence as it needs. The first number used in each succeeding history is 4297 steps farther along in the sequence than the first number used in the previous history, regardless of how many numbers were used in the previous history.

The actual computation is done in function subprogram RANG with pairs of 24-bit integer values contained in real (double precision on 32-bit machines) variables. The multiply is done by an explicit double precision method. This process provides a portable, reasonably fast, pseudorandom number generator that gives identical results on all systems on which MCNP currently runs.

CHAPTER 3
DESCRIPTION OF MCNP INPUT

Input to MCNP consists of several files, but the main one supplied by the user is the INP (the default name) file, which contains the input information necessary to describe the problem. Only a small subset of all available input cards will be needed in any particular problem. The input cards are summarized by card type on page 267.

In this chapter maximum dimensions are given for some MCNP input items; these are summarized on page 270. The user can increase any of these maximum values by altering the code and recompiling. Instructions are found in Appendix A.

All features of MCNP should be used with caution and knowledge. This is especially true of detectors and variance reduction schemes; you are encouraged to read the appropriate sections of Chapter 2 before using them.

The units used throughout MCNP are given in Chapter 1 on page 19.

I. INP FILE

The INP file can have two forms, initiate-run and continue-run. Either can contain an optional message block that replaces or supplements the MCNP execution line information discussed in Appendix A.

A. Message Block

A user may optionally put a set of cards called the message block before the problem identification title card in the INP file. In computer environments where there are no execution line messages, the message block is the only means for giving MCNP an execution message. Less crucially, it is a convenient way to avoid retyping an often-repeated message. The message block starts with a card which must have the string MESSAGE: in columns 1-8. The message block ends with a blank line delimiter before the title card. There may optionally be one or more continuation cards between the first card and the blank card. Everything in columns 9-80 on the first card and in columns 1-80 on continuation cards is interpreted as a message by MCNP. The syntax of this message is the same as for the regular execution line message discussed on page 30. The meanings of the components of the message are the same as for the execution line message, subject to the following rules, which are needed to resolve conflicts between the message block and execution line message if the two are used simultaneously. Information in the execution line takes precedence over the same information in the message block, in particular:

a. INP = *filename* is not a legitimate entry in the message block. The name INP may be changed on the MCNP execution line only.
b. In the case of A = B (filename substitution), if the construct A = appears in both the execution line message and in the message block, the message block entry is ignored.
c. If there is any overlay execution specification (like IP or IX) in the execution line message, all overlay execution specifications in the message block are ignored.
d. Any keyword entry in the execution line message causes any corresponding entry in the message block to be ignored. For example, C 7 (specifying continue with the seventh dump) in the execution line message would cause C 4 in the message block to be ignored.
e. However if C or C m appears in the message block but not in the execution line, the run is still a continue-run.

B. Initiate-Run

This form is used to set up a Monte Carlo problem (describe geometry, materials, tallies, etc.) and run if desired. The initiate-run file has the following form:

```
Message Block          }  Optional
Blank Line Delimiter   }
Title Card
Cell Cards
     .
     .
Blank Line Delimiter
Surface Cards
     .
     .
Blank Line Delimiter
Data Cards
     .
     .
Blank Line Terminator          Recommended
Anything else                  Optional
```

The first card in the file after the optional message block is the required problem title card. It is limited to one line from columns 1-80 and is used as a title in various places in the MCNP output.

With a valid set of data cards MCNP will run with or without the blank line terminator. With the terminator MCNP will stop reading the input file

there even if additional lines are in the file. Some users like to keep additional material, such as alternative versions of the problem or textual information, associated with the input file itself. The terminator will prevent such additional lines from being read.

C. *Continue-Run*

Continue-run is used to continue running histories in a problem that was terminated earlier - for example, to run the job 2 hours on one night and then to run it an additional hour some later night. It can also be used to reconstruct the output of a previous run. A continue-run must contain C or CN in the MCNP execution line or message block to indicate a continue-run.

In general, in addition to the C or CN option on the MCNP execution line (see Appendix A) two files are important for this procedure: (1) the run file (default name RUNTPE), and (2) an optional continue-run input file (default name INP).

The run file, generated by MCNP in the initiate-run sequence, contains the geometry, cross sections, problem parameters, tallies, and all other information necessary to restart the job. In addition the problem results at various stages of the run are recorded in a series of dumps. See the PRDMP card (page 261) for a discussion of the selection of dump times. As discussed below, the run may be restarted from any of these dumps.

The CN execution message option differs from the C option only in that the dumps produced during the continue-run are written immediately after the fixed data portion of the RUNTPE file rather than after the dump from which the continue-run started. The new dumps overwrite the old dumps, providing a way for the user to prevent unmanageable growth of RUNTPE files.

The optional continue-run input file must have the word CONTINUE starting in column one of the first line or after the optional Message Block and its blank line delimiter if used. This file has the following form:

Message Block Blank Line Delimiter	} Optional
CONTINUE	
Data Cards	
.	
.	
Blank Line Terminator	Recommended
Anything else	Optional

The data cards allowed in the continue-run input file are a subset of the data cards available for an initiate-run file. The allowed continue-run data cards are FQ, DD, NPS, CTME, IDUM, RDUM, PRDMP, LOST, DBCN, PRINT, KCODE, ZA, ZB, and ZC.

A very convenient feature is that if none of the above items is to be changed (and if the computing environment allows execution line messages), the continue-run input file is not required; only the run file RUNTPE and the C option on the MCNP execution line are necessary. For example, if you run a job for a minute or so and it quits on time limit but you want a few more particles run, simply execute with the C or CN message on the execute line, and the job will pick up where it stopped and continue until another time limit or particle cutoff is reached or until you stop it manually. This example assumes that a run file called RUNTPE from the initial run is in your local files.

The complete continue-run execution line option is C m or CN m, where m specifies which dump to pick up from the RUNTPE and to continue with. If m is not specified, the last dump is taken by default. If the initial run producing the RUNTPE was stopped because of particle cutoff (NPS card, page 258), NPS must be increased for a continue-run. The NPS card refers to total histories to be run, including preceding continue-runs and the initial run.

In a continue-run, entering a negative number on the NPS card will produce a print output file at the time of the requested dump. No more histories will be run. This can be useful when the printed output has been lost or you want to alter the content of the output with the PRINT card.

Be cautious if you are using a FILES card in your initial run. See page 264.

D. Card Format

All input lines (we will refer to "cards" in this manual) of INP are limited to columns 1-80 and consist of card images. Most input is entered in horizontal form; however, a vertical input format is allowed for data cards. Columns 1-72 are used for data, and columns 73-80 may be used for comments. In addition, if the symbol **$** (dollar sign) appears in a line, data entry on that line is terminated and anything following the **$** is interpreted as a comment. A future version of the code will drop the distinction between columns 1-72 and columns 73-80 and will rely entirely on the dollar sign to signal a comment field within a data line. The title card, limited to one line, can contain any information the user desires or may even be blank. It usually contains information describing the particular problem. Note that a blank card elsewhere is used as a delimiter or as a terminator.

Comment cards may be used anywhere in the INP file after the problem title card and before the last blank terminator card. These cards must have the letter C in the first column of the card followed by four blanks; columns 6–80 are available for any comments. Unlike the title card, these cards are printed only with the input file listing and not anywhere else in the MCNP output file. The FCn input card is available for user comments to be printed as a heading for tally n (as a tally title, for example). The SCn card is available for user comments to be printed as a heading for source probability distribution n.

1. Horizontal Input Format

Cell cards, surface cards, and data cards all have the same format. Columns 1–5 are reserved for the name (or number) associated with the card. The name (or number) can appear anywhere in columns 1–5. A card name that has a particle designator may require more than five columns. It must start early enough that the colon is no later than in column six. Blanks in the first five columns indicate a continuation of the data from the last named card. Completely blank cards are reserved as delimiters between major sections of the input file. Columns 6–72 are for free-field format data entry associated with the card name. Separation between data entries is by one or more blank columns. An individual entry must be entirely on one line, and there can be only one card of any given type for a given particle designation (see page 172). Continuation lines are considered part of the same card. Integers must be entered where integer input is required. Other numerical data may be entered as integer or floating point and will be read properly by MCNP. (In fact noninteger numerical data may be entered in any form acceptable to a FORTRAN E-edit descriptor.)

Four features incorporated in the code facilitate input card preparation:

1. nR means *repeat* the immediately preceding entry on the card n times. For example, 2 4R is the same as 2 2 2 2 2.

2. nI means *insert* n linear interpolates between the entries immediately preceding and following this feature. For example, 1.5 2I 3.0 on a card is the same as 1.5 2.0 2.5 3. In the construct X nI Y, if X and Y are integers, and if Y − X is an exact multiple of n+1, correct integer interpolates will be created. Otherwise only real interpolates will be created, but Y will be stored directly in all cases. In the above example, the 2.0 may not be exact, but in the example 1 4I 6 = 1 2 3 4 5 6, all interpolates are exact.

3. xM is a *multiply* feature and when used on an input card, it is replaced by the value of the previous entry on the card multiplied by the factor x. For example, 1 1 2M 2M 2M 2M 4M 2M 2M is equivalent to 1 1 2 4 8 16 64 128 256.

4. nJ may be used on an input card to *jump* over the entry where used and take the default value. As an example, the following two cards are identical in their effect:

   DD .1 1000
   DD J 1000

   J J J is also equivalent to 3J. The feature simply allows you to jump to a particular entry on a card without having to explicitly specify prior items on the card. This is very convenient if you know you want to use a default value but can't remember it. DBCN 7J 5082 is another good example.

These features apply to both integer and floating point quantities. If n (an integer) is omitted in the constructs nR, nI, and nJ, then n is assumed to be 1. If x (integer or floating point) is omitted in xM, it is a fatal error. The rules for dealing with adjacent special input items are as follows:

(1) nR must be preceded by a number or by an item created by R or M.

(2) nI must be preceded by a number or by an item created by R or M, and must be followed by a number.

(3) xM must be preceded by a number or by an item created by R or M.

(4) nJ may be preceded by anything except I and may begin the card input list.

```
Examples:  1 3M 2R       =  1 3 3 3
           1 3M I 4      =  1 3 3.5 4
           1 3M 3M       =  1 3 9
           1 2R 2I 2.5   =  1 1 1 1.5 2.0 2.5
           1 R 2M        =  1 1 2
           1 R R         =  1 1 1
           1 2I 4 3M     =  1 2 3 4 12
           1 2I 4 2I 10  =  1 2 3 4 6 8 10
           3J 4R         is illegal.
           1 4I 3M       is illegal.
           1 4I J        is illegal.
```

2. Vertical Input Format

Column input is particularly useful for cell parameters and source distributions. Cell importances or volumes strung out on horizontal input

lines are not very readable and often cause errors when users add or delete cells. In column format, all the cell parameters for one cell can be on a single line, labeled with the name of the cell. If a cell is deleted, the user deletes just one line of cell parameters instead of hunting for the data item that belongs to the cell in each of several multiline cell parameter cards. For source distributions, corresponding SI, SP, and SB values are side by side on each line.

In column format, card names are put side by side on one input line and the data values are listed in columns under the card names. A # is put somewhere in character positions 1-5 on the line with the card names. The card names must be all cell parameters, all surface parameters, or all something else. If a card name appears on a # card, there must not be a regular horizontal card by that name in the same input file. If the card names are cell parameters, the names of the cells may optionally be listed in character positions 1-5 in a column under the #. If any cell names are entered, all must be entered. If cell names are entered, the cells don't have to be in the same order as they are in the cell cards block. If cell names are omitted, the default order is the order of the cells in the cell card block. The same rules apply to surface parameters, but because we presently have only one surface parameter (AREA), column input of surface parameters is less useful.

There can be more than one block of column data in an input file. Typically, there would be one block for cell parameters and one for each source distribution. If a lot of cell parameter options are being used, additional blocks of column data would be needed.

The entries in each column do not need to be precisely under the card name at the top of the column, but you might want the columns to be reasonably neat for readability. The column format is intended for input data that naturally fit into columns of equal length, but less tidy data are not prohibited. If a longer column is to the right of a shorter column, the shorter column must be filled with enough J entries to eliminate any ambiguity about which columns the data items are in.

Special syntax items (R, M, I, and J) are not as appropriate in column format as they are on horizontal lines, but they are not prohibited. They are, of course, interpreted vertically instead of horizontally. Multiple special syntax items, such as 9R, are not allowed if there are names in character positions 1-5.

The form of a column input block is

#	S_1	S_2	...	S_m
K_1	D_{11}	D_{12}	...	D_{1m}
K_2	D_{21}	D_{22}	...	D_{2m}
.	.	.	.	.
.	.	.	.	.
K_n	D_{n1}	D_{n2}	...	D_{nm}

1. The # and the K_j are in columns 1–5, and the S_i and D_{ji} are in columns 6–72.
2. m must be small enough that S_1 through S_m and every D_{ji} through D_{jm} can fit in columns 6–72.
3. Each column, S_i through $D_{\ell i}$, where ℓ may be less than n, represents a regular input card, except as modified by the K_j.
4. The S_i must be valid MCNP card names. They must be all cell parameters, all surface parameters, or all something else.
5. D_{1i} through D_{ni} must be valid entries for an S_i card, except that $D_{\ell+1,i}$ through D_{ni} may be some J's possibly followed by some blanks.
6. If D_{ji} is nonblank, $D_{j,i-1}$ must also be nonblank. A J may be used if necessary to make $D_{j,i-1}$ nonblank.
7. The S_i must not appear anywhere else in the input file.
8. The K_j are optional. They are integers and if any are nonblank, all must be nonblank.
9. If the S_i are cell parameter card names, the K_j, if present, must be valid cell names. The same is true with surface parameters. Otherwise, the K_j are ignored and should not be present.
10. If the K_j are not blank, the D_{ji} must not be multiple special syntax items, such as 9R.

E. Particle Designators

Several of the input cards require a particle designator to distinguish between input data for neutrons and data for photons. These cards are IMP, EXT, FCL, WWN, WWE, WWP, WWGE, DXT, DXC, F, F5X, F5Y, F5Z, PHYS, ESPLT, and CUT. The particle designator consists of the symbol : (colon) and the letter N or P immediately after the name of the card. For example, to enter neutron importances, use an IMP:N card; enter photon importances on an IMP:P card. With a tally card, the particle designator follows the card name including tally number. For example, *F5:N indicates a neutron point detector energy tally. In the heating tally case, both particle designators may appear. The syntax F6:N,P indicates the combined heating tally for both neutrons and photons.

F. *Default Values*

Many MCNP input parameters have default values, which are summarized on page 267. Therefore you do not always have to explicitly specify every input parameter every time if the defaults match your needs. If an input card is left out, the default values for all parameters on the card are used (the photon cutoff card with its five entries is an example). However, if you want to change a particular defaulted parameter on a card but that parameter is preceded by others, you have to specify the others or use the nJ jump feature to jump over the parameters for which you still want the default. CUT:P 3J -.10 is a convenient way to use the defaults for the first three parameters on the photon cutoff card but change the fourth.

G. *Input Error Messages*

MCNP makes extensive checks (over 300) of the input file for user errors. A fatal error message is printed, both at the terminal and in the OUTP file, if the user violates a basic constraint of the input specification, and MCNP will terminate before running any particles. The first fatal error is real; subsequent error messages may or may not be real because of the nature of the first fatal message.

The FATAL option (see Appendix A) on the MCNP execution line instructs MCNP to ignore fatal errors and run particles, but the user should be extremely cautious about doing this. If only the first three overlays IPX or a subset of them are to be run, the FATAL option is not required since MCNP will not terminate for fatal errors until the beginning of the MCRUN overlay.

Most MCNP messages are warnings and are not fatal. The user should not ignore these messages but should understand their significance before making important calculations.

In addition to FATAL and WARNING messages, MCNP issues BAD TROUBLE messages immediately before any impending catastrophe, such as a divide by zero, which would otherwise cause the program to "crash." MCNP terminates as soon as the BAD TROUBLE message is issued. FATAL errors in the processing of the INP file are the most common reason for issuing a BAD TROUBLE message. Members of X-6 can explain the significance of particular messages to you.

H. *Geometry Errors*

There is one very important kind of input error that MCNP will not detect while processing data from the INP file. MCNP cannot detect overlapping cells or gaps between cells until a particle track actually

gets lost. Even then the precise nature of the error may remain unclear. However, there is much that you can and should do to check your geometry before starting a long computer run.

Use the geometry-plotting feature of MCNP to look at the system from several directions and at various scales. Be sure that what you see is what you intend. Any gaps or overlaps in the geometry will probably show up as dashed lines. The intersection of a surface with the plot plane is drawn as a dashed line if there is not exactly one cell on each side of the surface at each point. (Dashed lines may also appear if the plot plane happens to coincide with a plane of the problem or if there are any cookie-cutter cells in the source.)

Set up and run a short problem in which your system is flooded with particle tracks from an external source. The necessary changes in the INP file are as follows:

1. Add a VOID card. This will override some of the other specifications in the problem to make all the cells voids, turn heating tallies into flux tallies, and turn off any FM cards.

2. Add another cell and a large spherical surface to the problem such that the surface surrounds the system and the old outside world cell is split by the new surface into two cells: the space between the system and the new surface, which is the new cell, and the space outside the new surface, which is now the outside world cell. Be sure that the new cell has nonzero importance. Actually, it is best to make all nonzero importances equal. If the system is infinite in one or two dimensions, use one or more planes instead of a sphere.

3. Replace the source specifications by

```
SDEF  SUR=m  NRM=-1
```

where m is the number of the new spherical surface added in Step 2. If the new surface is a plane, you must specify the portion to be used by means of POS and RAD or possibly X, Y, and Z source distributions.
Because there are no collisions, a short run will generate a great many tracks through your system. If there are any geometry errors, they should cause some of the particles to get lost.

When a particle first gets lost, whether in a special run with the VOID card or in a regular production run, the history is rerun to produce some special output on the OUTP file. Event-log printing is turned on during the rerun. The event log will show all surface crossings and will tell you the path the particle took to the bad spot in the geometry. When the particle again gets lost, a description of the situation at that point is printed. You can usually deduce the cause of the lost particle from this output.

If the cause of the lost particle is still obscure, try plotting the geometry with the origin of the plot at the point where the particle got lost and with the horizontal axis of the plot plane along the direction the

particle was moving. The cause of the trouble is likely to appear as a dashed line somewhere in the plot or as some discrepancy between the plot and your idea of what it should look like.

II. CELL CARDS

Form: j m d geom params

j	= cell number; $1 \le j \le 99999$
m	= 0 if the cell is a void.
	= material number if the cell is not a void. This indicates
	= that the cell is to contain material m, which is specified on the Mm card. See page 247.
d	= absent if the cell is a void.
	= cell material density. A positive entry is interpreted as the atomic density in units of 10^{24} atoms/cm^3. A negative entry is interpreted as the mass density in units of g/cm^3.
geom	= specification of the geometry of the cell. It consists of signed surface numbers and Boolean operators that specify how the regions bounded by the surfaces are to be combined.
params	= optional specification of cell parameters by entries in the keyword=value form.

In the geometry specification, a signed surface number stands for the region on the side of the surface where points have the indicated sense. The plus sign for positive sense is optional. The regions are combined by Boolean operators: intersection (no symbol -- implicit, like multiplication in algebra); union, :; and complement, #. Parentheses may be used to control the order of the operations. Parentheses and operator symbols also function as delimiters. Where they are present, blank delimiters are not necessary. The default order of operations is complement first, intersection second, and union last. A number immediately after a complement operator, without parentheses, is interpreted as a cell number and stands for the geometry specification of that cell (in parentheses). For a simple cell (no union or complement operators), the geometry specification is just a blank-delimited list of the bounding surfaces and ambiguity surfaces of the cell with signs determined by the sense of the cell with respect to each surface. See the Geometry sections of Chapters 1, 2, and 4 for complete explanations of how to specify the geometry of cells in MCNP.

Cell parameters may be defined on cell cards instead of in the data cards section of the INP file. The form is keyword=value, where the

allowed keywords are IMP, VOL, PWT, EXT, FCL, WWN, DXC, NONU, and TMP, with particle designators where necessary. (This feature is not yet available for the cell parameter PD.) The equals sign is optional; a blank can be used instead. If a cell parameter is specified on any cell cards, it must be specified only on cell cards and not at all in the data cards section. However, it is all right to specify some cell parameters on cell cards and the others in the data cards section. A special syntax is available for TMP data. After a single TMP keyword you can put all of the temperatures of the cell in an order corresponding to the times on the THTME card.

Example: 10 16 −4.2 1 −2 3 IMP:N=4 IMP:P=8 EXT:N=−.4X

This says that cell 10 is to be filled with material 16 at a density of 4.2 g/cm^3. The cell consists of the intersections of the regions on the positive side of surface 1, the negative side of surface 2, and the positive side of surface 3. The neutron importance in cell 10 is 4 and the photon importance is 8. Neutrons in cell 10 are subject to an exponential transform in the minus X direction with stretching parameter 0.4.

Here are some precautions for when you are preparing cell cards:

1. Avoid excessively complicated cells. MCNP runs faster when the problem geometry is made up of many simpler cells rather than fewer more complicated cells. The maximum number of items (surface numbers, operators, and parentheses) in a cell description, after all complement operators have been expanded, is 100. Most cell descriptions should be much shorter than that.

2. Avoid adding unneeded surfaces to the geometry description of a cell through poor use of the complement operator. The extra surfaces make the problem run slower and may destroy the necessary conditions for volume and area calculations. See page 286.

3. Always use the geometry-plotting feature of MCNP to check the geometry of a problem. See Appendix B.

4. Flood the system with particles from an outside source to find errors in the geometry. See page 173.

5. If you add or remove cells, change all your cell parameter cards accordingly. The difficulty of this can be reduced if you use vertical format for your cell parameter cards. See page 170. Alternatively, define the values of cell parameters on cell cards and eliminate cell parameter cards entirely.

III. SURFACE CARDS

A. Surfaces Defined by Equations

Form: j n a list

j	=	surface number: $1 \le j \le 99999$, with asterisk
	=	if it is a reflecting surface.
n	=	absent for no coordinate transformation, or
	=	number of a TRn card.
a	=	equation mnemonic from Table 3.1
list	=	one to ten entries, as required.

The surface types, equations, their mnemonics, and the order of the card entries are given in Table 3.1. To specify a surface by this method, find the surface in Table 3.1 and determine the coefficients for the equation (you may need to consult a book on analytical geometry). The information is entered on a surface card according to the above form. Under certain conditions a surface can also be defined by specifying geometrical points, as discussed in the next two sections.

A point (x,y,z) is defined as having positive sense with respect to a surface when the expression for that surface evaluated at (x,y,z) is positive. The expression for a surface is the left side of the equation for the surface in Table 3.1. With the sphere, cylinder, cone, and torus, this definition is identical to defining the sense to be positive outside the figure. With planes normal to axes (PX, PY, or PZ), the definition gives positive sense for points with x, y, or z values exceeding the intercept of the plane. For the P, SQ and GQ surfaces, the user supplies all of the coefficients for the expression and thus can determine the sense of the surface at will. This is different from the other cases where the sense, though arbitrary, is uniquely determined by the form of the expression. Therefore, in a surface transformation (see the TRn card on page 197) a PX, PY, or PZ surface may have to be replaced by a P surface just to prevent the sense of the surface from getting reversed.

If columns 1–5 contain an asterisk as well as the surface number, a reflecting surface is defined. A particle track that hits a reflecting surface is reflected specularly. Detectors and DXTRAN should not usually be used in problems that have reflecting surfaces. See page 96. Tallies in problems with reflecting surfaces will need to be normalized differently. See page 44. The surface number of a reflecting surface can have no more than 4 digits.

Table 3.1 MCNP Surface Cards

Mnemonic	Type	Description	Equation	Card Entries
P	Plane	General	$Ax+By+Cz-D=0$	A B C D
PX	↓	Normal to X-axis	$x-D=0$	D
PY	↓	Normal to Y-axis	$y-D=0$	D
PZ	↓	Normal to Z-axis	$z-D=0$	D
SO	Sphere	Centered at Origin	$x^2+y^2+z^2-R^2=0$	R
S	↓	General	$(x-\bar{x})^2+(y-\bar{y})^2+(z-\bar{z})^2-R^2=0$	$\bar{x}$ $\bar{y}$ $\bar{z}$ R
SX	↓	Centered on X-axis	$(x-\bar{x})^2+y^2+z^2-R^2=0$	$\bar{x}$ R
SY	↓	Centered on Y-axis	$x^2+(y-\bar{y})^2+z^2-R^2=0$	$\bar{y}$ R
SZ	↓	Centered on Z-axis	$x^2+y^2+(z-\bar{z})^2-R^2=0$	$\bar{z}$ R
C/X	Cylinder	Parallel to X-axis	$(y-\bar{y})^2+(z-\bar{z})^2-R^2=0$	$\bar{y}$ $\bar{z}$ R
C/Y	↓	Parallel to Y-axis	$(x-\bar{x})^2+(z-\bar{z})^2-R^2=0$	$\bar{x}$ $\bar{z}$ R
C/Z	↓	Parallel to Z-axis	$(x-\bar{x})^2+(y-\bar{y})^2-R^2=0$	$\bar{x}$ $\bar{y}$ R
CX	↓	On X-axis	$y^2+z^2-R^2=0$	R
CY	↓	On Y-axis	$x^2+z^2-R^2=0$	R
CZ	↓	On Z-axis	$x^2+y^2-R^2=0$	R
K/X	Cone	Parallel to X-axis	$\sqrt{(y-\bar{y})^2+(z-\bar{z})^2}-t(x-\bar{x})=0$	$\bar{x}$ $\bar{y}$ $\bar{z}$ t^2 ±1
K/Y	↓	Parallel to Y-axis	$\sqrt{(x-\bar{x})^2+(z-\bar{z})^2}-t(y-\bar{y})=0$	$\bar{x}$ $\bar{y}$ $\bar{z}$ t^2 ±1
K/Z	↓	Parallel to Z-axis	$\sqrt{(x-\bar{x})^2+(y-\bar{y})^2}-t(z-\bar{z})=0$	$\bar{x}$ $\bar{y}$ $\bar{z}$ t^2 ±1
KX	↓	On X-axis	$\sqrt{y^2+z^2}-t(x-\bar{x})=0$	$\bar{x}$ t^2 ±1
KY	↓	On Y-axis	$\sqrt{x^2+z^2}-t(y-\bar{y})=0$	$\bar{y}$ t^2 ±1
KZ	↓	On Z-axis	$\sqrt{x^2+y^2}-t(z-\bar{z})=0$	$\bar{z}$ t^2 ±1 ±1 used only for 1 sheet cone
SQ	Ellipsoid Hyperboloid Paraboloid	Axes parallel to X-, Y-, or Z-axis	$A(x-\bar{x})^2+B(y-\bar{y})^2+C(z-\bar{z})^2$ $+2D(x-\bar{x})+2E(y-\bar{y})$ $+2F(z-\bar{z})+G=0$	A B C D E F G $\bar{x}$ $\bar{y}$ $\bar{z}$
GQ	Cylinder Cone Ellipsoid Hyperboloid Paraboloid	Axes not parallel to X-, Y-, or Z-axis	$Ax^2+By^2+Cz^2+Dxy+Eyz$ $+Fzx+Gx+Hy+Jz+K=0$	A B C D E F G H J K
TX	Elliptical or circular torus. Axis is parallel to X,Y or Z-axis		$(x-\bar{x})^2/B^2+(\sqrt{(y-\bar{y})^2+(z-\bar{z})^2}-A)^2/C^2-1=0$	$\bar{x}$ $\bar{y}$ $\bar{z}$ A B C
TY			$(y-\bar{y})^2/B^2+(\sqrt{(x-\bar{x})^2+(z-\bar{z})^2}-A)^2/C^2-1=0$	$\bar{x}$ $\bar{y}$ $\bar{z}$ A B C
TZ			$(Z-\bar{z})^2/B^2+(\sqrt{(x-\bar{x})^2+(y-\bar{y})^2}-A)^2/C^2-1=0$	$\bar{x}$ $\bar{y}$ $\bar{z}$ A B C
X Y Z P		Surfaces defined by points		See pages 180 and 183

Example 1: j PY 3

This describes a plane normal to the y-axis at y=3 with positive sense for all points with y > 3.

Example 2: j K/Y 0 0 2 .25 1

This specifies a cone whose vertex is at (x,y,z)=(0,0,2) and whose axis is parallel to the y-axis. The tangent t of the opening angle of the cone is 0.5 (note that t^2 is entered) and only the positive (right hand) sheet of the cone is used. Points outside the cone have a positive sense.

Example 3: j GQ 1 .25 .75 0 -.866
0 -12 -2 3.464 39

This is a cylinder of radius 1 cm whose axis is in a plane normal to the x-axis at x=6, displaced 2 cm from the x-axis and rotated 30° about the x-axis off the y-axis toward the z-axis. The sense is positive for points outside the cylinder. Such a cylinder would be much easier to specify by first defining it in an auxiliary coordinate system where it is symmetric about a coordinate axis and then using the TRn input card (see page 197) to define the relation between the basic and auxiliary coordinate systems. The input would then be

j 7 CX 1
*TR7 6 1 -1.732 0 30 60

See Chapter 4 for additional examples of the TRn card.

The TX, TY, and TZ input cards represent elliptical tori (fourth degree surfaces) rotationally symmetric about axes parallel to the x, y, and z axes, respectively. A TY torus is illustrated in Figure 3.1a. Note that the input parameters $\bar{x}$ $\bar{y}$ $\bar{z}$ a b c specify the ellipse

$$\frac{s^2}{b^2} + \frac{(r - a)^2}{c^2} = 1$$

rotated about the s-axis in the (r,s) cylindrical coordinate system (Figure 3.1b) whose origin is at $\bar{x}$, $\bar{y}$, $\bar{z}$ in the x, y, z system. In the case of a TY torus,

$$s = (y - \bar{y})$$

and
$$r = \sqrt{(x - \bar{x})^2 + (z - \bar{z})^2}$$

A torus is degenerate if $|a| < c$ where $0 < a < c$ produces the outer surface (Figure 3.1c), and $-c < a < 0$ produces the inner surface (Figure 3.1d).

Coordinate transformations for tori are limited to those in which each axis of the auxiliary coordinate system is parallel to an axis of the main

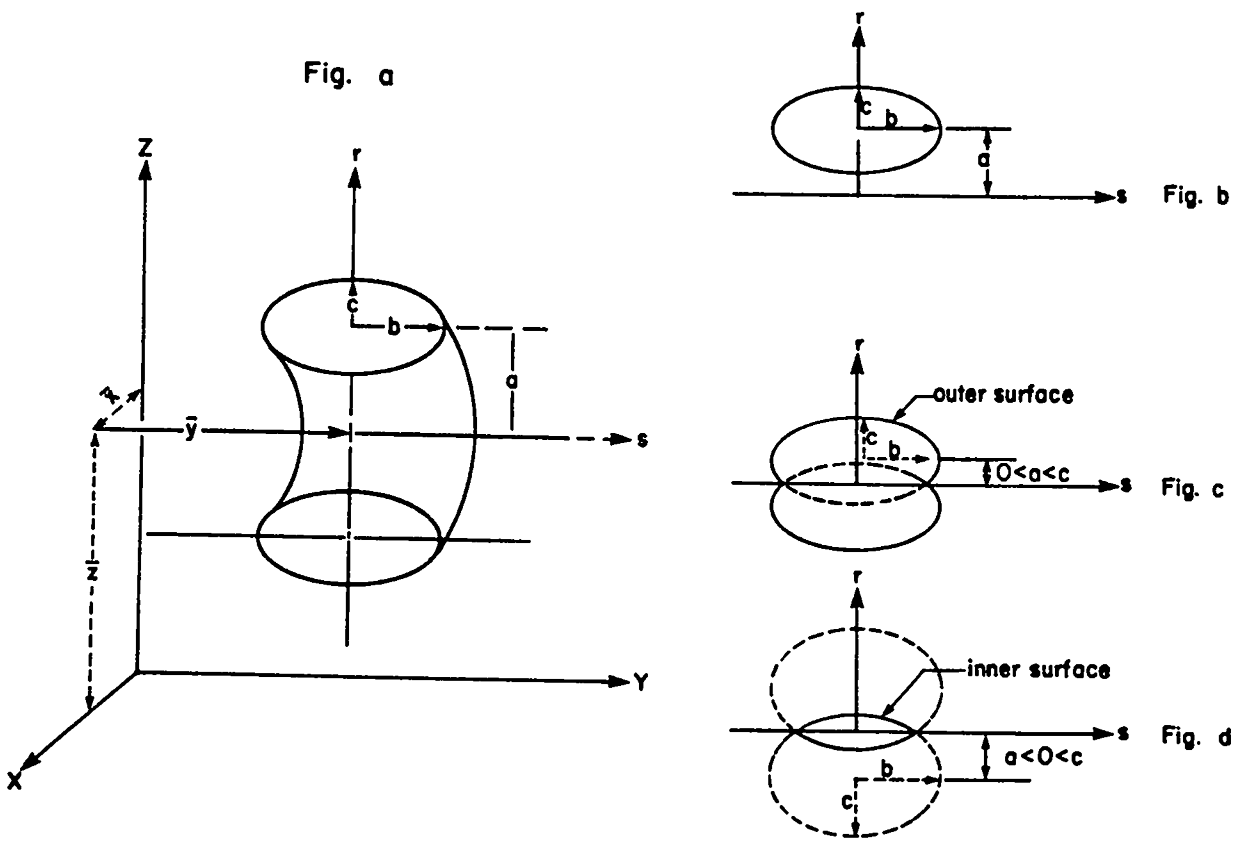


Figure 3.1 Torus

system.

B. Axisymmetric Surfaces Defined by Points

Form: j n a list

- j = surface number: $1 \le j \le 99999$
- n = absent for no coordinate transformation, or = number of TRn card.
- a = the letter X, Y, or Z
- list = one to three coordinate pairs.

Surface cards of type X, Y, and Z may be used to describe surfaces by coordinate points rather than by equation coefficients as in the previous section. The surfaces described by these cards must be symmetric about the x-, y-, or z-axis, respectively, and, if the surface consists of more than one sheet, the specified coordinate points must all be on the same sheet.

Each of the coordinate pairs defines a geometrical point on the surface. On the Y card, for example, the entries may be

j   Y y_1 r_1 y_2 r_2

where $r_i = \mathrm{SQRT}(x_i^2 + z_i^2)$ and y_i is the coordinate of point i.

If one coordinate pair is used, a plane (PX, PY, or PZ) is defined.

If two coordinate pairs are used, a linear surface (PX, PY, PZ, CX, CY, CZ, KX, KY, or KZ) is defined.

If three coordinate pairs are used, a quadratic surface (PX, PY, PZ, SO, SX, SY, SZ, CX, CY, CZ, KX, KY, KZ, or SQ) is defined.

When a cone is specified by two points, a cone of only one sheet is generated.

The senses of these surfaces (except SQ) are determined by the code to be identical to the senses one would obtain by specifying the surface by equations. For SQ, the sense is defined so that points sufficiently far from the axis of symmetry have positive sense. Note that this is different from the equation-defined SQ, where the user could choose the sense freely.

Example 1:   j   X   7 5 3 2 4 3

This describes a surface symmetric about the x-axis, which passes through the three (x,r) points (7,5), (3,2), and (4,3). This surface is a hyperboloid of two sheets, converted in MCNP to its equivalent

j   SQ   -.083333333 1 1 0 0 0 68.52083 -26.5 0 0.

Example 2:   j   Y   1 2 1 3 3 4

This describes two parallel planes at Y=1 and Y=3 and is a fatal error because the requirement that all points be on the same sheet is not met.

Example 3:   j   Y   3 0 4 1 5 0

This describes a sphere of radius 1 with center at (x,y,z) = (0,4,0).

Example 4:   j   Z   1 0 2 1 3 4

This surface is rejected because the points are on two different sheets of

the hyperboloid

$$x^2 + y^2 - 7z^2 + 20z - 13 = 0.$$

However, the surface

```
j   Z   2 1 3 4 5 9.380832
```

which has the same surface equation as above is accepted because all coordinates lie on a single surface, the right sheet of the hyperboloid.

```
Example 5:   1   Y  -3 2  2 1
             2   Y   2 3  3 3  4 2
             3   Y   2 1  3 1  4 2
```

This final example defines a cell bounded by a cone, hyperboloid, and an ellipsoid. The three surfaces define the donut-like cell that is symmetric about the y-axis. A cross section of this cell is seen in Figure 3.2. One surface goes through the points (-3,2) and (2,1). The second surface goes

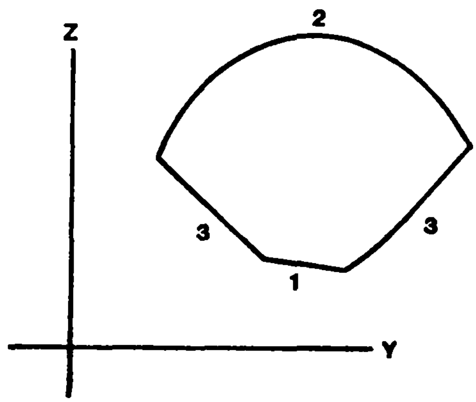


Figure 3.2

through (2,3), (3,3), and (4,2). The last surface is defined by the points (2,1), (3,1), and (4,2). These coordinate points are in the form (y,r). Using these cards, MCNP indicates that surface 1 is a cone of one sheet, surface 2 is an ellipsoid, and surface 3 is a hyperboloid of one sheet. The equation coefficients for the standard surface equations are printed out for the various surfaces when the PRINT input card or execution option is used. For example, an SQ card defining surface 3 is

```
3     SQ    1 -1.5 1 0 0 0 -.625 0 2.5 0
```

C. *General Plane Defined by Three Points*

Form: j n P X_1 Y_1 Z_1 X_2 Y_2 Z_2 X_3 Y_3 Z_3

j = surface number: $1 \le j \le 99999$
n = absent for no coordinate transformation, or
= number of TRn card.
(X_i, Y_i, Z_i) = coordinates of points to define the plane.

If there are four entries on a P card, they are assumed to be the general plane equation coefficients as in Table 3.1. If there are more than four entries, they give the coordinates of three points lying in the desired plane. The code converts them to the required surface coefficients to produce the plane

$$Ax + By + Cz - D = 0.$$

The sense of the plane is determined by requiring the origin to have negative sense. If the plane passes through the origin (D=0), the point $(0,0,\infty)$ has positive sense. If this fails (D=C=0), the point $(0,\infty,0)$ has positive sense. If this fails (D=C=B=0), the point $(\infty,0,0)$ has positive sense. If this fails, the three points lie in a line and a fatal error is issued.

IV. *DATA CARDS*

All MCNP input cards other than those for cell and surface are entered after the second blank card delimiter. These cards fall into the following categories:

CHAPTER 3
Data Cards

These card categories are described below. Only the cards listed on page 168 are allowed in a continue-run input file. No data card can be used more than once with the same number or particle type designations. For example, M1 and M2 are acceptable, as are CUT:N and CUT:P, but two M1 cards or two CUT:N cards are disallowed.

A. Problem Type (MODE) Card

Form: MODE $x_1 \ . \ . \ . \ x_i$

x_i = N for neutron transport
P for photon transport

Default: If the MODE card is omitted, MODE N is assumed.

Use: A MODE card is required for Mode N P and MODE P calculations.

MODE N P will cause both neutrons and photons to be transported. Photons will be generated in neutron collisions, but there is no provision for neutrons to be generated in photon collisions. All source particles in Mode N must be neutrons and in Mode P must be photons. In MODE N P the source particles can be either all neutrons or both neutrons and photons.

B. Cell and Surface Parameter Cards

Most of the following cards define values of cell parameters. AREA defines surface parameters. VECT, WWE, WWP, and WWGE are included here because they are closely related to some of the cell parameter cards.

Only the importance cards are required (except when the weight window is used); most of the others are for optional variance reduction techniques.

Entries on a cell or surface parameter card correspond in order to the cell or surface cards that appear earlier in the INP file. To get to the particular cell(s) or surface(s) on a card, you must supply the appropriate default values on the cards as spacers (the nR repeat or nJ jump features may help). The number of entries on a cell or surface parameter card should always equal the number of cells or surfaces in the problem or a FATAL error will result.

Many of these cards require a knowledge of both the Monte Carlo method and the particular variance reduction technique being used. Chapter 2 and some of the references listed at the end of the manual may provide some of this knowledge.

1. IMP Cell Importance Cards

Form: IMP:n x_1 x_2 . . . x_n

n = N for neutrons and P for photons
x_i = importance for cell i, i=1,2, ... number of cells in the problem

Default: If an IMP:P card is omitted in a MODE N P problem, all photon cell importances are set to unity unless the neutron importance is 0. Then the photon importance is 0 also.

Use: An IMP:N card is required for Mode N and Mode N P problems and an IMP:P card is required for Mode P problems unless weight window cards are used.

The importance of a cell is used to terminate the particle's history if the importance is zero, for geometry splitting and Russian roulette as described on page 123 to help particles move to more important regions of the geometry, and in the weight cutoff game described on page 257.

Because the user may wish to split photons differently than neutrons in a neutron/photon problem, two sets of importances can be used. It is a fatal error if the number of entries on either card (IMP:N or IMP:P) is not equal to the number of cells in the problem. The nR repeat and nM multiply features are especially useful with this card.

Example: IMP:N 1 2 2M 0 1 20R

The neutron importance of cell 1 is 1, cell 2 is 2, cell 3 is 4, cell 4 is 0, and cells 5 through 25 is 1. A track will be split 2 for 1 going from cell 2 into cell 3, each new track having half the weight of the original track before splitting. A track moving in the opposite direction will be terminated in about half (that is, probability=0.5) the cases but followed in the remaining cases with twice the weight.

Remember that tracks and contributions to detectors and DXTRAN spheres are killed in cells of zero importance.

A track will neither be split nor rouletted when it enters a void cell even if the importance ratio of the adjacent cells would normally call for a split or roulette. However, the importance of the nonvoid cell it left is remembered and splitting or Russian roulette will be played when the particle next enters a nonvoid cell. As an example of the benefit of not splitting into a void, consider a long, void pipe surrounded by a material like concrete where the importances are decreasing radially away from the pipe. Considerable computer time can be wasted by tracks bouncing back and forth across the pipe and doing nothing but splitting then immediately undergoing roulette. Splitting into a void increases the time per history but has no counterbalancing effect on the expected history variance. Thus the FOM is reduced by the increased time per history.

2. VOL Cell Volume Card

Form: VOL x_1 x_2 ... x_i
or: VOL NO

x_i = volume of cell i where i=1, 2, ... number of cells in the problem.

NO = no volumes or areas are calculated.

Default: MCNP attempts to calculate the volume of all cells unless "NO" appears on the VOL card. If no value is entered for a cell on the VOL card, the calculated volume is used.

Use: Optional card used to input cell volumes.

With the VOL card, if the number of entries does not equal the number of cells in the problem, it is a fatal error. Use the nJ feature to skip over cells for which you do not want to enter values. The entry NO on the VOL card will bypass the volume calculation altogether. It can be followed by values you want the code to use.

Volumes or masses of cells are required for some tallies. MCNP calculates the volumes of all cells that are rotationally symmetric (generated by surfaces of revolution) about any axis, even a skew axis. It

will also calculate the volumes of polyhedral cells. As a byproduct of the volume calculation, areas and masses are also calculated. These volumes, areas, and masses can be printed in the OUTP file. The user can enter values on the VOL card for the volume of any cell and these values, instead of the calculated values, will be used for tally purposes. If a cell volume required for a tally cannot be calculated and is not entered on the VOL or SDn cards, a fatal error message is printed.

The VOL card provides an alternative way to enter volumes required by tallies. Normally the SDn card would be used. The VOL card can be used only for cell volumes, whereas the SDn card can be used for cell and segment volumes or masses.

Volumes of cells or segments that cannot be calculated by MCNP or by the user can be obtained in a separate MCNP run using the ray-tracing technique described on page 160.

3. AREA Surface Area Card

Form: AREA $x_1 \; . \; . \; . \; x_i \; . \; . \; . \; x_n$

x_i = area of surface i where i=1, 2, ... number of surfaces in the problem.

Default: MCNP attempts to calculate the area of all surfaces. If no value is entered for a surface on the AREA card, the calculated area, if any, is used.

Use: Optional card used to input surface areas.

This card is analogous to the VOL card. MCNP calculates the area of surfaces as a byproduct of the volume calculation. If the volume of all cells on either side of the surface can be calculated, the area of the surface will be calculated. Otherwise the area calculation will fail. A ftal error occurs if an area is required for tallying purposes and is not available either from the MCNP calculation or from an AREA or SDn card.

The AREA card provides an alternative way to enter areas required by tallies. Normally the SDn card would be used. The AREA card can be used only for areas of whole surfaces, whereas the SDn card can be used for areas of surface segments as well as whole surfaces.

4. PWT Photon Weight Card

Form: PWT $W_1 \; W_2 \; . \; . \; . \; W_i \; . \; . \; . \; W_I$

W_i = relative threshold weight of photons produced at neutron

collisions in cell i
I = number of cells in the problem.

Default: $W_i = -1$.

Use: Recommended for MODE N P problems.

The PWT card is used only in Mode N P problems. Its purpose is to control the number and weight of neutron-induced photons produced at neutron collisions. Its application is further discussed on page 60.

For each cell with a positive W_i entry, only neutron-induced photons with weights greater than W_i*I_i/I_s are produced, where I_i and I_s are the importances of the collision and source cells, respectively. Russian roulette is played to determine if a neutron-induced photon with a weight below this value survives.

For each cell with a negative W_i entry, only neutron-induced photons with weights greater than $-W_i*W_s*I_i/I_s$ are produced, where W_s is the starting weight of the neutron for the history being followed, and I_i and I_s are the importances of the collision and source cells, respectively. Russian roulette is played to determine if a neutron-induced photon with weight below this value survives.

If $W_i = 0$, exactly one photon will be generated at each neutron collision in cell i, provided that photon production is possible. If $W_i = -1.0E6$, photon production in cell i is turned off.

The PWT card controls the production of neutron-induced photons by comparing the total weight of photons produced with a relative threshold weight specified on the PWT card. This threshold weight is relative to the cell importance and, if $W_i < 0$, to the source neutron weight. If more neutron-induced photons are desired, the absolute value of W_i should be lowered to reduce the weight and therefore increase the number of photons. If fewer neutron-induced photons are desired, the absolute value of W_i should be increased.

Following are some tips on how to choose PWT card entries.

For problems with energy-independent photon weight windows, the entries on the PWT card should be identical to those on the WWN1:P card. Neutron-induced photons will then be born inside the photon weight window bounds as much as possible.

For problems with energy-dependent photon weight windows, the entries on the PWT card should be the same as the lowest nonzero weight window for any weight window energy group on the WWNn:P cards.

For problems using photon cell importances (IMP:P), rather than photon weight windows (WWNn:P), a good first guess for PWT card entries is either the default value, $W_i = -1$, or set W_i in every cell to the average source weight.

5. EXT Exponential Transform Card

Form: EXT:n A_1 A_2 ... A_i ... A_I

n = N for neutrons and P for photons
A_i = entry for cell i. Each entry A_i is of the form
A = QVm, where Q describes the amount of stretching and
Vm defines the stretching direction.
I = number of cells in the problem.

Default: No transform, A_i = 0.

Use: Optional. Use cautiously with the weight window.

The exponential transform method stretches the path length between collisions in a preferred direction by adjusting the total cross section as follows: $\Sigma_t^* = \Sigma_t(1 - p\mu)$, where
Σ_t^* = artificially adjusted total cross section;
Σ_t = true total cross section;
p = the stretching parameter; and
μ = cosine of angle between particle direction and stretching direction.

The stretching parameter p can be specified by the stretching entry Q three ways:

Q = 0 ;	p = 0	exponential transform not used
Q = p ;	0 < p < 1	constant stretching parameter
Q = S ;	$p = \Sigma_a/\Sigma_t$,	where Σ_a is the capture cross section.

Letting $p = \Sigma_a/\Sigma_t$ can be used for implicit capture along a flight path, as described on page 63.

The stretching direction is defined by the Vm part of each A_i entry on the EXT card with three options.

1. Omit the Vm part of the A_i entry; that is, enter only the stretching entry A_i = Q for a given cell. This causes the stretching to be in the particle direction (μ=1), independent of the particle direction and is not recommended unless you want to do implicit capture along a flight path, in which case A_i=Q=S and the distance to scatter rather than the distance to collision is sampled.

2. Specify the stretching direction as Vm, the line from the collision point to the point (x_m,y_m,z_m), where (x_m,y_m,z_m) is specified on the VECT card (see next section). The direction cosine μ is now the cosine of the angle between the particle direction and the line drawn from the

collision point to point (x_m, y_m, z_m). The sign of A_i governs whether stretching is toward or away from (x_m, y_m, z_m).

3. Specify the stretching direction as Vm = X, Y, or Z, so the direction cosine μ is the cosine of the angle between the particle direction and the X-, Y-, or Z-axis, respectively. The sign of A_i governs whether stretching is toward or away from the X-, Y-, or Z-axis.

Example: EXT:N 0 0 .7V2 S -SV2 -.6V9 0 .5V9 SZ -.4X
VECT V9 0 0 0 V2 1 1 1

The 10 entries are for the 10 cells in this problem. Path length stretching is not turned on for photons or for cells 1, 2, and 7. Following is a summary of path length stretching in the other cells.

cell	A_i	Q	Vm	stretching parameter	direction
3	.7V2	.7	V2	p = .7	toward point (1,1,1)
4	S	S		$p = \Sigma_a/\Sigma_t$	particle direction
5	-SV2	S	-V2	$p = \Sigma_a/\Sigma_t$	away from point (1,1,1)
6	-.6V9	.6	-V9	p = .6	away from origin
8	.5V9	.5	V9	p = .5	toward origin
9	SZ	S	Z	$p = \Sigma_a/\Sigma_t$	along +Z-axis
10	-.4X	.4	-X	p = .4	along -X-axis

The exponential transform should not be used in the same cell as forced collisions or without good weight control, such as the weight window. The transform works well only when the particle flux has an exponential distribution, such as in highly absorbing problems.

6. <u>VECT Vector Input Card</u>

Form: VECT Vm x_m y_m z_m ... Vn x_n y_n z_n ...

m,n = any numbers to uniquely identify vectors Vm, Vn ...
x_m y_m z_m = coordinate triplets to define vector Vm.

Default: None.

Use: Optional.

The entries on the VECT card are quadruplets which define any number of vectors for either the exponential transform or user patches. See the EXT card (page 189) for a usage example.

7. FCL Forced Collision Card

Form: FCL:n $x_1 \; x_2 \; . \; . \; . \; x_i \; . \; . \; . \; x_I$

n = N for neutrons and P for photons
x_i = forced collision control for cell i. $-1 < x_i < 1$
I = number of cells in the problem.

Default: $x_i = 0$, no forced collisions.

Use: Optional. Exercise caution..

The FCL card controls the forcing of neutron or photon collisions in each cell. This is particularly useful for generating contributions to point detectors or DXTRAN spheres. The weight window game at surfaces is not played when entering forced collision cells.

If $x_i \neq 0$, all particles entering cell i are split into collided and uncollided parts with the appropriate weight adjustment (see page 127). If $|x_i| < 1$, Russian roulette is played on the collided parts with survival probability $|x_i|$ to keep the number of collided histories from getting too large. Fractional x_i entries are recommended if a number of forced collision cells are adjacent to each other.

If $x_i < 0$, the forced collision process applies only to particles entering the cell. After the forced collision, the weight cutoff is ignored and all subsequent collisions are handled in the usual analog manner. Weight windows are not ignored and are applied after contributions are made to detectors and DXTRAN spheres.

If $x_i > 0$, the forced collision process applies both to particles entering cell i and to the collided particles surviving the weight cutoff or weight window games. Particles will continue to be split into uncollided and (with probability $|x_i|$) collided parts until killed by either weight cutoff or weight windows.

Usage tips:

Let $x_i = 1$ or -1 unless a number of forced collision cells are adjacent to each other or the number of forced collision particles produced is higher than desired. Then fractional values are usually needed.

When weight window bounds bracket the typical weight entering the cell, choose $x_i > 0$. When weight window bounds bracket the weight typical of force collision particles, choose $x_i < 0$.

When using importances, use $x_i > 0$ because $x_i < 0$ turns off the weight cutoff game.

8-10. Weight Window Cards

Weight windows provide an alternative means to importances (IMP:n cards) and energy splitting (ESPLT:n cards) for specifying space and energy importance functions. The advantages of weight windows are that (1) they provide an importance function in both space and energy; (2) they control particle weights; (3) they are more compatible with other variance reduction features such as the exponential transform (EXT:n card); (4) they can be applied at surface crossings, collisions, or both; (5) the severity of splitting or Russian roulette can be controlled; (6) they can be turned off in selected space or energy regions; and (7) they can be automatically generated by the weight window generator. The disadvantages are that (1) weight windows are not as straightforward as importances; and (2) when the source weight is changed, the weight windows may have to be renormalized. You are strongly advised to read the section on weight windows beginning on page 140.

(a) WWE Weight Window Energies

Form: WWE:n E_1 E_2 ... E_j; $j \le 99$

n = N for neutrons and P for photons
E_i = energy interval, $E_{i-1} < E < E_i$, $E_0 = 0$

Default: If this card is omitted and the weight window is used, a single energy interval will be established corresponding to the energy limits of the problem being run.

Use: Optional. Use only with WWN card.

The weight window energy card defines the energy intervals for which weight window bounds will be specified on the WWN card. The minimum energy, which is not entered on the WWE card, is zero.

(b) WWN Weight Window Bounds

Form: WWNi:n w_{i1} w_{i2} ... w_{ij} ... w_{iJ}

n = N for neutrons and P for photons
w_{ij} = lower weight bound in cell j and energy interval $E_{i-1} < E < E_i$, E_0=0, as defined on the WWE card. If no WWE card, i = 1.
J = number of cells in the problem.

Default: None.

Use: Weight windows (WWN and WWP cards) are required unless importances (IMP card) are used.

The WWN card specifies the lower weight bound of the space and energy dependent weight windows. It must be used with the WWP card, and, if the weight windows are energy dependent, with the WWE card. The IMP:n card should not be used if a WWN:n card, where n is the same particle type, is used.

If $w_{ij} < 0$, any particle entering cell j is killed. That is, negative entries correspond to zero importance. If negative entries are used for one energy group, they should be used for all the other energy groups in the same cell.

If $w_{ij} > 0$, particles entering or colliding in cell j are split or rouletted according to the options on the WWP card, described in the next section.

If $w_{ij} = 0$, the weight window game is turned off in cell j for energy bin i and the weight cutoff game is turned on. The weight cutoffs on the CUT card should be specified as the lowest permissible weights desired in the problem. Otherwise, most particles entering cells with $w_{ij} = 0$ will be killed by the weight cutoff. The capability to turn the weight window game off in various energy and spatial regions is useful when these regions cannot be characterized by a single importance function or set of weight window bounds.

In terms of the weight window, particle weight bounds are always absolute and not relative; you have to explicitly account for weight changes from any other variance reduction techniques such as source biasing. You must specify one lower weight bound per cell per energy interval. There must be no holes in the specification; that is, if WWNi is specified, WWNj for $1 < j < i$ must also be specified.

Example 1:	WWE:N	E_1 E_2 E_3
	WWN1:N	w_{11} w_{12} w_{13} w_{14}
	WWN2:N	w_{21} w_{22} w_{23} w_{24}
	WWN3:N	w_{31} w_{32} w_{33} w_{34}

These cards define three energy intervals and the weight window bounds for a four-cell neutron problem.

Example 2: WWN1:P w_{11} w_{12} w_{13}

This card, without an accompanying WWE card, defines an energy independent photon weight window for a three-cell problem.

(c) WWP Weight Window Parameter Card

Form: WWP:n WUPN WSURVN MXSPLN MWHERE SWITCHN

n = N for neutrons and P for photons

WUPN = If the particle weight goes above WUPN times the lower weight bound, the particle will be split. Required: WUPN ≥ 2.

WSURVN = If the particle survives the Russian roulette game, its weight becomes WSURVN times the lower weight bound. Required: 1 < WSURVN < WUPN.

MXSPLN = No particle will ever be split more than MXSPLN-for-one or be rouletted more harshly than one-in-MXSPLN. Required: MXSPLN > 1.

MWHERE = decides where to check a particle's weight.
-1 means check the weight at collisions only
0 means check the weight at surfaces and collisions
1 means check the weight at surfaces only

SWITCHN = special parameter allowing the lower weight bounds to be specified as SWITCHN > 0 divided by the cell importances from the IMP card.

Defaults: WUPN=5; WSURVN=0.6*WUPN; MXSPLN=5; MWHERE=0; SWITCHN=0

Use: Weight windows are required unless importances are used.

The WWP card contains parameters that control use of the weight window lower bounds specified on the WWN cards or, if SWITCHN > 0, the IMP cards. Having SWITCHN > 0 and also having WWNi cards is a fatal error. If SWITCHN is zero, the lower weight bounds must be specified with the WWNi cards.

(d) Using Existing Cell Importances to Specify the Lower Weight Bound

An energy-independent weight window may be specified using existing importances from the IMP card and setting the fifth entry (SWITCHN) on the WWP card to a positive constant C. If this option is selected, the lower weight bounds for the cells become C/I, where I is the cell importance. The remaining entries on the WWP card are entered as described above. A suggested value for C is one in which source particles start within the weight window. If that is not possible, your window is probably too narrow or you need to respecify your source.

11-12. Weight Window Generation Cards

The weight window generator estimates the importances of the space-energy regions of phase space specified by the user. The space-energy weight window parameters are then calculated inversely proportional to the importances. Sometime in the future a space-energy-angle weight window and generator may be available, but for now, both the weight window and the generator cannot be used on problems with heavy angular dependence, such as streaming problems. The generator should be used in problems without a heavy angular dependence and with the geometry adequately specified for sampling purposes.

Recall that the generator estimates the average importance of a phase-space cell. If the cells are too large, the importance variation inside the cell will be large and the average importance will not represent the cell. Inadequate geometry specification also occurs with large importance differences between adjacent cells. Fortunately, the generator provides information about whether the geometry specification is adequate for sampling purposes.

The user is advised to become familiar with the section on the weight window in Chapter 2 before trying to use the weight window generator.

(a) WWG Weight Window Generation

Form: WWG I_t I_c W_g x_r y_r z_r J

I_t = problem tally number (n of the Fn card). The particular tally bin for which the weight window generator is optimized is defined by the TFn card.
I_c = reference cell, typically a source cell.
W_g = value of the generated lower weight window bound for cell I_c. If W_g = 0, it will be half the average source weight.
x_r y_r z_r = coordinates of point which particles should move toward or away from.
J = debug print flag. J=0/1 = off/on.

Default: No weight window values are generated.

Use: Optional.

The WWG card causes the optimum importance function for tally I_t to be generated on the WWP and WWNi cards, which are printed, evaluated, and summarized in the OUTP file. This resulting weight window importance function can then be used in subsequent runs. For many problems, this

generated importance function is superior to anything an experienced user can guess on an IMP card. To generate an energy-dependent weight window, use the WWGE card described below.

(b) WWGE Weight Window Generation Energies

Form: WWGE:n E_1 E_2 E_i ... E_j; $j \le 99$

n = N for neutrons and P for photons
E_i = upper energy bound for weight window group to be generated, $E_{i+1} > E_i$.

Default: If this card is omitted and the weight window is used, a single energy interval will be established corresponding to the energy limits of the problem being run. If the card is present but has no entries, ten energy bins will be generated with energies of $E_i = 10^{i-8}$ MeV and j = 10.

Use: Optional.

13. PDn Detector Contribution Card

Form: PDn P_1 P_2 ... P_i ... P_I

n = tally number
P_i = probability of contribution to detector n from cell i
I = number of cells in the problem.

Default: $P_i = 1$.

Use: Optional. Consider also using the DD card, page 243.

The PDn card reduces the number of contributions to detector tallies from selected cells that are relatively unimportant to a given detector, thus saving computing time. At each collision in cell i, the detector tallies are made with probability P_i ($0 \le P_i \le 1$). The tally is then increased by the factor $1/P_i$ to obtain unbiased results for all cells except those where P_i=0. This enables you to increase the running speed by setting P_i<1 for cells many mean free paths from the detectors. It also selectively eliminates detector contributions from cells by setting the P_i's to zero.

A useful application of this card is to tally individual cell contributions for several identical point detectors. A different PDn for

each detector selects the resulting tallies by cell. (This could also be done with a TALLYX patch.)

A default set of probabilities can be established for all tallies by use of a PD0 (zero) card. These default values will be overridden for a specific tally n by values entered on a PDn card.

14. DXC DXTRAN Contribution Card

Form: DXC:n P_1 P_2 ... P_i ... P_I

n = N for neutrons and P for photons
P_i = Probability of contribution to DXTRAN spheres from cell i
I = number of cells in the problem

Default: $P_i = 1$.

Use: Optional. Consider also using the DD card, page 243.

This card is analogous to the above PDn card but is used for contributions to DXTRAN spheres.

C. Coordinate Transformation (TRn) Card

Form: TRn O_1 O_2 O_3 B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 M

n = number of the transformation: $1 < n < 999$. *TRn means that the B_i are angles in degrees rather than being the cosines of the angles.
O_1 O_2 O_3 = displacement vector of the transformation.
B_1 to B_9 = rotation matrix of the transformation.
M = 1 means that the displacement vector is the location of the origin of the auxiliary coordinate system, defined in the main system.
= −1 means that the displacement vector is the location of the origin of the main coordinate system, defined in the auxiliary system.

Default: TRn 0 0 0 1 0 0 0 1 0 0 0 1 1

The maximum number of transformations in a single problem is 10.

The B matrix specifies the relationship between the directions of the axes of the two coordinate systems. B_i is the cosine of the angle (or the angle itself, in degrees in the range from 0 to 180, if the optional asterisk is used) between an axis of the main coordinate system (x,y,z) and an axis of the auxiliary coordinate system (x′,y′,z′) as follows:

Element	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9
Axes	x,x′	y,x′	z,x′	x,y′	y,y′	z,y′	x,z′	y,z′	z,z′

The meanings of the B_i do not depend on M. It is usually not necessary to enter all of the elements of the B matrix. The nJ special syntax (see page 170) may be used to skip over unused elements. The following five patterns are acceptable:

1. All nine elements.
2. Two of the three vectors either way in the matrix (6 values). MCNP will create the third vector by cross product.
3. One vector each way in the matrix (5 values). The component in common must be less than 1. MCNP will fill out the matrix by the Eulerian angles scheme.
4. One vector (3 values). MCNP will create the other two vectors in some arbitrary way.
5. None. MCNP will create the identity matrix.

A vector consists of the three elements in either a row or a column in the matrix. In all cases MCNP cleans up any small nonorthogonality and normalizes the matrix. In this process, exact vectors like (1,0,0) are left unchanged. A warning message is issued if the nonorthogonality is more than about 0.001 radian.

Pattern #5 is appropriate when the transformation is a pure translation. Pattern #4 is appropriate when the auxiliary coordinate system is being used to describe a set of surfaces that are all surfaces of rotation about a common skew axis. Patterns 2 and 3 are about equally useful in more general cases. Pattern #1 is required if one of the systems is right handed and the other is left handed.

Coordinate transformations in MCNP are used to simplify the geometrical description of surfaces and to relate the coordinate system of a surface source problem to the coordinate system of the problem that wrote the surface source file. See the surface source SSR card on page 214.

To use a transformation to simplify the description of a surface, choose an auxiliary coordinate system in which the description of the surface is easy, include a transformation number n on the surface card, and specify the transformation on a TRn card. See page 286 for an example showing how much easier it is to specify a skewed cylinder this way than as

a GQ surface. Often a whole cluster of cells will have a common natural coordinate system. All of their surfaces can be described in that system, which can then be specified by a single TRn card.

```
Example:    17     4  PX  5
            TR4    7  .9  1.3    0 -1  0  0  0  1  -1  0  0
```

Surface 17 is set up in an auxiliary coordinate system that is related to the main coordinate system by transformation number 4. (Presumably there are many other surfaces in this problem that are using the same transformation, probably because they came from the input file of an earlier problem. Otherwise there would be no reason to use a transformation to set up a surface as simple as a plane perpendicular to a coordinate axis.) MCNP will produce coefficients in the main coordinate system as if surface 17 had been entered as

```
17 P 0 -1 0 4.1
```

It will not produce

```
17 PY -4.1
```

that is located at the same place in space, because this PY surface has the wrong sense.

More examples of this transformation feature are in Chapter 4.

D. Source Specification

Every MCNP problem has one of four sources: general source (SDEF card), surface source (SSR card), criticality source (KCODE card), or user-supplied source (default if SDEF, SSR, and KCODE are all missing). All but the criticality source can use source distribution functions, which are specified on SIn, SPn, SBn, and DSn cards.

The following cards are used to specify the source.

Mnemonic (Columns 1-5)	Card Type	Page
SDEF	General source	200
SIn	Source information	206
SPn	Source probability	206
SBn	Source bias	206
DSn	Dependent source	210
SCn	Source comment	212
SSW	Surface source write	214
SSR	Surface source read	214
KCODE	Criticality source	217
KSRC	Source points	218

The MODE card also serves as part of the source specification in some cases by implying the type of particle to be started from the source.

The source has to define the values of the following MCNP variables for each particle it produces:

ERG	the energy of the particle (MeV)
TME	the time when the particle started (shakes)
UUU, VVV, WWW	the direction of the flight of the particle
XXX, YYY, ZZZ	the position of the particle
IPT	the type of the particle
WGT	the statistical weight of the particle
ICL	the cell where the particle started
JSU	the surface where the particle started, or zero if the starting point is not on any surface

Additional variables may have to be defined if there are point detectors or DXTRAN spheres in the problem.

1. SDEF General Source Card

Form: SDEF source variable = specification . . .

Use: Required for problems using the general source.

The equal signs are optional. The source variables are not quite the same as MCNP variables that the source must set. Many are intermediate quantities that control the sampling of the final variables. All have default values. The specification of a source variable has one of these three forms:

1. explicit value,
2. a distribution number prefixed by a D, or
3. the name of another variable prefixed by an F, followed by a distribution number prefixed by a D. Var = Dn means that the value of source variable var is sampled from distribution n. Var Fvar′ Dn means that var is sampled from distribution n that depends on the variable var′. Only one level of dependence is allowed. Each distribution may be used for only one source variable.

The above scheme translates into three levels of source description. The first level exists when a source variable has an explicit or default value (for example, a single energy) or a default distribution (for example, an isotropic angular distribution). The second level occurs when a source variable is given by a probability distribution. This level requires the SI and/or SP cards. The third level occurs when a variable depends on another variable. This level requires the DS card.

MCNP samples the source variables in an order set up according to the needs of the particular problem. Each dependent variable must be sampled after the variable it depends on has been sampled. If the value of one variable influences the default value of another variable or the way it is sampled, as SUR influences DIR, they may have to be sampled in the right order. The scheme used in MCNP to set up the order of sampling is complicated and may not always work. If it fails, a message will be printed. The fix in such instances may be to use explicit values or distributions instead of depending on defaults.

The type of the particle is determined by the MODE card. See page 184. For MODE N and MODE N P, the particle type is neutron. For MODE P, the particle type is photon.

Table 3.2 summarizes the source variables and lists their defaults.

Table 3.2
Source Variables

Variable	Meaning	Default
CEL	Cell	Determined from XXX,YYY,ZZZ and possibly UUU,VVV,WWW
SUR	Surface	Zero (means cell source)
ERG	Energy (MeV)	14 MeV
TME	Time (shakes)	0
DIR	μ, the cosine of the angle between VEC and UUU,VVV,WWW (Azimuthal angle is always sampled uniformly in 0° to 360°)	Volume case: μ is sampled uniformly in −1 to 1. (isotropic) Surface case: $p(\mu)=2\mu$ in 0 to 1. (cosine distribution)
VEC	Reference vector for DIR	Volume case: required unless isotropic Surface case: vector normal to the surface with sign determined by NRM
NRM	Sign of the surface normal	+1
POS	Reference point for position sampling	0,0,0
RAD	Radial distance of the position from POS or AXS	0
EXT	Cell case: distance from POS along AXS Surface case: Cosine of angle from AXS	0

AXS	Reference vector for EXT and RAD	No direction
X	x-coordinate of position	No X
Y	y-coordinate of position	No Y
Z	z-coordinate of position	No Z
CCC	Cookie-cutter cell	No cookie-cutter cell
ARA	Area of surface (required only for direct contributions to point detectors from plane surface source.)	Calculated area, or 1 if not calculable
WGT	Particle weight	1
EFF	Rejection efficiency criterion for position sampling	.01

WGT and EFF are different from the other source variables in that the specification must be an explicit value. A distribution is not allowed.

Most of the source variables are scalars. VEC, POS, and AXS are vectors. Where a value of a source variable is required, as on SDEF, SI, or DS cards, usually a single number is appropriate, but with VEC, POS, and AXS, the value must actually be a triplet of numbers, the x, y, and z components of the vector.

The source variables SUR, POS, RAD, EXT, AXS, X, Y, Z, and CCC are used in various combinations to determine the coordinates (x,y,z) of the starting positions of the source particles. With them you can specify three different kinds of volume distributions and three different kinds of distributions on surfaces. Degenerate versions of those distributions provide line and point sources. More elaborate distributions can be approximated by combining several simple distributions, using the S option of the SIn and DSn cards.

The three volume distributions are cartesian, spherical, and cylindrical. The value of the variable SUR is zero for a volume distribution. A volume distribution can be used in combination with the CEL variable to sample uniformly throughout the interior of a cell. A cartesian, spherical, or cylindrical region that completely contains a cell is specified and is sampled uniformly in volume. If the sampled point is found to be inside the cell, it is accepted. Otherwise it is rejected and another point is sampled. If you use this technique, you must make sure that the sampling region really does contain every part of the cell because MCNP has no way of checking for this. Cookie-cutter rejection, described below, can be used instead of or in combination with CEL rejection.

A cartesian volume distribution is specified with the variables X, Y, and Z. A degenerate case of the cartesian distribution in which the three variables are constants defines a point source. A single point source can be specified by giving values to the three variables right on the SDEF card. If there are several source points in the problem, it would usually

be easier to use a degenerate spherical distribution for each point. Other degenerate cases of the cartesian distribution are a line source and a rectangular plane source. A cartesian distribution is an efficient shape for the CEL rejection technique when the cell is approximately rectangular. It is much better than a cylindrical distribution when the cell is a long thin slab. It is, however, limited in that its faces can only be perpendicular to the coordinate axes.

A spherical volume distribution is specified with the variables POS and RAD. X, Y, Z, and AXS must not be specified or it will be taken to be a cartesian or cylindrical distribution. The sampled value of the vector POS defines the center of the sphere. The sampled value of RAD defines the distance from the center of the sphere to the position of the particle. The position is then sampled uniformly on the surface of the sphere of radius RAD. Uniform sampling in volume is obtained if the distribution of RAD is a power law with a=2, which is the default in this case. A common use of the spherical volume distribution is to sample uniformly in the volume between two spherical surfaces. The two radii are specified on the SIn card for RAD and the effect of a SPn −21 2 card is obtained by default (see page 206). If RAD is not specified, the default is zero. This is useful because it specifies a point source at the position POS. A distribution for POS, with an L on the SIn card, is the easiest way to specify a set of point sources in a problem.

A cylindrical volume distribution is specified with the variables POS, AXS, RAD, and EXT. The axis of the cylinder passes through the point POS in the direction AXS. The position of the particles is sampled uniformly on a circle whose radius is the sampled value of RAD, centered on the axis of the cylinder. The circle lies in a plane perpendicular to AXS at a distance from POS which is the sampled value of EXT. A common use of the cylindrical distribution is to sample uniformly in volume within a cylindrical shell. The distances of the ends of the cylinder from POS are entered on the SIn card for EXT and the inner and outer radii are entered on the SIn card for RAD. Uniform sampling between the two values of EXT and power law sampling between the two values of RAD, with a=1 which gives sampling uniform in volume, are provided by default. A useful degenerate case is EXT=0, which provides a source with circular symmetry on a plane.

Warning: Never position any kind of degenerate volume distribution in such a way that it lies on one of the defined surfaces of the problem geometry. Even a bounding surface that extends into the interior of a cell can cause trouble. If possible, use one of the surface distributions instead. Otherwise, move to a position just a little way off of the surface. It will not make any detectable difference in the answers, and it will prevent particles from getting lost.

The value of the variable SUR is nonzero for a distribution on a surface. If X, Y, and Z are specified, their sampled values determine the

position. You must in this case make sure that the point really is on the surface because MCNP does not check. If X, Y, and Z are not specified, the position is sampled on the surface SUR. The shape of the surface, which can be either a spheroid, sphere, or plane, determines the way the position is sampled. Sampling with CEL rejection is not available, but cookie-cutter rejection is and can be used to do anything that CEL rejection would do.

If the value of SUR is the name of a spheroidal surface, the position of the particle is sampled uniformly in area on the surface. A spheroid for this purpose must have its axis parallel to one of the coordinate axes. There is presently no provision for easy nonuniform or biased sampling on a spheroidal surface. A distribution of cookie-cutter cells could be used to produce a crude nonuniform distribution of position.

If the value of SUR is the name of a spherical surface, the position of the particle is sampled on that surface. If the vector AXS is not specified, the position is sampled uniformly in area on the surface. If AXS is specified, the sampled value of EXT is used for the cosine of the angle between the direction AXS and the vector from the center of the sphere to the position point. The azimuthal angle is sampled uniformly in the range from 0° to 360°. A nonuniform distribution of position, in polar angle only, is available through a nonuniform distribution of EXT. A biased distribution of EXT can be used to start more particles from the side of the sphere nearest the tallying regions of the geometry. The exponential distribution function (−31; see page 209) is usually most appropriate for this.

If the value of SUR is the name of a plane surface, the position is sampled on that plane. The sampled value of POS must be a point on the plane. You must make sure that POS really is on the plane because MCNP, for the sake of speed, does not check it. The sampled position of the particle is at a distance from POS equal to the sampled value of RAD. The position is sampled uniformly on the circle of radius RAD centered on POS. Uniform sampling in area is obtained if the distribution of RAD is a power law with $a=1$, which is the default in this case.

Cookie-cutter rejection is available for both cell and surface sources. If CCC is present, the position sampled by the above procedures is accepted if it is within cell CCC and is resampled if it is not, exactly like CEL rejection in the cell source case. You must be careful not to specify a cookie-cutter cell such that MCNP mistakes it for a real cell. There should be no trouble if the cookie-cutter cells are bounded by surfaces used for no other purpose in the problem and if the cookie-cutter cell cards are at the end of the list of cell cards. Don't make a cookie-cutter cell more complicated than it has to be. For a surface source, the only thing that matters is the intersection of the cookie-cutter cell with the source surface. An infinitely long cell of

uniform cross section, bounded by planes and cylinders, is usually adequate for a plane surface source.

Warning: The combination of either CEL or CCC rejection with biased sampling of the position is nearly always an unfair game. If you use this combination, you must make sure that it really is a fair game because MCNP is not able to detect the error.

The source variables SUR, VEC, NRM, and DIR are used to determine the initial direction of flight of the source particles. The direction of flight is sampled with respect to the reference vector VEC, which, of course, can itself be sampled from a distribution. The polar angle is the sampled value of the variable DIR. The azimuthal angle is sampled uniformly in the range from o° to 360°. If VEC and DIR are not specified for a volume distribution of position (SUR=0), an isotropic distribution of direction is produced by default. If VEC is not specified for a distribution on a surface (SUR≠0), the vector normal to the surface, with the sign determined by the sign of NRM, is used by default. If DIR is not specified for a distribution on a surface, the cosine distribution p(DIR)=2*DIR, 0<DIR<1, is used by default. A biased distribution of DIR can be used to make more source particles start in a direction towards the tallying regions of the geometry. The exponential distribution function (−31; see page 209) is usually most appropriate for this.

Discrete values of DIR are allowed. DIR=1 gives a monodirectional source in the direction of VEC. This is sometimes useful as an approximation to an actual source that is at a large distance from the geometry of the problem. In most cases discrete values of DIR will prevent direct contributions to point detectors from being scored. The direct contribution will be scored only if the source is on a plane surface, is sampled uniformly in area within a circle (using RAD sampled from SP −21 1), VEC is perpendicular to the surface (the default), and DIR=1. A cookie-cutter cell is allowed and a value of ARA is usually necessary.

The efficiency criterion EFF applies to both CCC and CEL rejection. If in any source cell or cookie-cutter cell the acceptance rate is too low, the problem is terminated for inefficiency. The criterion for termination is MAX(number of successes,10) < EFF * number of tries. The default value of EFF, 0.01, lets a problem get by at rather low efficiency, but for the rare problem in which low source efficiency is unavoidable, you may need to specify a lower value for EFF.

Example: SDEF (no entries)

This card specifies a 14−MeV isotropic point source at position 0,0,0 at time 0 with weight 1 (all defaults).

2. SIn Source Information Card
3. SPn Source Probability Card
4. SBn Source Bias Card

Form: SIn option $I_1 \dots I_k$

n = distribution number (n = 1,999)
option = how the I_i's are to be interpreted. Allowed values are:
blank – bin boundaries for a histogram distribution
H – bin boundaries for a histogram distribution, for scalar variables only
L – discrete source variable values
A – points where a probability density distribution is defined
S – distribution numbers
$I_1 \dots I_k$ = source variable values or distribution numbers

Default: SIn H $I_i \dots I_k$

Form: SPn option $P_1 \dots P_k$
or: SPn f a b

n = distribution number (n = 1,999)
option = how the P_i are to be interpreted. Allowed values are:
blank – same as D for an H or L distribution. Probability density for an A distribution on SI card.
D – bin probabilities for an H or L distribution on SI card
C – cumulative bin probabilities for an H or L distribution on SI card
V – for cell distributions only. Probability is proportional to cell volume (times P_i if the P_i are present)
$P_i \dots P_k$ = source variable probabilities
f = designator (negative number) for a built-in function
a,b = parameters for the built-in function (see Table 3.3)

Default: SPn D $P_1 \dots P_k$

Form: SBn option $B_1 \dots B_k$
or: SBn f a b

n, option, f, a, and b are the same as for the SPn card, except that the only values allowed for f are –21 and –31
$B_i \dots B_k$ = source variable biased probabilities

Default: SBn D B_1 ... B_k

The first form of the SP card, where the first entry is positive or nonnumeric, indicates that it and its SI card define a probability distribution function. The entries on the SI card are either values of the source variable or, when the S option is used, distribution numbers. The entries on the SP card are probabilities that correspond to the entries on the SI card.

When the H option is used, the numerical entries on the SI card are bin boundaries and must be monotonically increasing. The first numerical entry on the SP card must be zero and the following entries are bin probabilities or cumulative bin probabilities, depending on whether the D or C option is used. The probabilities need not be normalized. The variable is sampled by first sampling a bin according to the bin probabilities and then sampling uniformly within the chosen bin.

When the A option is used, the entries on the SI card are values of the source variable at which the probability density is defined. The entries must be monotonically increasing, and the lowest and highest values define the range of the variable. The numerical entries on the SP card are values of the probability density corresponding to the values of the variable on the SI card. They need not be normalized. In the sampling process, the probability density is linearly interpolated between the specified values. The first and last entries on the SP card will typically be zero, but nonzero values are also allowed.

When the L option is used, the numerical entries on the SI card are discrete values of the source variable, such as cell numbers or the energies of photon spectrum lines. The entries on the SP card are either probabilities of those discrete values or cumulative probabilities, depending on whether the D or C option is used. The entries on the SI card need not be monotonically increasing.

The S option allows sampling among distributions, one of which is chosen for further sampling. This feature makes it unnecessary to fold distributions together and is essential if some of the distributions are discrete and others are linearly interpolated. The distributions listed on an SI card with the S option can themselves also have the S option. MCNP can handle this structure to a depth of about 20, which should be far more than necessary for any practical problem. Each distribution number on the SI card can be prefixed with a D, or the D can be omitted. If a distribution number is zero, the default value for the variable is used. A distribution can appear in more than one place with an S option, but a distribution cannot be used for more than one source variable.

The V option on the SP card is a special case used only when the source variable is CEL. This option is useful when the cell volume is a

factor in the probability of particle emission. If MCNP cannot calculate the volume of such a cell and the volume is not given on a VOL card, you have a fatal error.

The SB card is used to provide a probability distribution for sampling that is different from the true probability distribution on the SP card. Its purpose is to bias the sampling of its source variable to improve the statistical accuracy of the problem. The weight of each source particle is adjusted to compensate for the bias. All rules that apply to the first form of the SP card apply to the SB card.

The second form of the SP card, where the first entry is negative, indicates that a built-in analytic function is to be used to generate a continuous probability density function for the source variable. Built-in functions can be used only for scalar variables. See Table 3.3 for a description of these functions.

Table 3.3
Built-In Functions for Source Probability and Bias Specification

Source Variable	Function No. and Input Parameters			Description
ERG	−2	a		Maxwell fission spectrum
ERG	−3	a	b	Watt fission spectrum
ERG	−4	a	b	Gaussion fusion spectrum
ERG	−5	a		Evaporation spectrum
ERG	−6	a	b	Muir velocity Gaussian fusion spectrum
ERG	−7	a	b	Spare
DIR, RAD, or EXT	−21	a		Power law $p(x)=c\|x\|^a$
DIR or EXT	−31	a		Exponential: $p(\mu)=ce^{a\mu}$
TME	−41	a	b	Gaussian distribution of time

<u>f = −2</u> Maxwell fission energy spectrum: $p(E) = C\ E^{1/2} \exp(-E/a)$, where a is a temperature in MeV.
Default: a = 1.2895 MeV

<u>f = −3</u> Watt fission energy spectrum: $p(E) = C \exp(-E/a) \sinh(bE)^{1/2}$.
Defaults: a = 0.965 MeV, b = 2.29 MeV^{-1}. See Appendix H for additional parameters appropriate to neutron-induced fission in various materials and for spontaneous fission.

f = -4 Gaussian fusion energy spectrum: $p(E) = C \exp[-((E-b)/a)^2]$, where a is the width in MeV and b is the average energy in MeV. Width here is defined as the ΔE above b where the value of the exponential is equal to e^{-1}. If $a < 0$, it is interpreted as a temperature in MeV and b must also be negative. If $b = -1$, the D-T fusion energy is calculated and used for b. If $b = -2$, the D-D fusion energy is calculated and used for b. Note that a is not the "full-width-at-half-maximum," but is related to it by $\mathrm{FWHM} = a\,(\ln 16)^{1/2}$.
Defaults: $a = -0.01$ MeV, $b = -1$ (DT fusion at 10 keV).

f = -5 Evaporation energy spectrum: $p(E) = C\,E \exp(-E/a)$.
Default: $a = 1.2895$ MeV.

f = -6 Muir velocity Gaussian fusion energy spectrum:
$p(E) = C \exp - ((E^{1/2} - b^{1/2})/a)^2$, where a is the width in $\mathrm{MeV}^{1/2}$, and b is the energy in MeV corresponding to the average speed. Width here is defined as the change in velocity above the average velocity $b^{1/2}$, where the value of the exponential is equal to e^{-1}. To get a spectrum somewhat comparable to $f = -4$, the width can be determined by $a = (b+a_4)^{1/2}-b^{1/2}$, where a_4 is the width used with the Gaussian fusion energy spectrum. If $a < 0$, it is interpreted as a temperature in MeV. If $b = -1$, the D-T fusion energy is calculated and used for b. If $b = -2$, the D-D fusion energy is calculated and used for b.
Defaults: $a = -0.01$ MeV, $b = -1$ (DT fusion at 10 keV).

f = -7 Spare energy spectrum. The basic framework for another energy spectrum is in place to make it easier for a user to add a spectrum of his own. The subroutines to change are SPROB, SPEC, SMPSRC, and possibly CALCPS.

f = -21 Power law: $p(x) = c|x|^a$.
The default depends on the variable. For DIR, $a = 1$. For RAD, $a = 2$, unless AXS is defined or JSU $\neq 0$, in which case $a = 1$. For EXT, $a = 0$.

f = -31 Exponential: $p(\mu) = ce^{a\mu}$.
Default: $a = 0$.

f = -41 Gaussian distribution of time:
$p(t) = c \exp[-(1.6651092(t-b)/a)^2]$, where a is the width at half maximum in shakes and b is the mean in shakes.
Defaults: a=no default, b=0.

The built-in functions can be used only for the variables shown in Table 3.3. Any of the built-in functions can be used on SP cards, but only

−21 and −31 can be used on SB cards. If a function is used on an SB card, only that same function can be used on the corresponding SP card. The combination of a regular table on the SI and SP cards with a function on the SB card is not allowed.

A built-in function on an SP card may be biased or truncated or both by a table on SI and SB cards. The biasing affects only the probabilities of the bins, not the shape of the function within each bin. If it is biased, the function is approximated within each bin by n equally probable groups such that the product of n and the number of bins is as large as possible but not over 300. Unless the function is −21 or −31, the weight of the source particle is adjusted to compensate for truncation of the function by the entries on the SI card.

Several special defaults are available for distributions that use the built-in functions.

1. If SB f is present and SP f is not, an SP f with default input parameters is, in effect, provided by MCNP.

2. If only an SI card is present for DIR, RAD, or EXT, an SP f with default input parameters is, in effect, provided. f=−21 except for DIR with JSU=0 and then f=−31.

3. If only SP −21 or SP −31 is present for DIR or EXT, an SI 0 1, for −21, or SI −1 1, for −31, is, in effect, provided.

4. If SI x and SP −21 are present for RAD, the SI is treated as if it were SI 0 x.

5. If SI x and SP −21 or SP −31 are present for EXT, the SI is treated as if it were SI −x x.

5. DSn — Dependent Source Distribution Card

Form: DSn option $J_1 \ldots J_k$
or: DSn T $I_1\ J_1 \ldots I_k\ J_k$
or: DSn Q $V_1\ S_1 \ldots V_k\ S_k$

n = distribution number (n=1,999)
option = how the J_i are to be interpreted. Allowed values are:
blank or H − source variable values in a continuous distribution, for scalar variables only
L − discrete source variable values
S − distribution numbers
T = values of the dependent variable follow values of the independent variable, which must be a discrete scalar variable
I_i = values of the independent variable
J_i = values of the dependent variable
Q = distribution numbers follow values of the independent variable, which must be a scalar variable

V_i = monotonically increasing set of values of the independent variable

S_i = distribution numbers for the dependent variable

Default: DSn H $J_1 \ldots J_k$

The DS card is used instead of the SI card for a variable that depends on another source variable, as indicated on the SDEF card. No SP or SB card is used. MCNP first determines the value of the independent source variable as usual by sampling the probability function of the independent variable. Then the value of the dependent variable is determined according to the form of the DS card.

If the first form of the DS card is used, it must have the same number of bins or discrete values as the SI card of the independent distribution (n entries are required to specify n discrete values; n+1 entries are required to specify n bins). If the sampled value of the independent variable is $I_i + [f(I_{i+1} - I_i)]$, the value of the dependent variable is $J_i + [f(J_{i+1} - J_i)]$, where the terms in f are used only for continuous distributions. The interpolation factor f always exists whether or not it is needed for the independent distribution. It is not necessary for the distributions of the independent and dependent variables to be both discrete or both continuous. All combinations work correctly.

When the T option is used on a DS card, the sampled value of the independent variable is sought among the I_i, and if a match is found, the independent variable gets the value J_i. If no match is found, the dependent variable gets its default value. The purpose of the T option is to shorten the input when a dependent variable should usually get the default value.

When the Q option is used on a DS card, the V_i define a set of bins for the independent variable. The sampled value of the independent variable is compared with the V_i, starting with V_1, and if the sampled value is less than or equal to V_i, the distribution S_i is sampled for the value of the dependent variable. The value of V_k must be greater than or equal to any possible value of the independent variable. If a distribution number S_i is zero, the default value for the variable is used. The Q option is the only form of the DS card that can be used when the distribution of the independent variable is a built-in function.

6. SCn Source Comment Card

Form: SCn comment

n = distribution number (n=1,999)

The comment is printed as part of the header of distribution n in the source distribution table and in the source distribution frequency table.

Default: No comment.

Examples of the General Source

Example 1:

```
SDEF   ERG=D1   POS=x y z   WGT=w
SI1    H  E1  E2 ... Ek
SP1    D  0   P2 ... Pk
SB1    D  0   B2 ... Bk
```

This is a point isotropic source at x,y,z with a biased continuous-energy distribution and average source particle weight w. The starting cell is not specified. MCNP will determine it from the values of x, y, and z.

Example 2:

```
SDEF   SUR=m   AXS=i j k   EXT=D6
SB6    -31   1.5
```

This is a source on surface m. The presence of AXS and EXT implies that surface m is a sphere because AXS and EXT are not otherwise used together for sources on a surface. By default, the particles are emitted in a cosine distribution. They are emitted outward if the positive normal to the sphere is outward, which it is for all the spherical surface types but might not be if the sphere is specified as type SQ. The position on the surface is biased toward the direction i,j,k by an exponential bias (specified by −31) such that the maximum and minimum source particle weights are $e^{1.5} = 4.48$ and $e^{-1.5} = 0.223$. By default, MCNP provides the effect of two cards: SI6 −1 1 and SP6 −31 0.

Example 3:

```
SDEF   SUR=m   NRM=-1   DIR=D1   WGT=w
SB1    -21   2
```

This is an inward-directed source on spherical surface m, assuming the positive normal of the surface is directed outward. If $w = \pi r^2$, where r is the radius of sphere m, this source in conjunction with a VOID card, a VOL card, and type 2 and type 4 tallies, is suitable for estimating the areas of surfaces and the volumes of cells. See page 160. By default, MCNP provides the effect of two cards: SI1 0 1 and SP1 −21 1. The directional bias by the SB1 card causes higher track density toward the center of the sphere, where presumably the cells of greatest interest lie, than it would be if the unbiased cosine distribution were used. This bias,

incidentally, provides a zero-variance estimate of the (known) volume of the sphere m.

```
Example 4:      SDEF    ERG=D1  POS=x y z  CEL=m  RAD=D2
                        EXT=D3  AXS=i j k
                SP1     -3
                SI2     r1  r2
                SI3     l
```

This source is distributed uniformly in volume throughout cell m, which presumably approximates a cylinder. The cell is enclosed by a sampling volume centered at x,y,z. The axis of the sampling volume is the line through x,y,z in the direction i,j,k. The inner and outer radii of the sampling volume are r_1 and r_2, and it extends along i,j,k for a distance $\pm l$ from x,y,z. The user has to make sure that the sampling volume totally encloses cell m. The energies of the source particles are sampled from the Watt fission spectrum using the default values of the two parameters, making it a Cranberg spectrum. By default, MCNP interprets SI3 l as if it were actually SI3 $-l$ $+l$ and provides the effect of two cards: SP2 -21 1 and SP3 -21 0.

```
Example 5:      SDEF    SUR=m  POS=x y z  RAD=D1  DIR=1  CCC=n
                SI1     r
```

This is a monodirectional source emitted from surface m in the direction of the positive normal to the surface. The presence of POS and RAD implies that surface m is a plane because POS and RAD are not otherwise used together for sources on a surface. The position is sampled uniformly in area on the surface within radius r of point x,y,z. The user must make sure that point x,y,z actually lies on surface m. The sampled position is rejected and resampled if it is not within cookie-cutter cell n. The starting cell is found from the position and the direction of the particle. By default, MCNP interprets SI1 r as if it were actually SI1 0 r and provides the effect of card SP1 -21 1.

```
Example 6:      SDEF    POS D1  ERG FPOS D2
                SI1       L    5 3.3 6    75 3.3 6
                SP1      .3    .7
                DS2       S    3    4
                SI3       H    2   10   14
                SP3       D    0    1    2
                SI4      -3    a    b
```

This is a point isotropic source in two locations, shown by two x,y,z's on the SI1 card. The code will determine the starting cell. With probability .3 the first location will be picked and with probability .7 the second location will be chosen. Each location has a different energy spectrum, pointed to by the DS2 card. All other needed source variables will use their default values.

7. SSW Surface Source Write Card

Form: SSW S_1 S_2 ... S_n SYM=m PTY=n

The = signs are optional, as are the SYM=m and PTY=n entries.

S_i = problem surface number, with the appropriate sense of inward or outward particle direction, for which particle-crossing information is to be written to the surface source file WSSA.

m = 0, no symmetry assumed.

m = 1, spherical symmetry assumed. The list of problem surface numbers must contain only one surface and it must be a sphere.

n = 0 record all tracks

n = 1 record only neutron tracks

n = 2 record only photon tracks

Default: SYM=0 PTY=0

Use: Optional, as needed.

This card is used to write a surface source file for use in a subsequent MCNP calculation. Care must be taken to include enough geometry beyond the specified surfaces to account for albedo effects. If the SYM=1 option is used, fewer words per particle need to be written to the surface source file and certain biasing options become available when reading the surface source file.

8. SSR Surface Source Read Card

Form: SSR keyword=values keyword=values

The = signs are optional.

OLD S_1 S_2 ... S_n
list of problem surface numbers, a subset of the surfaces on the SSW card that created the file WSSA, now called RSSA.

NEW S_{a1} S_{a2} ... S_{an} S_{b1} S_{b2} ... S_{bn}
problem surface numbers upon which the surface source is to

start particles in this run. The n entries may be repeated to start the surface source in a,b,... transformed locations.

COL m collision option flag

m = −1 start from the surface source file only those particles that came directly from the source without a collision

m = 1 start from the surface source file only those particles that had collisions before crossing the recording surface

m = 0 start particles without regard to collisions

WGT x Each particle weight is multiplied by the constant x as it is accepted for transport.

TR n transformation number. Track positions and velocities are transformed from the auxiliary coordinate system (the coordinate system of the problem that wrote the surface source file) into the coordinate system of the current problem, using the transformation on the TRn card, which must be present in the INP file of the current problem.

TR Dn Distribution number for a set of SIn, SPn, and SBn cards. If the surface source is transformed into several locations, the SIn card lists the transformation numbers and the SPn and SBn cards give the probabilities and bias of each transformation.

The following four keywords are used only with spherically symmetric surface sources, that is, sources generated with SYM=1 on the SSW card.

AXS u v w Direction cosines that define an axis through the center of the surface sphere in the auxiliary (original) coordinate system. This is the reference vector for EXT.

EXT Dn n is the number of a distribution (SIn, SPn, and SBn cards) that will bias the sampling of the axial coordinate of position on the sphere.

POA c Particles with a polar angle cosine relative to the source surface normal that falls between 1 and c will be accepted for transport All others are disregarded and no weight adjustment is made.

BCW r zb ze $0 < zb < ze$ All particles with acceptable polar angles relative to the surface normal are started so that they will pass through a cylindrical window of radius r, starting at zb from the center of the source sphere, and ending at ze from the center. The axis of the cylinder is parallel to the z−axis of the auxiliary (original) coordinate system and contains the center of the source sphere. The weight of each source particle is adjusted to compensate for this biasing of position and direction.

Defaults:	OLD	all surfaces in the original run
	NEW	the surfaces in the OLD list.
	COL	m = 0
	WGT	1
	TR	no transformation
	AXS	no axis
	EXT	no position bias
	POA	c = 0
	BCW	no cylindrical window

Use: Required for surface source problems.

Because the collision history of particles inside the source surface is lost, the surface source does not work with next-event estimators such as DXTRAN or point and ring detectors. If either DXTRAN or detectors are used in a surface source problem, a fatal error message is issued.

The particle type is determined primarily by the type of the particle on the RSSA file, but particles incompatible with the problem mode are rejected without weight adjustment. In MODE N, photons are rejected. In MODE P, neutrons are rejected.

```
Example 1:  Original run:  SSW   1  2  3
            Current  run:  SSR   OLD 3 2  NEW 6 7 12 13  TR D5  COL 1
                           SI5   L  4  5
                           SP5   .4  .6
                           SB5   .3  .7
```

Particles starting on surface 1 in the original run will not be started in the current run because 1 is absent from the list of OLD surface numbers. Particles recorded on surface 2 in the original run will be started on surfaces 7 and 13 and particles recorded on surface 3 in the original run will be started on surfaces 6 and 12, as prescribed by the mapping from the OLD to the NEW surface numbers. The COL keyword causes only particles that crossed surfaces 2 and 3 in the original problem after having undergone collisions to be started in the current problem. The TR entry indicates that distribution function 5 describes the required surface transformations. According to the SI5 card, surfaces 6 and 7 are related to surfaces 3 and 2, respectively, by transformation TR4; surfaces 12 and 13 are related to 3 and 2 by TR5. The physical probability of starting on surfaces 6 and 7 is 40% according to the SP5 card, and the physical probability of starting on surfaces 12 and 13 is 60%. The SB5 card causes the particles from surfaces 3 and 2 to be started on surfaces 6 and 7 30% of the time with weight multiplier 4/3 and to be started on surfaces 12 and 13 70% of the time with weight multiplier 6/7.

```
Example 2:  Original run:  SSW   3   SYM 1
            Current  run:  SSR   AXS 0 0 1   EXT D99
                           SI99  -1   .5   1
                           SP99  C    .75  1
                           SB99  0    .5   .5
```

All particles written to surface 3 in the original problem will be started on surface 3 in the new problem, which must be exactly the same because no OLD, NEW, COL, or TR keywords are present. Because this is a spherically symmetric problem, indicated by the SYM 1 flag in the original run, the position on the sphere can be biased. It is biased in the z-direction with a cone bias described by distribution 99.

9. KCODE Criticality Source Card

Form: KCODE NSRCK RKK IKZ KCT MSRK KNRM

NSRCK = nominal source size per cycle
RKK = initial guess for K_{eff}
IKZ = number of cycles to be skipped before beginning tally accumulation
KCT = number of cycles to be done
MSRK = number of source points to allocate storage for
KNRM = method of tally normalization.
0 means normalize tallies by weight
not zero means normalize tallies to number of particles

Defaults: NSRCK=no default; RKK=1.0; IKZ=5; KCT=0, that is, do not terminate on the number of cycles; MSRK=4500 or 1.5*NSRCK, whichever is larger; and KNRM=0.

Use: This card is required for criticality calculations.

The KCODE card specifies the MCNP criticality source that is used for determining k_{eff}. No other source (SDEF, SSR, or user supplied) may be present with a KCODE card. The criticality source uses total fission nubar values unless overridden by a TOTNU NO card. The KCODE does not work in MODE P problems. See page 148 for further information.

The NSRCK entry is the nominal source size for each cycle and is frequently taken to be in the range of 300 < NSRCK ≤ 3000. The IKZ entry is the number of cycles to skip before beginning tally accumulation (this is important if the initial source guess is poor). The KCT entry specifies the number of cycles to be done before the problem ends. A zero entry means never terminate on the number of cycles but terminate on time. The

MSRK is the maximum number of source points for which storage will be allocated. If an SRCTP file with a larger value of MSRK is read for the initial source, the larger value is used.

If in the first cycle the source being generated overruns the current source, the initial guess (RKK) is probably too low. The code then proceeds to print a comment, continues without writing a new source, calculates k'_{eff}, reads the initial source back in, and begins the problem using k'_{eff} instead of RKK. If the generated source overruns the current source after the first cycle, the job aborts.

10. KSRC Source Points for KCODE Calculation

Form: KSRC $x_1\ y_1\ z_1 \quad x_2\ y_2\ z_2 \quad \ldots$

$x_i,\ y_i,\ z_i$ = location of initial source points

Default: None. If this card is absent, an SRCTP source file must be supplied for a criticality calculation.

Use: Optional card for use with criticality calculations.

This card contains up to NSRCK (x,y,z) triplets, which are locations of initial source points for a KCODE criticality calculation. At least one point must be in a cell containing fissile material. The points must be away from cell boundaries. It is not necessary to input all NSRCK coordinate points. MCNP will start approximately (NSRCK/number of points) particles at each point. Usually one point in each fissile region is adequate, since MCNP will quickly calculate and use the new fission source distribution. The energy of each particle in the initial source is sampled from a fission spectrum hardwired into MCNP.

An SRCTP file from a previous criticality calculation may be used instead of a KSRC card. If the current problem has a lot in common with the previous problem, using the SRCTP file may save some computer time. Even if the problems are quite different, the SRCTP file may still be usable if some of the points in the SRCTP file are in cells containing fissile material in the current problem. Points in void or zero importance cells will be deleted. The number of particles actually started at each point will be such as to produce approximately NSRCK initial source particles.

11. Subroutines SOURCE and SRCDX

If SDEF, SSR, or KCODE cards are not present in the INP file, a user supplied source is assumed and is implemented by calling subroutine SOURCE.

which the user must provide. Appendix A gives details about how to supply the SOURCE subroutine to MCNP with HISTORIAN and Chapter 4 has examples of a SOURCE subroutine. The parameters that must be specified within the subroutine are listed and defined on page 200.

Prior to calling subroutine SOURCE, isotropic direction cosines u,v,w (UUU,VVV,WWW) are calculated. Therefore, you need not specify the direction cosines if you want an isotropic distribution.

The SIn, SPn, and SBn cards can also be used with the SOURCE subroutine. A random number generator RANG() is available for use by SOURCE for generating random numbers between 0 and 1. Up to 50 numerical entries can be entered on each of the IDUM and RDUM cards for use by SOURCE. The IDUM entries must be integers and the RDUM entries floating point numbers.

If you are using a detector or DXTRAN and your source has an anisotropic angular distribution, you will also need to supply an SRCDX subroutine to specify PSCs for each detector or DXTRAN sphere (see Chapters 2 and 4).

Four spare variables, SPARE1, SPARE2, SPARE3, and ISPACE, are stored in the particle bank and are reserved for the user. Depending on the application, you may need to reset them to 0 in SOURCE for each history; MCNP does not reset them.

E. Tally Specification

The tally cards are used to specify what type of information the user wants to gain from the Monte Carlo calculation; that is, current across a surface, flux at a point, heating in a region, etc. This information is requested by the user by using a combination of the following cards. To obtain tally results, only the Fn card is required; the other tally cards provide various optional features.

Mnemonic (Columns 1-5)	Card Type	Page
Fna	Tally type	220
FCn	Tally comment	224
En	Tally energies	225
Tn	Tally times	225
Cn	Tally cosines	226
FQn	Tally print hierarchy	227
FMn	Tally multiplier	228
DEn/DFn	Dose energy / Dose function	232
EMn	Energy multiplier	233

The n is a user-chosen tally number no longer than three digits; choices of n are given in the following section. When a choice of n is made for a particular tally type, any other input card used with that tally (such as En for energy bins) is given the same value of n by the user.

Much of the information on these cards is used to describe tally "bins," subdivisions of the tally space into discrete and contiguous increments such as cosine, energy, or time. Usually when the user subdivides a tally into bins, MCNP can also provide the total tally summed over appropriate bins (such as over energy bins). Absence of any bin specification card results in one unbounded bin rather than one bin with a default bound. No information is printed about the limits on the unbounded bin.

If there are reflecting planes in the problem, the user may have to normalize the tallies in some special way (this can be done by setting the weight of the source particles or by using the FMn card). See page 44 in Chapter 2 for a more complete discussion.

Printed with each tally bin is the relative error of the tally corresponding to one standard deviation. These errors *cannot* be believed reliable (hence neither can the tally itself) unless the error is fairly low. Results with errors greater than 50% are useless, results between 20% and 50% can be believed to within a factor of a few, results between 10% and 20% are questionable, results less than 10% are generally (but not always) reliable except for detectors, and detector results are generally reliable below 5%. Results are not reliable until they become stable as a function of the number of histories run (that is, the variance of the variance is low). An indication of tally stability can be seen in the tally fluctuation charts at the end of the printed output.

1. Fna Tally Cards

Six basic neutron tally types and five basic photon tally types are available in MCNP as standard tallies. All are normalized to be per source particle unless changed by the user with a TALLYX subroutine.

Mnemonic	Tally Description	Fn units	*Fn units
F1:N or F1:P	Current integrated over a surface	particles	MeV
F2:N or F2:P	Flux averaged over a surface	particles/cm^2	MeV/cm^2
F4:N or F4:P	Flux averaged over a cell	particles/cm^2	MeV/cm^2
F5a:N or F5a:P	Flux at a point or ring detector	particles/cm^2	MeV/cm^2
F6:N or F6:N,P or F6:P	Energy deposition averaged over a cell	MeV/g	jerks/g
F7:N	Fission energy deposition averaged over a cell	MeV/g	jerks/g

The tallies are identified by tally type and particle type as follows. Tallies are given the numbers 1, 2, 4, 5, 6, 7, or increments of 10 thereof, and are given the particle designator :N or :P (or :N,P only in the case of tally type 6). Thus you may have as many of any basic tally as you need, each with different energy bins or flagging or anything else. F4:N, F14:N, F104:N, and F234:N are all legitimate neutron cell flux tallies; they could all be for the same cell(s) but with different energy or multiplier bins, for example. Similarly F5:P, F15:P, and *F305:P are all photon point detector tallies. Having both an F1:N card and an F1:P card in the same INP file is not allowed. The tally number may not exceed three digits.

Tally types 1, 2, 4, and 5 are normally weight tallies (particles in the above table); however, if the Fn card is flagged with an asterisk (for example, *F1:N), energy times weight will be tallied. The asterisk flagging can also be used on tally types 6 and 7 to change the units from MeV/g to jerks/g (1 jerk = 1 GJ = 10^9 J). All of the units are shown in the above table.

Only the F2 surface flux tally requires the surface area. The area calculated is the total area of the surface that may bound several cells, not a portion of the surface that bounds only a particular cell. If you need only the segment of a surface, you might segment the full surface with the FSn card (see page 236) and use the SDn card (see page 238) to enter the appropriate values. You can also redefine the geometry as another solution to the problem.

The detector total is restricted to 20. The tally total is limited to 100. Note below that a single type 5 tally may create more than one detector.

(a) Surface and Cell Tallies (tally types 1, 2, 4, 6, and 7)

Simple Form: Fn:pl S_1 ... S_k
General Form: Fn:pl S_1 (S_2 ... S_3) (S_4 ... S_5) S_6 S_7 ...

n = tally number.
pl = N or P or N,P
S_i = problem number of surface or cell for tallying, or T.

In the simple form above, MCNP creates k surface or cell bins for the requested tally, listing the results separately for each surface or cell. In the more general form, a bin is created for each surface or cell listed separately and for each collection of surfaces or cells enclosed within a set of parentheses. Entries within parentheses may also appear separately or in other combinations. Parentheses indicate that the tally is for the union of the items within the parentheses. For unnormalized tallies (tally type 1), the union of tallies is a sum, but for normalized tallies (types 2, 4, 6, and 7), the union results in an average.

The symbol T entered on surface or cell Fn cards is shorthand for a region which is the union of all of the other entries on the card. A tally is still made for the individual entries on the Fn card plus the union of all the entries.

If a tally label of the surfaces or cells in the output requires more than eleven characters, including spaces, MCNP defines an alphabetical or numerical designator for printing purposes. The designator [for example, G is (1 2 3 4 5 6)] is printed with the tally output. This labeling scheme is usually required for tallies over the union of a long list of surfaces or cells.

Example 1: F2:N 1 3 6 T

This card specifies four neutron flux tallies, one across each of the surfaces 1, 3, and 6 and one which is the average of the flux across all three of the surfaces.

Example 2: F1:P (1 2) (3 4 5) 6

This card provides three photon current tallies, one for the sum over surfaces 1 and 2; one for the sum over surfaces 3, 4, and 5; and one for surface 6 alone.

Example 3: F371:N (1 2 3) (1 4) T

This card provides three neutron current tallies, one for the sum over surfaces 1, 2, and 3; one for the sum over surfaces 1 and 4; and one for the sum over surfaces 1, 2, 3, and 4. The point of this example is that the T bin is not confused by the repetition of surface 1.

Another case for study is in the DEMO example in Chapter 5.

(b) Detector Tallies (tally type 5)

Form for point detectors: Fn:pl X Y Z $\pm R_o$

n = tally number.
pl = N for neutrons or P for photons,
X Y Z = location of the detector point.
$\pm R_o$ = radius of the sphere of exclusion:
in centimeters if R_o is entered as positive,
in mean free paths if entered as negative.
(Negative entry illegal in a void region.)

Form for ring detectors: Fna:pl a_o r $\pm R_o$

n = tally number.
a = the letter X, Y, or Z.
pl = N for neutrons or P for photons.
a_o = distance along axis "a" where the ring plane intersects the axis.
r = radius of the ring in centimeters.
$\pm R_o$ = same meaning as for point detectors, but describes a sphere about the point selected on the ring.

Default: R_o = 0 if missing or jumped over.

Use: You are encouraged to read about detectors, starting on page 87 in Chapter 2, before using them because they are very susceptible to unreliable results if used improperly. Remember that contributions to a detector are not made through a region of zero importance. Ring (rather than point) detectors should be used in all problems with axial symmetry. A detector located right on a surface will probably cause trouble. Detectors and DXTRAN can be used in problems with the $S(\alpha,\beta)$ thermal treatment, but no $S(\alpha,\beta)$ contributions will be made to them. Detectors used with reflecting surfaces usually give wrong answers (see page 96). Consider using the PDn and DDn cards.

For more than one detector with the same n or na designation, sets of the above input parameters are simply continued on the same Fn or Fna card. If more than one detector of the same type (an F5:N and an F15:N, for example) are at the same location, the time-consuming contribution calculation upon collision is made only once and not independently for each detector, according to the rules in Chapter 2. Thus it is inexpensive to add more than one detector (each with a different response function, for example) at the same location as another.

The printout for detectors is normally in two parts: (1) the total of all contributions to the detector (as a function of any defined bins such as energy) and (2) the direct (or uncollided) contribution to the detector from the source. The direct contribution is always included in the total of all contributions. Adding the symbol ND at the end of a type 5 detector tally card inhibits the separate printing of the direct contribution for that tally. In coupled neutron/photon problems, the direct contribution in photon tallies is from photons created at neutron collisions.

Rules of Thumb for R_o: R_o should be about 1/8 to 1/2 mean free path for particles of average energy at the sphere and zero in a void. Supplying R_o in terms of mean free path will increase the variance and is not recommended unless you have no idea how to specify it in centimeters. R_o must not encompass more than one material. MCNP cannot check this and the consequences may be disastrous.

2. FCn Tally Comment Card

Form: FCn any desired information...

Default: No comment.

Use: Especially recommended for nonstandard tallies.

Anything entered in columns 6–72 on this card will appear as the title heading of tally Fn. This card is particularly useful when tallies are modified in some way, so later readers of the output will be warned of modified or nonstandard tallies. The comment on this card may be continued for as many lines as you like simply by following the FCn card with additional cards with the continued comment in columns 6–72 (that is, columns 1–5 are blank). A blank continuation tally comment card is not allowed; MCNP will interpret that as the end of the input file and ignore any following cards.

3. En Tally Energy Card

Form: En E_1 ... E_k

n = tally number.
E_i = upper bound (in MeV) of the i^{th} energy bin for tally n.

Default: If the En card is absent, there will be one bin over all energies unless this default has been changed by an E0 card.

Use: Required if EMn card is used.

The entries on the En card must be entered in the order of increasing magnitude. If a particle has an energy greater than the last entry, it will not be tallied, but you will be warned that this has happened. If the last entry is greater than the upper energy limit E_{max} specified on the PHYS card, the last bin will be lowered to E_{max}. If there are several bins above E_{max}, the extra bins are eliminated.

An E0 (zero) card may be used to set up a default energy bin structure for all tallies. A specific En card will override the default structure for tally n.

MCNP automatically provides the total over all specified energy bins, but the total can be inhibited for a given tally by putting the symbol NT at the end of the En card for that tally.

Example: E11 .1 1 20

This will separate an F11 current tally into four energy bins: (1) from the energy cutoff to 0.1 MeV, (2) from 0.1 to 1.0 MeV, (3) from 1.0 to 20.0 MeV, and (4) a total over all energy.

4. Tn Tally Time Card

Form: Tn T_1 ... T_k

n = tally number.
T_i = upper bound (in shakes) of the i^{th} time bin for tally n.

Default: If the Tn card is absent, there will be one bin over all times unless this default has been changed by a T0 card.

Use: Required if TMn card is used.

The times on the Tn card must be entered in order of increasing magnitude. If a particle has a time greater than the last entry on the Tn card, it will not be tallied, but you will be warned that this has happened. The last time bin entry should always be less than or equal to the time cutoff (see page 257). If time bins greater than the time cutoff are entered, the first bin limit over the cutoff will be lowered to the cutoff. All higher bins will be eliminated.

A T0 (zero) card may be used to set up a default time bin structure for all tallies. A specific Tn card will override the default structure for tally n.

MCNP automatically provides the total over all specified time bins, but the total can be inhibited for a given tally by putting the symbol NT at the end of the Tn card for that tally.

Example: T2 -1 1 1.0+37 NT

This will separate an F2 flux surface tally into three time bins: (1) from $-\infty$ to -1.0 shake, (2) from -1.0 shake to 1.0 shake, and (3) from 1.0 shake to 1.0e37 shakes, effectively infinity. No total bin will be printed in this example.

5. Cn Cosine Card (tally type 1 only)

Form: Cn $C_1 \ldots C_k$

n = tally number.
C_i = upper cosine limit of the i^{th} angular bin for surface current tally n. $C_1 > -1$. $C_k = 1$.

Default: If the Cn card is absent, there will be one bin over all angles unless this default has been changed by a C0 card.

Use: Tally type 1. Required if CMn card is used. Consider FQn card.

The angular limits described by the Cn card are defined with respect to the positive normal to the surface at the particle point of entry. The positive normal to the surface is always in the direction of a cell that has positive sense with respect to that surface. The cosines must be entered in increasing order, beginning with the cosine of the largest angle less than 180° to the normal and ending with the normal (cos=1). A lower bound of -1 is set in the code and should not be entered on the card. The last entry must always be 1.

A C0 (zero) card may be used to set up a default angular bin structure for all tallies. A specific Cn card will override the default structure

for tally n. Note that the selection of a single cosine bin for an F1 tally gives the total and not the net current crossing a surface.

MCNP does not automatically provide the total over all specified cosine bins, but the total can be generated for a given tally by putting the symbol T at the end of the Cn card for that tally.

Example: C1 −.866 −.5 0 .5 .866 1

This will tally currents within the angular limits (1) 180° to 150°, (2) 150° to 120°, (3) 120° to 90°, (4) 90° to 60°, (5) 60° to 30°, and (6) 30° to 0° with respect to the positive normal. No total will be provided.

As an example of the relation between a surface normal and sense for the C1 card, consider a source at the origin of a coordinate system and a plane (PY) intersecting the +y axis. An entry of 0 and 1 on the C1 card will tally all source particles transmitted through the plane in the 0 to 1 cosine bin (0° to 90°) and all particles reflected back across the plane in the −1 to 0 cosine bin (90° to 180°). A plane (PY) intersecting the −y axis will result in a tally of all source particles transmitted through the second plane in the −1 to 0 bin (90° to 180°) and all particles reflected back across the plane in the 0 to 1 bin (0° to 90°). Note that the positive normal direction for both planes is the same, the +y axis.

6. FQn Print Hierarchy Card

Form: FQn a_1 a_2 ... a_8

n = tally number
a_i = F − cell, surface, or detector
D − direct or flagged
U − user
S − segment
M − multiplier
C − cosine
E − energy
T − time

Default: Order as given above.

Use: Recommended where appropriate.

The a_i's are the letters representing all eight possible types of tally bins. This card can be used to change the order in which the output is printed for the tallies. For a given tally, the default order shown

above is changed by entering a different ordering of the letters, which are space delimited.

A subset of the letters can be used, in which case MCNP places them at the end of the FQn card and precedes them with the unspecified letters in the default order. The first letter is for the outermost loop of the nest in the tally printout coding. The last two sets of bins make a table: the next to last set goes vertically; the last set of bins goes horizontally in the table.

Note that by the default order you will get a table in E and T; any other bins in a tally will just be listed vertically down the output page. Thus if you have a tally with only cell, user, and energy bins, the print for that tally will be a hard-to-read vertical list. Specifying U E as the only entries or last two entries on the FQn card will produce the same output, but in an easy-to-read table.

An FQ0 (zero) card may be used to change the default order for all tallies. A specific FQn card will then override that order for tally number n.

An example of this card is in the DEMO example in Chapter 5.

7. FMn Tally Multiplier Card

Form: FMn (bin set 1) (bin set 2) ... T

n = tally number
(bin set i) = ((multiplier set 1) (multiplier set 2) ... (attenuator set))
T = absent for no total over bins
= present for total over all bins

attenuator set = C -1 m_1 px_1 m_2 px_2 ...
multiplier set i = C m (reaction list 1) (reaction list 2) ...

C = multiplicative constant
-1 = flag indicating attenuator rather than multiplier set
m = material number identified on an Mm card
px = density times thickness of attenuating material; atom density if positive, mass density if negative
(reaction list i) = sums and products of ENDF or special reaction numbers, described below.

Parentheses:

(a) If a given multiplier set contains only one reaction list, the parentheses surrounding the reaction list may be omitted. Parentheses within a reaction list are forbidden.

(b) If a given bin set consists of more than a single multiplier or attenuator set, each multiplier or attenuator set must be surrounded by parentheses, and the combination must also be surrounded by parentheses.

(c) If the FMn card consists only of a single bin set, and that bin set consists only of a single multiplier or attenuator bin, surrounding parentheses may be omitted.

Default: If the C entry is negative (for type 4 tally only), C is replaced by |C| times the atom density of the cell where the tally is made.

Use: Optional. Use the attenuators only when they are thin. Use only the multiplicative constant for tally types 6 and 7.

The FMn card is used to calculate any quantity of the form $C \int \varphi(E)\, R_m(E)\, dE$, where $\varphi(E)$ is the energy-dependent fluence (particles/cm^2) and R(E) is an operator of additive and/or multiplicative response functions from the cross-section libraries or specially designated quantities. The constant C is any arbitrary scalar quantity that can be used for normalization. The material number m must appear on an Mm card, but need not be used in a geometrical cell of the problem.

A reaction list consists of one or more reaction numbers delimited by spaces and/or colons. A space between reaction numbers means multiply the reactions. A colon means add the reactions. The hierarchy of operation is multiply first and then add. One bin is created for each reaction list. Thus, if R_1, R_2, and R_3 are three reaction numbers, the form $R_1\ R_2 : R_3$ represents one reaction list (one bin) calling for reaction R_3 added to the product of reactions R_1 and R_2. No parentheses are allowed within the reaction list. The product of R_1 with the sum of R_2 and R_3 would be represented by the form $R_1\ R_2 : R_1\ R_3$ rather than by the form $R_1\ (R_2 : R_3)$. The latter form would produce two bins with quite a different meaning (see Examples 1 and 2 below).

The reaction cross sections are *microscopic* (with units of barns) and not macroscopic. Thus, if the constant C is the atomic density (in atoms per barn·cm), the results will include the normalization "per cm^3." The examples in Chapter 4 will illustrate the normalization.

Ary number of ENDF or special reactions can be used in a multiplier set as long as they are present in the MCNP cross-section libraries, or in special libraries of dosimetry data. If neither a material nor any reactions are given, the tally is multiplied by the constant C. A special case allows C = 1 and m = anything (not blank). In this case the tally becomes the number of tracks, regardless of weight.

In addition to the approximately one hundred standard ENDF reaction numbers available (for example, R = 1, 2, 16, representing σ_{tot}, σ_{el}, $\sigma_{n,2n}$), the following nonstandard special R numbers may be used:

Neutrons:
- −2 absorption cross section
- −4 average heating number (MeV/collision)
- −5 gamma-ray production cross section, barns
- −6 total fission cross section
- −7 fission ν
- −8 fission Q (MeV/fission)

Photons:
- −1 incoherent scattering cross section
- −2 coherent scattering cross section
- −3 photoelectric cross section
- −4 pair production cross section
- −5 total cross section
- −6 photon heating number

A list of many of the ENDF reaction numbers can be found in Appendix G. The total and elastic cross sections, R=1 and R=2, are adjusted for temperature dependence. All other reactions are interpolated directly from the library data tables. Note that for tritium production, the R number differs from one nuclide to another. Note also that tally types 6 and 7 already include reactions, so the FMn card makes little sense for n = 6 or 7. Only the constant-multiplier feature should be used for these tally types, generally. Photon production reactions can be specified according to the MTRP prescription in Table F.6.

An attenuator set of the form C −1 m px, where m is the material number and px is the product of density and thickness, allows the tally to be modified by the factor $e^{-\sigma_{tot}px}$ representing an exponential line-of-sight attenuator. This capability makes it possible to have attenuators without actually modeling them in the problem geometry.

<u>Caution</u>: The assumption is made that the attenuator is thin, so that simple exponential attenuation without buildup from scattering is valid.

The attenuator set may include more than one layer:

$$C \quad -1 \quad m_1 \quad px_1 \quad m_2 \quad px_2$$

in which case the factor is $e^{-\sigma_1 px_1 - \sigma_2 px_2}$. The attenuator set may also be part of a bin set, for example,

$$((C_1 \quad m_1 \quad R_1)\ (C_2\ m_2\ R_2)\ (C_3\ -1\ m_3\ px_3))$$

in which case the attenuation factor is applied to every bin created by the multiplier sets. Note that both the inner and the outer parentheses are required for this application.

Tallies are posted in all multiplier bins for each score. MCNP does not automatically provide the total over all specified multiplier bins for a particular tally. The total is available for a tally, however, by putting the symbol T at the end of the FMn card for that tally.

Example 1: FMn C m R_1 R_2 : R_1 R_3

Example 2: FMn C m R_1 (R_2 : R_3)

These two examples reiterate that parentheses cannot be used for algebraic hierarchy within a reaction list. The first example produces a single bin with the product of reaction R_1 with the sum of reactions R_2 and R_3. The second case creates two bins, the first of which is reaction R_1 alone; the second is the sum of R_2 and R_3, without reference to R_1.

```
Example 3:      F2:N    1     2     3     4
                FM2    (C1)  (C2)  (C3)  (C4)  T

Example 4:      F12:N   1     2     3     4
                FM22    C1

Example 5:      F22:N   (1 2 3)  4  T
                FM42   (C1)  (C2)  (C3)  (C4)
```

These three examples illustrate the syntax when only the constant-multiplier feature is used. All parentheses are required in these examples. Tally 2 creates 20 bins: the flux across each of surfaces 1, 2, 3, and 4 with each multiplied by each constant C_1, C_2, C_3, C_4, and the sum of the four constants. Tally 12 creates 4 bins: the flux across each of surfaces 1, 2, 3, and 4 with each multiplied by the constant C_1. Tally 22 creates 12 bins: the flux across surface 1 plus surface 2 plus surface 3, the flux across surface 4, and the flux across all four surfaces with each multiplied by each constant C_1, C_2, C_3, and C_4. An FQn card with an entry of F M or M F would print these bins of the tallies in an easy-to-read table rather than strung out vertically down the output page.

Several more examples of the FMn card are in Chapter 4. The DEMO example in Chapter 5 also illustrates the general form of the card.

Using MCNP tallies, there are two ways to obtain the energy deposited in a material in terms of rads (1 rad = 100 ergs/g. When the actual material of interest is present in the MCNP model, the simplest way is to use the heating tally with units MeV/g in conjunction with C=1.602E-05 on the companion FMn card, where (C=1.602E-03 ergs/MeV)/(100 ergs/g). When the material is not present in the model, rads can be obtained from type 1, 2, and 4 tallies by using an FMn card where C is equal to the factor above times $N_o\eta \times 10^{-24}/A$, where N_o is Avogadro's number and η and A are the number of atoms/molecule and the atomic weight, respectively, of the material of interest. The implicit assumption when the material is not present is that it does not affect the radiation transport significantly.

8. DEn Dose Energy Card
DFn Dose Function Card

Form: DEn A E_1 ... E_k
DFn B F_1 ... F_k

n = tally number.
E_i = an energy (in MeV).
F_i = the corresponding value of the dose function.
A = LOG or LIN interpolation method for energy table.
B = LOG or LIN interpolation method for dose function table.

Defaults: If A or B is missing, LOG is chosen for that table.

Use: Tally comment recommended.

This feature allows you to enter a pointwise response function (such as flux-to-dose conversion factors) as a function of energy to modify a regular tally. Both cards must have the same number of numerical entries, and they must be monotonically increasing in energy. Particle energies outside the energy range defined on these cards use either the highest or lowest value.

By default MCNP uses log-log interpolation between the points rather than a histogram function as is done for the EMn card. The energy points specified on the DEn card do not have to equal the tally energy bins specified with the En card for the Fn tally. Unlike EMn card use, there can be many points on the DEn and DFn cards, but the response can be tallied in only a few energy bins such as one unbounded energy bin.

If n is zero on these two cards, the function will be applied to all tallies that do not have DEn and DFn cards specifically associated with them.

LIN or LOG may be chosen independently for either table. Thus any combination of interpolation (log-log, linear-linear, linear-log, or log-linear) is possible. The default log-log interpolation is appropriate for the ANSI/ANS flux-to-dose rate conversion factors (they are listed in Appendix H); kermas for air, water, and tissue; and energy absorption coefficients.

Example: DE5 E_1 E_2 E_3 E_4 .... E_k
DF5 LIN F_1 F_2 F_3 F_4 .... F_k

This example will cause a point detector tally to be modified according to the dose function F(E) using logarithmic interpolation on the energy table and linear interpolation on the dose function table.

9. EMn Energy Multiplier Card

Form: EMn M_1 . . . M_k

n = tally number.
M_i = multiplier to be applied to the i^{th} energy bin.

Default: None.

Use: Requires En card. Tally comment recommended.

This card can be used with any tally (specified by n) to scale the usual current, flux, etc. by a response function. There should be one entry for each energy entry on the corresponding En card. When a tally is being recorded within a certain energy bin, the regular contribution is multiplied by the entry on the EMn card corresponding to that bin. For example, a dose rate can be tallied with the appropriate response function entries. Tallies can also be changed to be per unit energy if the entries are $1/\Delta E$ for each bin. Note that this card modifies the tally by an energy-dependent function that has the form of a histogram and not a continuous function. It also requires the tally to have as many energy bins as there are histograms on the EMn card. If either of these two effects is not desired, see the DEn and DFn cards.

A set of energy multipliers can be specified on an EM0 (zero) card that will be used for all tallies for which there is not a specific EMn card.

10. TMn Time Multiplier Card

Form: TMn M_1 . . . M_k

n = tally number.
M_i = multiplier to be applied to the i^{th} time bin.

Default: None.

Use: Requires Tn card. Tally comment recommended.

This card is just like the EMn card except that the entries multiply time bins rather than energy bins. The Tn and TMn cards must have the same number of entries. Note that this card modifies the tally by a time-dependent function that has the form of a histogram and not a continuous function.

A set of time multipliers can be specified on a TM0 (zero) card that will be used for all tallies for which there is not a specific TMn card.

For example, if the entries are $1/\Delta T$, where ΔT is the width of the corresponding time bin, the tally will be changed to be per unit time with the units of $1/\Delta T$.

11. CMn Cosine Multiplier Card (tally type 1 only)

Form: CMn $M_1 \ldots M_k$

n = tally number.
M_i = multiplier to be applied to the i^{th} cosine bin.

Default: None.

Use: Tally type 1. Requires Cn card. Tally comment recommended.

This card is just like the EMn and TMn cards except that the entries multiply cosine bins. The number of entries on the CMn card must be the same as on the Cn card. Note that this card modifies the tally by an angular-dependent function that has the form of a histogram and not a continuous function.

A set of cosine multipliers can be specified on a CM0 (zero) card that will be used for all tallies for which there is not a specific CMn card.

For example, if you want the directionally dependent F1 tally results to be per steradian, the i^{th} entry on the CM1 card is

$$\frac{1}{2\pi(\cos\theta_i - \cos\theta_{i-1})}$$

where θ_o is 180°.

12. CFn Cell-Flagging Card (tally types 1, 2, 4, 6, 7)

Form: CFn $C_1 \ldots C_k$

n = tally number.
C_i = problem cell numbers whose tally contributions are to be flagged.

Default: None.

Use: Not with detectors. Consider FQn card.

Particle tracks may be "flagged" when they leave designated cells, and the contributions of these flagged tracks to a tally are listed separately in addition to the normal total tally. This is a method of determining the tally contribution from tracks that have passed through a certain part of the geometry.

Cell flagging cannot be used for detector tallies. The same purpose can be accomplished with a TALLYX subroutine.

The cell flag is turned on only upon leaving a cell. A source particle born in a flagged cell does not turn the flag on until it leaves the cell.

In MODE N P the flagged neutron tallies are those caused by neutrons passing through the flagged cell, but the flagged photon tallies can be caused by either a photon passing through a flagged cell or a neutron passing through a flagged cell and then leading to a photon which is tallied.

```
Example:   F4:N  6    10    13
           CF4    3    4
```

In this example the flag is turned on when a neutron leaves cell 3 or 4. The print of Tally 4 is doubled. The first print is the total track length tally in cells 6, 10, and 13. The second print is the tally in these cells for only those neutrons that have passed through cell 3 or 4 at some time before making their contribution to the cell 6, 10, or 13 tally.

13. SFn Surface-Flagging Card (tally types 1, 2, 4, 6, 7)

Form: SFn $S_1 \ldots S_k$

n = tally number.
S_i = problem surface numbers whose tally contributions are to be flagged.

Default: None.

Use: Not with detectors. Consider FQn card.

This feature is identical to cell flagging except that particles turn the flag on when they cross the specified surfaces. Thus a second tally print is given for only those particles that have crossed one or more of the surfaces specified on the SFn card.

Surface flagging cannot be used for detector tallies but a TALLYX subroutine can be written to do the same thing.

The situation for photon tallies in MODE N P is like that for the CFn card: a photon can be flagged either because it has passed through a flagged surface or because it was created by a neutron that passed through a flagged surface.

Both a CFn and an SFn card can be used for the same tally. The tally is flagged if the track passes through one or more of the specified cells or crosses one or more of the surfaces. Only one flagged output for a tally is produced from the combined CFn and SFn card use.

14. FSn Tally Segment Card (tally types 1, 2, 4, 6, 7)

Form: FSn S_1 ... S_k

n = tally number.
S_i = signed problem number of a segmenting surface.

Default: No segmenting.

Use: Not with detectors. May require SDn card. Consider FQn card.

This card allows you to subdivide a cell or a surface into segments for tallying purposes, the advantage being that it is then not necessary to specify the problem geometry with extra cells just for tallying. The segmenting surfaces specified on the FSn card are listed with the regular problem surfaces, but they need not be part of the actual geometry and hence do not complicate the cell/surface relationships.

If k surfaces are entered on the FSn card, k+1 surface or volume segments are created. Tally n is subdivided into k+1 segment bins according to the order and sense of the segmenting surfaces listed on the FSn card. If the symbol T is on the FSn card, there will be an additional total bin. Segmenting is done according to the following scheme:

Fn:N S (or C)
FSn S_1 ... S_k T (optional)

Tally n over surface S (or in cell C) will be subdivided into the following bins:

(1) the portion with the same sense with respect to surface S_1 as the sign given to S_1.

(2) the portion with the same sense with respect to surface S_2 as the sign given to S_2 but excluding that already scored in another segment, . . .

(k) the portion with the same sense with respect to surface S_k as the sign given to S_k but excluding that already scored in another segment.

(k+1) everything else,
(k+2) entire surface or cell if T is present on FSn card.

If the symbol T is absent from the FSn card, the $(k+2)^{th}$ bin is missing and MCNP calculates the tally only for each segment (including the "everything else" segment). If multiple entries are on the Fn card, each cell or surface in the tally is segmented according to the above rules. For tally types 1 or 2, the segmenting surfaces divide a problem surface into segments for the current or flux tallies. The segmenting surfaces divide a problem cell into segments for tally types 4, 6, or 7. For normalized tallies, the segment areas (for type 2), volumes (for type 4), or masses (for types 6 and 7) may have to be provided. See the discussion under the SDn card.

```
Example 1:        F2:N    1
                  FS2     -3  -4
```

This example subdivides surface 1 into three sections and calculates the neutron flux across each of them. There are three prints for the F2 tally: (1) the flux across that part of surface 1 that has negative sense with respect to surface 3, (2) the flux across that part of surface 1 that has negative sense with respect to surface 4 but that has not already been scored (and so must have positive sense with respect to surface 3), (3) everything else (that is, the flux across surface 1 with positive sense with respect to both surfaces 3 and 4).

```
Example 2:        F2:N    1
                  FS2     -3  4
```

The order and sense of the surfaces on the FS2 card are important. This example produces the same numbers as does Example 1 but changes the order of the printed flux. Bins two and three are interchanged.

It is possible to get a zero score in some tally segments if the segmenting surfaces and their senses are not properly specified. In Example 1 above, if all tallies that are positive with respect to surface 3 are also all positive with respect to surface 4, the second segment bin will have no scores.

```
Example 3:        F1:N    1  2  T
                  FS1     -3  T
```

This example produces three current tallies: (1) across surface 1, (2) across surface 2, and (3) the sum across surfaces 1 and 2. Each tally will be subdivided into three parts: (1) that with a negative sense with

respect to surface 3, (2) that with a positive sense with respect to surface 3, and (3) a total independent of surface 3.

Several additional examples of the FSn card are in Chapter 4.

15. SDn Segment Divisor Card (tally types 1, 2, 4, 6, 7)

Form: SDn $(D_{11}\ D_{12}\ \ldots\ D_{1m})\ (D_{21}\ D_{22}\ \ldots\ D_{2m})\ \ldots\ (D_{k1}\ D_{k2}\ \ldots\ D_{km})$

n = tally number.
k = number of cells or surfaces on Fn card, including T if present.
m = number of segmenting bins on the FSn card, including the remainder segment, and the total segment if FSn has a T.
D_{ij} = area, volume, or mass of j_{th} segment of the i_{th} surface or cell for tally n.
The parentheses are optional.

Defaults:

A. For cell or surface without segmenting (tally types 2, 4, 6, and 7):
1) nonzero entry on SDn card, 2) nonzero entry on VOL or AREA card, 3) volume or area calculated by MCNP, 4) fatal error

B. For cell or surface with segmenting (tally types 2, 4, 6, and 7):
1) nonzero entry on SDn card, 2) volume or area calculated by MCNP 3) fatal error

C. For surface in a type 1 tally:
1) nonzero entry on SDn card, 2) no divisor.

Use: Not with detectors. May be required with FSn card. Can be used without FSn card.

For segmented cell volumes or surface areas defined by the FSn card that are not automatically calculated by MCNP, the user can provide volumes, areas, or masses on this segment divisor card to be used by tally n. This card is similar to the VOL and AREA cards but is used for specific tallies, whereas the other two are used for the entire problem geometry. For tally type 2 the entry is area, for tally type 4 the entry is volume, and for tally types 6 and 7 the entries are masses. Tally type 1 (the current tally) is not normally divided by anything, but with the SD1 card the user can introduce any desired divisor, for example, area to tally surface current density.

Example F4:N 1 2 3 T
SD4 1 1 1 1

Note that the SDn card may be used to define tally divisors even if the tally is not segmented. In this example the tally calculates the flux in the three cells plus the union of the three cells. The VOL card can be used to set the volume divisor of the three cells (to unity, for example), but it cannot do anything about the divisor for the union. Its divisor is the sum of the volumes (whether MCNP-calculated or user-entered) of the three cells. But the divisors for all four of the cell bins may be set to unity by means of the SDn card. This overrides entries on the VOL and AREA cards.

16. FUn TALLYX Input Card

Form: FUn X_1 X_2 ... X_k

n = tally number.
X_i = input parameter establishing user bin i.

Default: If the FU card is absent, subroutine TALLYX is not called.

Use: Used only with a user-supplied TALLYX subroutine.

This card is used with a user-supplied tally modification subroutine TALLYX. The k entries on the FUn card serve three purposes: (1) each entry establishes a separate user tally bin for tally n, (2) each entry can be used as an input parameter for TALLYX to define the user bin it establishes, and (3) the entries appear in the output as labels for the user bins. IPTAL(LIPT+3,1,ITAL) is the pointer to the location in the TDS array of the word preceding the location of the data entries from the FUn card. Thus if the FUn card has the form shown above,

TDS(L+1) = X_1
TDS(L+2) = X_2
.
.
TDS(L+k) = X_k

where L = IPTAL(LIPT+3,1,ITAL)
k = IPTAL(LIPT+3,4,ITAL) − 1
= IPTAL(LIPT+3,3,ITAL) − 1
n = JPTAL(LJPT+1,ITAL)
ITAL = program number of the tally

MCNP automatically provides the total over all specified user bins. The total can be inhibited for a tally, however, by putting the symbol NT at the end of the FUn card for that tally. Then the card has the form

FUn X_1 X_2 ... X_k NT

and there is one change in the preceding list of variables:

k = IPTAL(LIPT+3,4,ITAL) − 1
  = IPTAL(LIPT+3,3,ITAL)

The discussion of the IPTAL and JPTAL arrays on page 475 in Appendix E and the following description of TALLYX may be useful.

SUBROUTINE TALLYX *User−supplied Subroutine*

Use: Called for tally n only if an FUn card is in the INP file.

TALLYX is called whenever a tally with an associated FUn card is scored. The locations of the calls to TALLYX are such that TALLYX is the very last thing to modify a score before it is posted in the tally. TALLYX calls can be initiated by more than one FUn card for different values of n; a branch must be constructed inside the subroutine based on which tally Fn is calling TALLYX, where n = JPTAL(LJPT+1,ITAL). TALLYX has the following form:

```
      SUBROUTINE TALLYX(T,IB)
*CALL CM
*CALL RC
      User-supplied FORTRAN statements
      RETURN
      END
```

The quantity T (first argument of TALLYX) that is scored in a standard tally may be multiplied or replaced by anything. The modified score T is then put into one of the k user bins established by the FUn card. In TALLYX(T,IB) the second argument IB is defined to allow for more than one pass through TALLYX per tally score. By default, IB=0, which means make one pass through the MCNP coding where user bin tally scores are posted. If the user sets IB<0 in TALLYX, no score will be made. If the user sets IB>0, passes through the user bin loop including TALLYX will be made until IB is reset to zero. This scheme allows for tally modification and posting in more than one user bin. The variable IBU is the variable designating the particular user bin established by the FUn card. Its value is 1 before

the first pass through the user bin loop. The indices of the current user, segment, cosine, energy, and time bins (IBU, IBS, IBC, IBE, and IBT, respectively) and the flag JBD that indicates flagged- or direct-versus-not are in Common for optional modification by TALLYX. Note that the index of the multiplier bin is not available and cannot be modified. NTX is a variable in blank Common. It is set equal to NX just before the CALL TALLYX in TALLYD and TALLY. The variable NX is set to unity just before the start of the user bins loop and is incremented after the CALL TALLYX, so NTX contains the number of the TALLYX call. An example of using NTX to tally in every user bin before leaving the user bin loop follows:

```
      SUBROUTINE TALLYX(T,IB)
*CALL CM
*CALL RC
      T = whatever
      IBU = NTX
      IB = 1
      IF(NTX.GE.IPTAL(LIPT+3,4,ITAL)-1) IB = 0
      RETURN
      END
```

If IBU is out of range, no score is made and a count of out-of-range scores is incremented. If excessive loops through TALLYX are made, MCNP assumes IB has been incorrectly set and terminates the job with a FATAL error (excessive is greater than the product of the numbers of bins of all kinds in the tally). Several examples of the FUn card and TALLYX are in Chapter 4. The procedure for implementing a TALLYX subroutine is the same as for the user-provided SOURCE subroutine outlined in Appendix A on page 421.

17. TFn Tally Fluctuation Card

Form: TFn $I_1 \ldots I_8$

n = tally number.
I_i = bin number for bin type i. $1 \le I_i \le$ last
last = IPTAL(LIPT+i,3,ITAL)
= total number of bins in one of the eight bin types.

Default: 1 1 last last 1 last last last

Use: Whenever one or more tally bins are more important than the default bin. Particularly useful in conjunction with the weight window generator.

At the end of the output, one chart for each tally is printed to give an indication of tally fluctuations; that is, how well the tally has converged. The tally mean, relative error, and figure of merit (FOM = $1/(\sigma^2 t)$, where σ is the relative error printed with the tally and t is computer time in minutes) are printed as functions of the number of histories run. The FOM should be roughly constant. The TF card determines for which bin in tally n the fluctuations are printed. It also determines which tally bin is optimized by the weight window generator, page 195.

Unless otherwise specified on a TFn card (n cannot be zero), the tally fluctuation chart for tally n is for the tally bin described by the above default, which means specifically

1. first cell, surface, or detector on Fn card
2. total rather than flagged or uncollided flux
3. last user bin
4. last segment bin
5. first multiplier bin on FMn card
6. last cosine bin
7. last energy bin
8. last time bin.

The TFn card allows you to change the above definition for a given tally and specify for which tally bin the chart will be printed. The eight entries on the card correspond (in order) to the above list of bin indices for the eight dimensions of the tally bins array. The order is fixed and not affected by an FQn card.

The mean printed in a chart will correspond to some number in the regular tally print. If you have more than one surface listed on an F2 card, for example, the chart will be for the first surface only; charts can be obtained for all surfaces by having a separate tally for each surface.

You may find the J feature useful to jump over *last* entries. Remember that totals are calculated for energy, time, and user bins (unless inhibited by using NT), so that *last* for eight energy bins is 9. If one segmenting surface divides a cell or surface into two segments, *last* in that case is 2, unless T is used on the FS card, in which case *last* is 3. If there are no user bins or cosine bins, for example, *last* is 1 for each; *last* is never less than 1.

Example: Suppose an F2 tally has four surface entries, is segmented into two segments (the segment plus everything else) by one segmenting surface, and has eight energy bins. By default one chart will be produced for the first surface listed, for the whole surface (and not a segment), and totaled over energy. If we wish a chart for the fifth energy bin of the third surface in the first segment, we would use TF2 3 2J 1 2J 5.

18. DDn Detector Diagnostics Card

Form: DDn k_1 m_1 k_2 m_2 ...

n = blank or zero for overall default values
= 1 for neutron DXTRAN spheres
= 2 for photon DXTRAN spheres
= tally number for specific detector tally
k_i = criterion for playing Russian roulette
m_i = criterion for printing large contributions

Defaults: If k_i is not specified on a DDn card, k_i on the DD card is used. If that is not specified, k_1 on the DD card is used. If that is not specified, k_i = 0.1 is used. A similar sequence of defaults is followed for m_i, with the final default being m_i = 1000.

Use: Optional. Remember that Russian roulette will be played for detectors and DXTRAN unless specifically turned off by use of the DD card. Consider also using the PDn or DXC cards.

This card (1) using a Russian roulette game, can speed up calculations significantly by limiting small contributions that are less than some fraction k of the average contribution per history to detectors or DXTRAN spheres, and (2) can provide more information about the origin of large contributions or the lack of a sufficient number of collisions close to the detector or DXTRAN sphere. The information provided about large contributions may be useful for setting cell importances or source-biasing parameters.

For a given detector or DXTRAN sphere, the Russian roulette criterion works as follows:

1. If k_i is positive, all contributions to the detector or sphere are made for the first 200 histories. Then the average contribution per history is computed (and will be continually updated throughout the problem). Thereafter, any contribution to the detector or sphere larger than k_i times this average contribution will always be made, but any contribution smaller than k_i times the average will be subject to the Russian roulette game. (k_i is not allowed to be greater than 1.)

2. If k_i is negative, contributions larger than $|k_i|$ will always be made, and contributions smaller than $|k_i|$ will be subject to Russian roulette. This applies to all histories from the beginning of the problem, and the 200th history has no significance.

3. If k_i is zero, no Russian roulette game will be played for the detector or sphere.

Probably, k = 0.5 is suitable for most problems; the nonzero default value 0.1 means that the game is *always played unless explicitly turned off by the user.*

The second entry, m_i, determines the condition for printing diagnostics for large contributions. If the entry is zero, there is no diagnostic print. If the entry is positive, two possibilities exist.

(1) If the corresponding k_i is positive or zero, no diagnostic prints will be made for the first 200 histories. Thereafter, the first 100 contributions larger than m_i times the (continually updated) average tally per history will be printed.

(2) If the corresponding k_i is negative, the first 100 contributions larger than m_i times $|k_i|$ will be printed.

Remember that when k_i is positive the Russian roulette game is played on the basis of the estimated average contribution per history. Because the estimate improves with each history, the game is based on a different value for each history. This can make debugging a problem more complicated, and the variance estimate does not quite obey the Central Limit Theorem. A procedure worth considering is to determine the average contribution per history in a preliminary run and then to use the negative of this value in subsequent longer runs. The Russian roulette game is played without regard to particle time or energy; thus time and energy bins for which the ultimate tally is small may lose a disproportionate share of scores by the roulette game.

The DD card eliminates <u>tracks</u> with DXTRAN but only <u>contributions</u> with detectors.

Example:	DXT:N	x_1	y_1	z_1	RI_1	RO_1	
		x_2	y_2	z_2	RI_2	RO_2	
		x_3	y_3	z_3	RI_3	RO_3	
	DXT:P	x_4	y_4	z_4	RI_4	RO_4	
	F15X:P	a_1	r_1	R_1			
		a_2	r_2	R_2			
	DD	.2	100	.15	2000		
	DD1	-1.1E25	3000	J	J	J	3000
	DD15	.4	10				

Detector/sphere	k	m
sphere 1	-1.1E25	3000
sphere 2	.15	2000
sphere 3	.2	3000
sphere 4	.2	100
detector 1	.4	10
detector 2	.15	2000

Another example of the DD card and a description of its output is in Chapter 5. For a more detailed discussion of the Russian roulette game, see page 97 in Chapter 2.

19. DXT DXTRAN Card

Form: DXT:n x_1 y_1 z_1 RI_1 RO_1 x_2 y_2 z_2 RI_2 RO_2 ... DWC_1 DWC_2 DPWT

n = N for neutrons and P for photons
x_i y_i z_i = coordinates of the point at the center of the i^{th} pair of spheres
RI_i = radius of the i-th inner sphere in cm
RO_i = radius of the i-th outer sphere in cm
DWC1 = upper weight cutoff in the spheres
DWC2 = lower weight cutoff in the spheres
DPWT = minimum photon weight. Entered on DXT:N card only.

Defaults: Zero for DWC_1, DWC_2, and DPWT.

Use: Optional. Consider using the DXC:N, DXC:P, or DD cards when using DXTRAN.

DXTRAN is used to improve the particle sample in the vicinity of a tally (see page 128). It should not be misconstrued as a tally itself, such as a detector; it is used in conjunction with tallies as a variance reduction technique. DXTRAN spheres must not overlap.

The technique is most effective when the geometry inside the spheres is very simple and can be costly if the inside geometry is complicated, involving several surfaces. The inner sphere is intended to surround the region of interest. The outer sphere should surround neighboring regions that may scatter into the region of interest. In MCNP, the relative importance of the two regions is five. That is, five times as many particles (with one fifth the weight) are aimed at the inner sphere.

Rule of Thumb for RI and RO: The inner radius RI should be at least as large as the tally region, and RO–RI should be about one mean free path for particles of average energy at the spheres. DXTRAN can be used around detectors, but the combination may be very sensitive to reliable sampling.

There can be up to five sets of X Y Z RI RO on each DXT card. There is only one set of DWC_1 and DWC_2 entries for each particle type. This pair is entered after conclusion of the other data and (with DXT:N) before the one value of DPWT. The weight cutoffs apply to pseudoparticle tracks inside the outer radii, have default values of zero, and function exactly as the two weight cutoffs do on the CUT cards. The DXTRAN photon weight cutoffs have no effect unless the simple physics is used, with one

exception: upon leaving the sphere, track weights (regardless of what physics is used) are checked against the cutoffs of the CUT:P card.

The minimum photon weight limit DPWT on the DXT:N card parallels almost exactly the minimum photon weight entries on the PWT card. One slight difference is that in Russian roulette during photon production inside DXTRAN spheres, the factor for relating current cell importance to source cell importance is not applied. Thus, the user must have some knowledge of the weight distribution of the pseudoparticles (from a short run with the DD card, for example) inside the DXTRAN sphere, so the lower weight limit for photon production may be intelligently specified. As in the case of the PWT entries, a negative entry will make the minimum photon weight relative to the source particle starting weight. The default value is zero, which means photon production will occur at each neutron pseudoparticle collision in a material with nonzero σ_γ inside the DXTRAN sphere.

DXTRAN can be used in a problem with the $S(\alpha,\beta)$ thermal treatment, but no contributions to the DXTRAN spheres are made from any $S(\alpha,\beta)$ thermal collisions. DXTRAN should also be used with the same caution as for a detector in a problem with a reflecting surface (see page 96).

If more than one set of DXTRAN spheres is used in the same problem, they can "talk" to each other in the sense that collisions of DXTRAN particles in one set of spheres cause contributions to another set of spheres. The contributions to the second set have, in general, extremely low weights but may be numerous with an associated large increase in computer time. In this case the DXTRAN weight cutoffs probably will be required to kill the very-low-weighted particles. The DD card can give you an indication of the weight distribution of DXTRAN pseudoparticles.

F. Material Specification Cards

The cards in this section specify the isotopic composition of the materials in the cells and which cross-section evaluations are to be used.

Cards appropriate to this section are

1. Mm Material Card

Form: Mm $ZAID_1$ $fraction_1$ $ZAID_2$ $fraction_2$ . . .

m corresponds to the material number on the cell cards

$ZAID_i$ = either a full ZZZAAA.nnX or partial ZZZAAA element or nuclide identifier for constituent i where ZZZ is the atomic number, AAA is the atomic mass, nn is the library identifier, and X is the class of data

$fraction_i$ = atomic fraction (or weight fraction if entered as a negative number) of constituent i in the material.

Default: None.

Use: Optional, but required if you want materials in cells.

Neutrons. For naturally occurring elements, AAA = 000. Thus, ZAID = 74182.01 represents the isotope ^{182}W and ZAID = 74000.01 represents the element tungsten. Natural elements not available from among those listed in Appendix G must be constructed on an Mm card by adding together the individual isotopes if they are available.

If the density for cells with AAA = 000 is input in g/cm^3, MCNP will assume the atomic weight for the natural element. If the isotopic distribution for the element differs from the natural element, the atom density should be entered on the cell cards to ensure the correct atom density for these cells. This is most important for photon-only problems.

The ZZZ and AAA quantities are determined for neutrons by looking at the list of cross sections in Appendix G and finding the appropriate ZAID associated with an evaluation that you want. If you do not find an evaluation for a nuclide that you need, contact X-6.

Photons. If a photon-only problem is being run, the AAA can be set to 000 and the nnX can be omitted. Photon cross sections are specified exactly like the neutron cross sections, but for photons ZZZAAA.nnX equals ZZZ000. There is no distinction between isotope and element for photons.

Photon cross sections are available from the public file MCPLIB for Z = 1 to 94.

Nuclide Fraction. The nuclide fractions may be normalized to 1.0 or left unnormalized. For instance, if the material is H_2O the fractions can be entered as 0.667 and 0.333 or as 2 and 1 for H and O respectively. If the fractions are entered with negative signs they are assumed to be weight fractions; otherwise, atomic fractions. Weight fractions and atom fractions cannot be mixed on the same Mm card.

The total number of "nuclide fraction" entries allowed is MEMAX = 150. The total number of different cross-section tables allowed is MAXE = 50.

2. DRXS Discrete Reaction Cross-Section Card

Form: DRXS $ZAID_1$ $ZAID_2$ . . . $ZAID_i$ . . .
or *blank*

$ZAID_i$ = Identifying number of the form ZZAAA.nn, where ZZ is the atomic number, AAA the mass number, and nn the neutron library identifier.

Default: Continuous-energy cross-section treatment if DRXS is absent.

Use: Optional.

Nuclides listed on the optional DRXS card are given a discrete energy treatment instead of the regular fully continuous-energy cross-section treatment if the necessary discrete data are available. Check the list in Appendix G for availability. If the DRXS card is present but has no entries after the mnemonic, discrete cross sections will be used for every nuclide, if available.

Unless you are transporting neutrons in an energy region where resonances and hence self-shielding are of little importance, it is not recommended that this card be used. However, if the problem under consideration meets this criterion, using the DRXS card can reduce computer storage requirements and enhance timesharing.

All discrete reaction libraries are based on a 262 energy group structure rather than the previous 240 energy groups. Several groups have been added below 1 eV, making the discrete treatment much more appropriate for thermal neutron problems near room temperature. Also, all discrete reaction libraries (except for DRL79) now have photon production data given in expanded format.

Use of these discrete cross sections will not result in the calculation being what is commonly referred to as a multigroup Monte Carlo calculation because the only change is that the cross sections are represented in a histogram form rather than a continuous-energy form. The angular treatment used for scattering, energy sampling after scattering, etc., is performed using identical procedures and data as in the continuous-energy treatment. The user wanting to make a truly multigroup Monte Carlo calculation should use the MCMG patch to MCNP.

3. TOTNU Total Fission Card

Form: TOTNU NO
or *blank*

Default: If the TOTNU card is absent, prompt $\bar{\nu}$ is used for non-KCODE calculations and total $\bar{\nu}$ is used for KCODE calculations.

Use: All steady-state problems should use this card.

In a non-KCODE problem, the absence of a TOTNU card causes prompt $\bar{\nu}$ to be used for all fissionable nuclides for which prompt $\bar{\nu}$ values are available. If a TOTNU card is present but has no entry after it, total $\bar{\nu}$ will be used for those fissionable nuclides for which total values are available. A TOTNU card with NO as the entry is the same as if the card were absent, that is, prompt $\bar{\nu}$ is used.

In a KCODE calculation, the absence of a TOTNU card causes total $\bar{\nu}$ to be used for all fissionable nuclides for which total values are available. If a TOTNU card is present but has no entry after it, total $\bar{\nu}$ is again used. A TOTNU card with NO as the entry causes prompt $\bar{\nu}$ to be used for all fissionable nuclides for which prompt values are available.

The nuclide list of Appendix G indicates data available for each fissionable nuclide. The MCNP neutron cross-section summary print from XACT will show whether prompt or total was used.

4. NONU Fission Turnoff Card

Form: NONU $a_1\ a_2\ \ldots\ a_i\ \ldots\ a_{mxa}$
or *blank*

$a_i = 0$ fission in cell i to be treated as capture
$a_i = 1$ fission in cell i to be real
mxa = number of cells in the problem

Default: If the NONU card is absent, fission is treated as real fission.

Use: Optional, as needed.

This card allows turning off fission in any cell. The fission is then treated as simple capture and accounted for on the loss side of the ledger table as the "Dead Fission" entry. If the NONU card is not used, all cells are given their regular treatment of real fission, that is, the same as if all entries were one. If the NONU card is present but blank, all a_i's are

assumed to be zero and fission in all cells is treated like capture. The NONU card cannot be added to a continue-run.

Sometimes it is desirable to run a problem with a fixed source in a multiplying medium. For example, if an operating reactor power distribution were specified as a function of position in the core, neither the KCODE nor the non-KCODE (without the NONU card) portion of the code would calculate the problem correctly. The KCODE calculation would converge to its own power distribution, whereas the non-KCODE calculation would be impossible to run because of the criticality of the system. Using the NONU card in the non-KCODE mode allows this problem to run correctly by treating fission as simple capture.

5. AWTAB Atomic Weight Card

Form: AWTAB $ZAID_1$ AW_1 $ZAID_2$ AW_2 ...

$ZAID_i$ = ZAID used on the Mm material card but not including the nn specification.

AW_i = atomic weight

Default: If the AWTAB card is absent, MCNP will use the atomic weights in the cross section tables.

Use: Optional, as needed.

Entries on this card override the existing atomic weights as contained in the cross-section tables.

6. XSn Cross-Section File Card

Use: Optional, as an alternative to the XSDIR file.

The XSn card serves the same purpose as the XSDIR file. You can use XSn cards in place of or in addition to the XSDIR file. Each XSn card, with n = 1 to 99, is used to describe one cross section table. The entries for the XSn card are identical to those in XSDIR with the following two exceptions:

(1) the + is not used for continuation, and

(2) the pairs of ZA and atomic weight are omitted.

The latter are needed only for photon-only problems and must then be input on the AWTAB card. A detailed description of the required entries is provided in Appendix F.

7. VOID Material Void Card

Form: VOID no entries
or: VOID C_1 C_2 . . . C_i

C_i = cell number

Default: None.

Use: Debugging geometry and calculating volumes.

The first form is used when calculating volumes stochastically (see page 160) and in checking for geometry errors (see page 173). When there are no entries on the VOID card, all cells in the problem are made void by setting the material number and density to zero, FM cards are turned off, heating tallies are turned into flux tallies, and, if there is no NPS card, the effect of an NPS 100000 card is created. If there is a TALLYX subroutine, it may need to be changed, too.

The second form is used to selectively void cells instead of replacing the material number and density on each selected cell card by a zero by hand. It can be a convenience if you want to check whether the presence of some object in your geometry makes any significant difference in the answers.

G. *Energy and Thermal Treatment Specification*

The following cards are involved with energy and other physics aspects of MCNP:

Mnemonic (Columns 1-5)	Card Type	Page
PHYS	Energy physics cutoff	252
ESPLT	Energy splitting and roulette	253
TMP	Free-gas thermal temperature	254
THTME	Thermal times	255
MTm	$S(\alpha,\beta)$ material	256

All energy entries on these cards are in units of MeV, and all times are in shakes.

1. PHYS Energy Physics Cutoff Card

(a) Neutrons

Form: PHYS:N E_{max} EMCNF

E_{max} = upper limit for neutron energy, MeV.
EMCNF = energy boundary (MeV) above which neutrons are treated with implicit capture and below which they are treated with analog capture.

Default: E_{max} = very large; EMCNF = 0.0 MeV.

Use: Optional.

E_{max} is the upper limit for neutron energy. All neutron cross-section data above E_{max} are expunged. If E_{max} is not specified, there is no upper energy expunging of cross-section data to save computer storage space. The physics of MCNP is such that if a neutron energy is greater than the maximum energy in a table (typically 20 MeV), the cross section for the maximum energy is used with no extrapolation. If, either by source or collision, a particle is born above E_{max}, it is rejected and the particle energy is resampled.

EMCNF controls the type of capture. Any neutron with energy greater than EMCNF will receive the implicit capture treatment; below EMCNF, it will receive analog capture. This parameter is analogous to EMCPF on the PHYS:P card and is useful in eliminating low-energy histories when using a thermal treatment. Substantial computer time may be saved in a region of low absorption (especially if the region is heterogeneous and bounded by a reflecting surface) simply by reducing the number of tracks. EMCNF should be set to operate when a neutron enters a thermal regime, typically a few kT. However, analog capture may undesirably kill important particles before they are tallied or participate in physics important to the problem.

If EMCNF = E_{max}, analog capture is used regardless of the value of WC1 on the CUT card. If WC1 = 0, analog capture is used regardless of the value of EMCNF.

(b) Photons

Form: PHYS:P EMCPF

EMCPF = Photons with energy greater than EMCPF (in MeV) will be tracked using the simple physics treatment that includes implicit capture. Photons with energy less than EMCPF

will be treated with the more detailed physics that includes analog capture.

Default: If this card is omitted, the detailed treatment will be used at all energies below 100 MeV.

Use: Optional.

Photon cross-section data are available from 0.001 MeV to 100 MeV. Below the lower limit, MCNP allows only analog capture regardless of EMCPF, resulting in quick termination of photon histories.

The simple physics treatment, intended primarily for higher energy photons, considers the following physical processes: photoelectric effect (without fluorescence), scattering (Compton) from free electrons without the use of form factors, and pair production. The highly peaked coherent Thomson scattering is ignored.

The detailed physics treatment differs in that for photons of energies between 1 keV and 100 MeV there is provision for fluorescent emission, and the Thomson and Klein-Nishina differential cross sections are modified by by appropriate form factors which take electron binding effects into account.

For a detailed discussion of the simple and detailed photon physics treatments, see Chapter 2.

2. ESPLT Energy Splitting and Roulette Card

Form: ESPLT:n N_1 E_1 . . . N_5 E_5

n = N for neutrons. and P for photons.
N_i = number of tracks into which a particle will be split.
E_i = energy (MeV) at which particles are to undergo splitting.

Default: Omission of this card means that energy splitting will not take place for those particles for which the card is omitted.

Use: Optional; use energy-dependent weight windows instead.

The ESPLT card allows for splitting and Russian roulette in energy, as the IMP card allows for splitting and Russian roulette as a function of geometry. Energy splitting may result in low weight particles that are inadvertently killed by the weight cutoff game (CUT card). Because energy dependent weight windows perform the same function as the ESPLT card, are not limited to five energy groups, may have spatial dependence, and are

more compatible with other variance reduction features, use of the ESPLT card is discouraged.

For a coupled neutron/photon problem, both ESPLT:N and ESPLT:P cards can be used.

The entries on this card consist of pairs of energy-biasing parameters, N_i and E_i, with a maximum of five pairs allowed. (The number of pairs can be different on the ESPLT:N and ESPLT:P cards.) N_i can be noninteger and also can be between 0 and 1, in which case Russian roulette on energy is played. For N_i between 0 and 1 the quantity becomes the survival probability in the roulette game.

If the particle's energy falls below E_i, the specified splitting or roulette always occurs. If the particle's energy increases above E_i, the inverse game is normally played. For example, suppose roulette is specified at 1 eV with survival probability 0.5; if a particle's energy increases above 1 eV, it is split 2 for 1.

A neutron's energy may increase by fission or from thermal up-scattering. There are cases when it may not be desirable to have the inverse splitting or roulette game played on energy increases (particularly in a fission-dominated problem). An option allows splitting or roulette to be played only for energy decreases and not for energy increases when the splitting parameters N_i are negative.

Example: ESPLT:N 2 .1 2 .01 .25 .001

This example specifies a 2 for 1 split when the neutron energy falls below 0.1 MeV, another 2 for 1 split when the energy falls below 0.01 MeV, and Russian roulette when the energy falls below 0.001 MeV with a 25% chance of surviving.

3. TMP Free-Gas Thermal Temperature Card

Form: TMPn T_{1n} T_{2n} ... T_{in} ... T_{In}

n = index of time on the THTME card
T_{in} = temperature of i^{th} cell at time n, in MeV
I = number of cells in the problem

Default: 2.53×10^{-8} MeV, room temperature.

Use: Optional. Required when THTME card is used. Needed for low-energy neutron transport at other than room temperature.

The TMP cards provide MCNP the time-dependent thermal cell

temperatures that are necessary for the free-gas thermal treatment of low-energy neutron transport described on page 57. This treatment becomes important when the neutron energy is about 4 times higher than the temperature of heavy nuclei or 400 times higher than the temperature of light nuclei. Thus the TMP cards should be used when parts of the problem are not at room temperature and neutrons are transported with energies within a factor of 400 from the thermal temperature.

Thermal temperatures are entered as a function of time with a maximum of 99 time entries allowed. These times are entered on a thermal time (THTME) card. The thermal temperatures at time t_{1n} are listed, cell by cell, on the TMP1 card; the cell thermal temperatures at time t_{2n} are listed on the TMP2 card, etc. A linear interpolation is used to determine the cell thermal temperatures at times between two entries. Time values before t_{1n} or after t_{ln} use the thermal temperatures at the nearest time entry.

A fatal error occurs if a zero temperature is specified for a nonvoid cell.

Consider using the parameter EMCNF on the PHYS:N card to reduce the number of thermal collisions, which are generally quite time consuming.

We use kT to denote the thermal temperature of a cell and use units of MeV. The following formulas can be used to provide the values of kT for temperatures in degrees kelvin, Celsius, Rankine, and Fahrenheit.

$$
\begin{aligned}
kT(\mathrm{MeV}) &= 8.617\times10^{-11}T \text{ where T is in degrees K} \\
&= 8.617\times10^{-11}(T+273.15) \text{ where T is in degrees C} \\
&= 4.787\times10^{-11}T \text{ where T is in degrees R} \\
&= 4.787\times10^{-11}(T+459.67) \text{ where T is in degrees F}
\end{aligned}
$$

4. THTME Thermal Times Card

Form: THTME $t_1\ t_2\ \ldots\ t_n\ \ldots\ t_N$

t_n = time in shakes at which thermal temperatures are specified on the TMP card
N = total number of thermal times specified

Default: Zero; temperature is not time dependent.

Use: Optional. Use with TMP card.

The THTME card specifies the times at which the thermal temperatures on the TMPn cards are provided. The temperatures on the TMP1 card are at time t_1 on the THTME card, the temperatures on the TMP2 card are at time t_2

on the THTME card, etc. The times must be monotonically increasing: $t_n < t_{n+1}$. For each entry on the THTME card there must be a TMPn card.

5. MTm $S(\alpha,\beta)$ Material Card

Form: MTm X

X = $S(\alpha,\beta)$ identifier corresponding to a particular component on the Mm card.

Default: None.

Use: Optional, as needed.

For any material defined on an Mn card, a particular component of that material (represented by a ZAID number) may be associated through an MTm card with an $S(\alpha,\beta)$ data set if that data set exists. The $S(\alpha,\beta)$ data for that ZAID are used in every cell in which that material is specified. For a particular ZAID in a material, the free-gas treatment may be used down to the energy where $S(\alpha,\beta)$ data are available. At that point, the $S(\alpha,\beta)$ treatment automatically overrides the free-gas treatment (that is, there is no mixing of the two treatments for the same ZAID in the same material at a given energy).

Typically the free-gas model will be used for a particular ZAID of a material down to 4 eV and then the $S(\alpha,\beta)$ treatment will take over. In general, $S(\alpha,\beta)$ effects are most significant below 2 eV.

The $S(\alpha,\beta)$ treatment is invoked by identifiers on MTm cards. The *m* refers to the material *m* defined on a regular Mm card. The appearance of an MTm card will cause the loading of the corresponding $S(\alpha,\beta)$ data from the thermal data file.

Substantial computer time may be saved when using the $S(\alpha,\beta)$ treatment by using the parameter EMCNF on the PHYS:N card to reduce the number of thermal collisions.

The currently available $S(\alpha,\beta)$ identifiers for the MTm card are listed in Table G.1 of Appendix G.

There is currently no theory or code in MCNP for $S(\alpha,\beta)$ contributions to detectors or DXTRAN spheres.

Examples:
```
M1     1001 2  8016 1       light water
MT1    LWTR.07

M14    1001 2  6012 1       polyethylene
MT14   POLY.03
```

M8	6012 1	graphite
MT8	GRPH.01	

H. Problem Cutoff Cards

The following cards can be used in an initiate-run or a continue-run input file to specify parameters for some of the ways to terminate tracks in MCNP.

The cards in this section are:

1. CUT Cutoffs Card

Form: CUT:n T E WC1 WC2 SWTM

n = N for neutrons and P for photons
T = time cutoff in shakes, 1 shake=10^{-8} sec,
E = lower energy cutoff in MeV,
WC1 and WC2 = weight cutoffs, and
SWTM = minimum source weight

Use: Optional, as needed.

Neutron default: T=very large, E=0.0 MeV, WC1=-0.50, WC2=-0.25, SWTM=minimum source weight if the general source is used.

For a coupled neutron/photon problem, both a CUT:N and a CUT:P card can be used.

If a neutron's time becomes greater than T, its transport is stopped and it is killed. Even though MCNP is time dependent, neutron decay is not considered. Any neutron with energy lower than E is killed.

If a neutron's weight WGT falls below WC2 times the ratio R of the source cell importance to the current cell importance, then with probability WGT/(WC1*R), the neutron survives and is assigned WGT=WC1*R. If negative values are entered for the weight cutoffs, the values

$$|WC1| \cdot W_s \text{ and } |WC2| \cdot W_s$$

will be used for WC1 and WC2, respectively, where W_s is the minimum weight

assigned to a source neutron from an MCNP general source. These negative entries are recommended for most problems. If only WC1 is specified, then WC2=0.5*WC1. See page 126 for a discussion of weight cutoffs.

In a coupled neutron/photon problem, photons are generated before the neutron weight cutoff game is played.

If WC1 is set to zero, capture is treated explicitly by analog rather than implicitly by reducing the neutrons's weight according to the capture probability. If EMCNF = E_{max} on the PHYS card, analog capture is used regardless of the value of WC1 except for neutrons leaving a DXTRAN sphere.

SWTM (source weight minimum) can be used to make the weight cutoffs relative to the minimum starting weight of a source particle for user source as is done automatically for the general source. The entry will in general be the minimum starting weight of all source particles, including the effects of energy and direction biasing. The entry is also effective for the general source as well. Then SWTM is multiplied by the W entry on the SDEF card but is unaffected by any directional or energy biasing. This entry is ignored for a KCODE calculation.

Photon default: T=neutron cutoff, E=0.001 MeV, WC1=−0.50, WC2=−0.25,
SWTM=minimum source weight if the general source is used.

The CUT:P weight cutoffs are analogous to the CUT:N card except that they are used only for energies above the EMCPF entry on the PHYS:P card (see page 252) and there is no provision for explicit analog capture in the simple physics treatment. For energies below EMCPF, the cutoffs have no effect because the detailed physics considers only analog capture, with one exception: photons leaving a DXTRAN sphere. Their weight is always checked against the CUT:P weight cutoff upon exiting. If only WC1 is specified, then WC2 = 0.5*WC1.

In a coupled neutron/photon problem, the photon weight cutoffs are the same as the neutron weight cutoffs unless overridden by a CUT:P card. Again, the photon weight cutoffs have no effect at energies below EMPCF (except with DXTRAN as noted above).

MCNP allows only analog capture below 0.001 MeV. Since the photoelectric cross section is virtually 100% of the total cross section below that energy for all isotopes, tracks will be quickly captured and terminated.

2. NPS History Cutoff Card

Form: NPS N

N = number of particle histories.

Default: None.

Use: As needed to terminate the Monte Carlo calculation.

The single entry N on this card is used to terminate the Monte Carlo calculation after N histories have been transported – unless the calculation is terminated earlier for some other reason such as computer time cutoff.

In a continue-run, NPS is the total number of particles including runs before the continue-run; it is cumulative. However, a negative NPS entry means to print an output file at the time of the last history run and then stop.

In a surface source problem, either more or less than all of the particle histories on the RSSA surface source file will be run, depending on the value N entered on the NPS card. If N < NP1, where NP1 is the number of original histories, Russian roulette with weight adjustment will be played with each history in the file, using a survival probability of N/NP1. If N > NP1, the histories will be split N/NP1 to 1, and the fractional part is taken care of by sampling. This can be done equally well for nonspherical sources by cell importance splitting. With a spherical source, each multiple occurrence of the history is sampled for a different starting location on the source sphere, possibly improving the spatial statistics of the results. In either case, the use of the NPS card will not provide additional information about the original source distributions or the transport to the recording surface crossing.

In a criticality calculation, the NPS card has no meaning and a warning error message is issued if it is used.

3. CTME Computer Time Cutoff Card

Form: CTME x

x = maximum amount of computer time (in minutes) to be spent in the Monte Carlo calculation.

Default: None.

Use: As needed.

For a continue-run job the time on the CTME card is the time relative to the start of the continue-run; it is not cumulative.

Five normal ways to terminate an MCNP calculation are the NPS card, the CTME card, the job time limit, the end of a surface source file, and the number of cycles on a KCODE card. If more than one is in effect, the

one hit first will control. MCNP checks the computer time remaining in a running problem and will terminate the job itself, leaving enough time to wrap up and terminate gracefully.

On CTSS, the time remaining can be changed interactively while a job is running through the use of (CTRL-E)IT *time* feature, where time is the new time remaining in minutes.

I. User Data Arrays

Two arrays, IDUM and RDUM, are in MCNP variable COMMON and are available to the user. They are included in the dumps on the RUNTPE file and may therefore be used for any purposes, including accumulating information over the entire course of a problem through several continue-runs. Each array is dimensioned 50, and they may be filled by cards in the input file INP. IDUM is an integer array and RDUM is a floating point array.

1. IDUM Integer Array Card

Form: IDUM $I_1 \ldots I_n$, $1 \le n \le 50$

Default: All array values zero.

Use: Useful only in user-modified versions of MCNP.

The entries (up to 50) on this card fill the IDUM array with integer numbers. If floating point numbers are entered on this card, they will be truncated and converted to integers.

2. RDUM Floating Point Array Card

Form: RDUM $R_1 \ldots R_n$, $1 \le n \le 50$

Default: All array values zero.

Use: Useful only in user-modified versions of MCNP.

The entries (up to 50) on this card fill the RDUM array with floating point numbers.

J. Peripheral Cards

The following cards have nothing to do with the transport of particles but offer a variety of conveniences:

1. PRDMP Print and Dump Cycle Card

Form: PRDMP NDP NDM

NDP = increment for printing tallies
NDM = increment for dumping to RUNTPE file

Default: Print only after the calculation has successfully ended.
Dump every 15 minutes and at the end of the problem.

Use: Recommended, especially for complex problems.

The PRDMP card allows the user to change the interval at which tallies are printed to the OUTP file and information is dumped to the RUNTPE file. Positive entries mean that after every NDP histories the summary and tallies are printed to the output file, and after every NDM histories a dump is written to the run file. A negative entry changes the unit from histories to minutes of computer time.

In a criticality calculation, positive entries on the PRDMP card are interpreted as the number of cycles rather than the number of particles started. Printing and dumping are done only at the ends of cycles.

2. LOST Lost Particle Card

Form: LOST LOST(1) LOST(2)

LOST(1) = number of particles which can be lost before the job aborts
LOST(2) = maximum number of debug prints that will be made for lost particles

Defaults: 10 lost particles and 10 debug prints.

Use: Discouraged. Losing more than 10 particles is rarely justifiable.

The word "lost" means that a particle gets to an ill-defined section of the geometry and does not know where to go next. This card should be used cautiously: you should know why the particles are being lost, and the number lost should be statistically insignificant out of the total sample. Even if only one of many particles gets lost, there could be something seriously wrong with the geometry specification. Geometry plots in the area where the particles are being lost can be extremely useful in isolating the reason that particles are being lost. See page 173.

3. DBCN Debug Information Card

Form: DBCN x_1 x_2 x_3 . . . x_{11}

x_1 = the starting pseudo random number
x_2 = debug print interval
x_3 and x_4 = history number limits for event log printing
x_5 = maximum number of events in the event log to print per history. Zero is interpreted as 600.
x_6 = flag to produce a file containing timing statistics
x_7 = flag to produce a detailed print from the volume and surface area calculations
x_8 = number of the history whose starting pseudorandom number is to be used to start the first history of this problem.
x_9 = collision number for the collision loop breakpoint
x_{10} = seconds between time interrupts. Zero is interpreted as 10.
x_{11} = flag to allow collision lines to print in lost particle event log.

Defaults: All entries are defaulted to zero, which has a special meaning as described below.

Use: Optional.

The 11 entries on this card are used primarily for debugging problems and the code itself. The J jump feature is quite useful with this card. Because this card can be used in a continue-run, saving the RUNTPE is extremely valuable for diagnosing troubles that occur late in a long-running problem.

(1) The first entry is the random number used for starting the transport of the first particle history in a run. See also entry x_8. If you have difficulties with history number 124 and wish to restart a calculation with history 124 as the first history, you could enter the random number that started history 124. X_1 must be odd to satisfy requirements of the random

number generator. The beginning random number of a particular errant history is labeled SOURCE RN on a debug print. The starting random number of the last particle run is also listed on the Creation and Loss ledger table of the regular output. The zero default entry means use $(5^{19})^2$. See the caution after the eleventh DBCN item.

(2) X_2 is used to print out information about every $x_2{}^{th}$ particle. The information consits of: (a) the particle history number, (b) the total number of collisions, (c) the total number of random numbers generated, and (d) the random number at the beginning of the history. This information is printed at the beginning of the history and is preceded by the letters DBCN in the output to facilitate a pattern search.

(3) and (4) Event log printing is done for NPS between these limits. The information includes a step-by-step account of a history, such as where and how a particle is born, which surface it crosses and which cell it enters, what happens to it in a cell, etc. If event logs from a late portion of a long-running problem are desired and the RUNTPE is available, the first dump before the desired event logs can be used in a continue-run with the DCBN card.

(5) This controls the maximum number of events in the event log to print per history. The zero default entry is interpreted as 600.

(6) If this sixth entry is nonzero, a binary file SAMFIL is produced containing timing statistics of the various parts of MCNP from a memory access pattern analysis. This is a feature of CTSS only. The timing routine is statistical; every few milliseconds it scores what routine the job is in. These scores, or number of hits NHIT, are listed on the HTALO output file along with percentages. If the entry is greater than 4, it indicates the interval between samples, in milliseconds.

(7) A nonzero entry here causes a detailed print from the volume and surface area calculations. This is of use only to X-6 code developers.

(8) If x_8 is nonzero, the starting random number of the problem is the random number which would normally be picked for the $x_8{}^{th}$ history. It is usually easier to adjust the random number sequence with x_8 instead of x_1, which requires you to type in the correct pseudorandom number. See the *caution* after the eleventh DBCN item. The zero default entry means 1.

(9) When the number of collisions in the current history NCH(IPT) equals the number entered for x_9, a normally unused piece of code is traversed. This piece of code contains an otherwise unused label (260 in HSTORY) where

a breakpoint can be set. The breakpoint occurs before the call to either the COLIPN or COLIDP subroutines. It can be useful in conjuction with a dynamic debugger when you want to skip over a certain number of collisions and get to the one of interest.

(10) X_{10} is seconds between time interrupts in TRNSPT. The zero default entry is interpreted as 10 seconds. If a history takes more than 10 seconds of computer time, MCNP checks it a second time (sometimes the timing routines are unreliable, so a second check is made). If the history has taken 10 seconds on each of these two checks, MCNP assumes that something is wrong and stops the job. X_{10} can change this time increment and can be used on a continue-run after the initial run terminated because of this time problem.

This entry also affects the time increment MCNP reserves for itself to terminate a job before the job time limit is reached. This increment for interactive jobs is twice x_{10}.

(11) A nonzero entry allows collision lines to print in the lost particle event log.

CAUTION: The TRANSM subroutine plays a Russian Roulette game on contributions to detectors and DXTRAN based on the estimated average contribution per collision after 200 histories. If you want to generate an event log for some history greater than 200 by starting that history as the first one using the DBCN card, the two histories will not be the same because the roulette game based on a different average contribution causes the random number sequences to differ in any problems with detectors or DXTRAN unless the first entry on the DD card is zero.

4. FILES File Creation Card

Form: FILES unit no. filename access form record length

unit no. = 1 to 99
filename = name of the file
access = sequential or direct
form = formatted or unformatted
record length = record length in a direct access file

Default: None; none; sequential; formatted if sequential, unformatted if direct; not required if sequential, no default if direct.

Use: When a user-modified version of MCNP needs files whose characteristics may vary from run to run.

If this card is present, the first two entries are required and must not conflict with existing MCNP units and files. The words "sequential," "direct," "formatted," and "unformatted" may be abbreviated. If more than one file is on the FILES card, the defaults are not much help, but the abbreviations will keep it brief. The maximum number of files allowed is six, unless the dimension of the KUFIL array in Fixed Common is increased.

Example: FILES 21 ANDY S F 0 22 MIKE D U 512

If the filename is DUMN1 or DUMN2, the user may optionally use the execution line message to designate a file whose name might be different from run to run, for instance in a continue-run.

Example: FILES 17 DUMN1
MCNP INP=TEST3 DUMN1=POST3

Caution: The names of any user files in a continue-run will be the same as in the initial run, which means the names are not automatically sequenced if a file of the same name already exists. A second output file from a continue-run will clobber an existing file of the same name. Furthermore, if you are using the FILES card for an input file and do a continue-run, you will have to provide the coding for keeping track of the record number and then positioning the correct starting location on the file when you continue or MCNP will start reading the file at the beginning.

The FILES card is not legal in a continue-run input file because there is no way to readjust dynamically allocated storage for a possible requested increase in buffer space.

5. PRINT Output Prints

Form: PRINT x

x = no entry gives the full output print
x = $x_1, x_2, \ldots$ prints the tables specified by the numbers $x_1, x_2, \ldots$

Default: Absence of the PRINT card or the PRINT option in the execution line will result in a less than full output print.

Use: Optional.

The following output will be printed automatically:
- a listing of the input file

- a listing of the cross sections used
- the creation and loss ledger table
- the problem activity (such as population) in each cell
- summary of photon production if a MODE 1 problem
- the tallies, and
- the tally fluctuation charts.

You will always get the above information; there is no way to turn it off.

The PRINT card with entries (assigned numbers corresponding to output tables) causes the specified tables to be printed in addition to the default tables. The order of the entries on the card is insignificant. Entries are allowed only on the PRINT card and not with the PRINT option in the execution message. The PRINT option takes precedence over the input card, however.

The table number appears in the upper righthand corner of the table for reference. This provides a convenient pattern to search for when scanning the output file on a terminal. The pattern is PRINT TABLE n, where n is preceded always by one space and is a one- or two-digit number from the table below. Following is a summary of the table numbers.

Table Number	Table Description
10	Source coefficients and distribution
20	Weight window information
30	Tally description
40	Material composition
50	Cell volumes and masses, surface areas
60	Cell importances
62	Forced collision and exponential transform
70	Surface coefficients
72	Cell temperatures
80	Use of storage space
90	KCODE source data
100	Cross section tables
110	First 50 starting histories
120	KCODE cycle summary
126	Particle activity in each cell
130	Neutron/photon weight balance
140	Neutron/photon nuclide activity
150	DXTRAN diagnostics
170	Source distribution frequency tables, surface source
180	Weight window generator optional summary
190	Weight window generator summary
200	Weight window generated windows

Example: PRINT 110 40 150

The output file will contain all the default information plus material composion, first 50 histories information, and DXTRAN diagnostics.

Remember that the PRINT control can be used in a continue-run even when it was not used for the initial run. Using the NPS card with a negative entry on it plus the PRINT card in a continue-run input file is a way to create the BCD output file starting with the Ledger Table (which is after Table 110) for the initial run even though you originally had only the short default output.

V. *SUMMARY OF MCNP INPUT FILE*

A. *Input Cards*

The following table lists the various input cards and when they are required. An *R* indicates that the card is required for the specified Mode. An *O* indicates that the card is optional depending on the user's needs and depending upon the other included cards. An *X* indicates that a card should not be used with a specific Mode.

Two kinds of defaults are involved in the following table: (1) if a particular entry on a given card has a default value, that value is listed in the appropriate location on the card, and (2) the omission of a card from the input file sometimes has a default meaning, and if so, the default description is preceded by an asterisk.

Table 3.4 Summary of MCNP Input Cards

Mode			Card and Defaults	
N	N,P	P	General Categories	Page
O	O	O	Message block plus blank terminator	165
R	R	R	Problem title card	168
R	R	R	Cell cards plus blank terminator	175
R	R	R	Surface cards plus blank terminator	177
R	R	R	Data cards plus blank terminator	183
O	O	O	C Comment card	168
			Problem type card	page 184
O	R	R	MODE N	

			<u>Cell parameter cards</u>		page 184
R	R	R	IMP	required unless weight windows used	
0	0	0	VOL	0	
0	0	0	AREA	0	
X	0	X	PWT	-1	
0	0	0	EXT	0	
0	0	0	VECT	none	
0	0	0	FCL	0	
0	0	0	WWE	none	
R	R	R	WWN	required unless importances used	
0	0	0	WWP	5 3 5 0 0	
0	0	0	WWG	none	
0	0	0	WWGE	single energy interval	
0	0	0	PDn	1	
0	0	0	DXC	1	
			<u>Transformation cards</u>		page 197
0	0	0	TRn	MAP = 1	
			<u>Source specification cards</u>		page 199
0	0	0	SDEF	ERG=14 TME=0 POS=0,0,0 WGT=1	
0	0	0	SIn	H I_i ... I_k	
0	0	0	SPn	D P_i ... P_k	
0	0	0	SBn	D B_i ... B_k	
0	0	0	DSn	H J_i ... J_k	
0	0	0	SCn	none	
0	0	0	SSW	SYM 0	
0	0	0	SSR	OLD NEW COL m=0	
a	a	X	KCODE	no default 1 5 0 MAX(4500,1.5*NSRCK) 0	
0	0	X	KSRC		

(a) The KCODE card is required for criticality calculations.

			<u>Tally specification cards</u>		page 219
0	0	0	Fnα	R_o = 0 for n = 5	
0	0	0	FCn	none	
0	0	0	En	very large	
0	0	0	Tn	very large	
0	0	0	Cn	1	
0	0	0	FQn	F D U S M C E T	
0	0	0	FMn	1	
0	0	0	DEn/DFn	none	
0	0	0	EMn	1	
0	0	0	TMn	1	

0	0	0	CMn	1
0	0	0	CFn	none
0	0	0	SFn	none
0	0	0	FSn	none
0	0	0	SDn	0
0	0	0	FUn	(Requires SUBROUTINE TALLYX)
0	0	0	TFn	1 1 last last 1 last last last
0	0	0	DD	0.1 1000
0	0	0	DXT	- - - - - 0 0 0
			<u>Material specification cards</u>	page 246
0	0	0	Mn	none
0	0	X	DRXS	•fully continuous
0	0	X	TOTNU	•prompt $\bar{\nu}$ for non-KCODE; total $\bar{\nu}$ for KCODE
0	0	X	NONU	•fission treated as real fission
0	0	0	AWTAB	•atomic weights from cross-section tables
0	0	0	XSn	none
0	0	0	VOID	none
			<u>Energy and Thermal cards</u>	page 251
0	0	X	PHYS:N	•very large 0
X	0	0	PHYS:P	•100
0	0	0	ESPLT	•no energy splitting or roulette
0	0	X	TMP	2.53×10^{-8}
0	0	X	THTME	0
0	0	X	MTm	none
			<u>Problem cutoffs</u>	page 257
0	0	X	CUT:N	very large 0 -0.5 -0.25 SWTM
X	0	0	CUT:P	very large .001 -0.5 -0.25 SWTM
0	0	0	NPS	none
0	0	0	CTME	none
			<u>User arrays</u>	page 260
0	0	0	IDUM	0
0	0	0	RDUM	0
			<u>Peripheral cards</u>	page 260
0	0	0	PRDMP	end -15
0	0	0	LOST	10 10
0	0	0	DBCN	0 0 0 0 0 0 0 0 0 0 0 0
0	0	0	FILES	- - sequential formatted -
0	0	0	PRINT	•short output

•This describes the effect of not using this particular card.

B. Storage Limitations

Table 3.5 summarizes some of the more important limitations that have to be considered when setting up a problem. It may be necessary to modify MCNP to change one or more of these restrictions for a particular problem.

Table 3.5
Storage Limitations

Entries in the description of a cell	*100 after processing
Total number of tallies	NTALMX = 100
Detectors	MXDT = 20
Neutron DXTRAN spheres	MXDX = 5
Photon DXTRAN spheres	MXDX = 5
NSPLT or PSPLT card entries	*10
Different cross section tables	MAXE = 50
Nuclides (repeatable) on all material cards	MEMAX = 150
$S(\alpha,\beta)$ entries	MXMTX = 100
Entries on IDUM card	*50
Entries on RDUM card	*50

*Set as a dimension in an array

CHAPTER 4

EXAMPLES

In this chapter, cookbook examples of several topics provide instructive, real examples that you can follow and learn from. They should be studied in conjunction with the theory and instructions of Chapters 1, 2, and 3.

You must understand the geometry discussions in Chapters 1 and 2 before studying the following examples. The concept of uniting and intersecting regions of space bounded by surfaces to make a cell must be fully appreciated; the following examples should help solidify this concept.

Following the geometry specification examples are examples of coordinate transformation, tally options, source specifications, a SOURCE subroutine, and SRCDX subroutines for point detectors and/or DXTRAN spheres. The tally examples include the FMn and FSn cards and the TALLYX subroutine for user-defined tallies using the FUn card.

I. GEOMETRY SPECIFICATION

Several more examples of the union and complement operators are given to help you understand these features. In all examples, the cell numbers will be circled; the surface numbers will not be circled but will appear next to the surface they represent. All cells are voids.

All examples in this chapter are available at Los Alamos from CFS under the /X6CODE/CTSS/MCNPEX/UNION node. The input file for the first example is called EXP1, etc. You are encouraged to experiment with these files by plotting and modifying them.

The next several examples become progressively more difficult and usually take advantage of what you learned in the preceding ones. Remember that unless altered by parentheses, the hierarchy of operations is that intersections are performed first and then unions.

Example 1: In Figure 4.1a, surfaces 2 and 4 are cylinders and the others are planes with their positive sides to the right. Cells 1 and 2 are easy to specify:

```
1    0  1 -2 -3
2    0  3 -4 -5
```

Cell 3 is somewhat subtle, and you need to have in mind Figure 1.5 and its

explanation. Remember that a union adds regions and an intersection gives you only the areas that overlap or are common to both regions. Regions may be added together more than once--or duplicated--with the union operator.

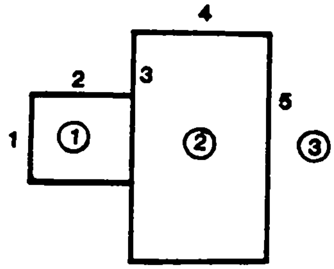


Figure 4.1a

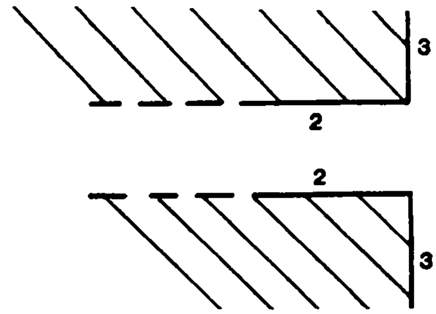


Figure 4.1b

Let us start the definition of cell 3 at surface 2 (this is not a requirement). The expression 2 −3 defines the following region: everything in the *universe* outside surface 2 intersected with everything to the left of surface 3. This region is hatched in Figure 4.1b. Let us examine in detail how Figure 4.1b was derived. First look at each region separately. The area with a positive sense with respect to surface 2 is shown in Figure 4.1c. It includes everything outside surface 2 extending to infinity in all directions. The area with negative sense with respect to surface 2 is undefined so far. The area with negative sense with respect to surface 3 is shown in Figure 4.1d. It includes everything to

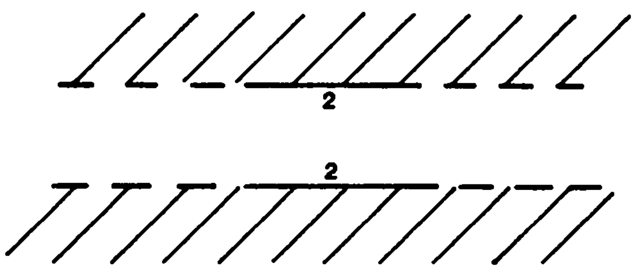


Figure 4.1c

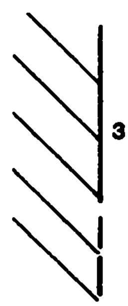


Figure 4.1d

the left of surface 3 extending to infinity, or half the universe. Recall that an intersection of two regions gives only the area common to ***both*** regions or the areas that overlap. Superimposing Figures 4.1c and 4.1d results in Figure 4.1e. The cross-hatched regions show the space common to both regions. This is the same area hatched in Figure 4.1b.

Let us now deal with surface 1. To the quantity 2 −3 we will add everything with a negative sense with respect to surface 1 as indicated by the expression 2 −3 : −1, or (2 −3) : −1 if you prefer. Recall (1) that in

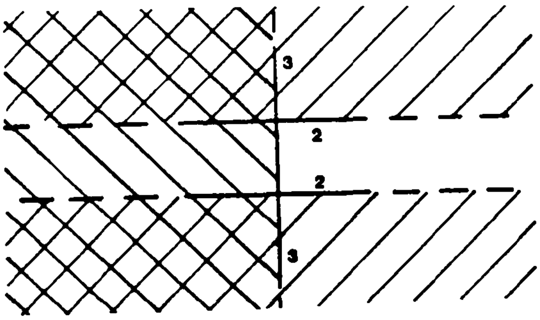


Figure 4.1e

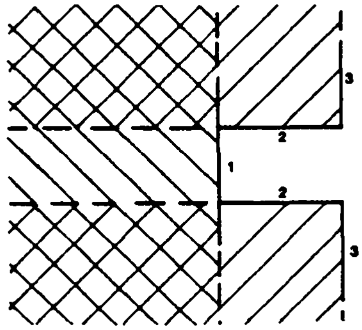


Figure 4.1f

the hierarchy of operations, intersections are performed first and then unions (so the parentheses are unnecessary in the previous expression), and (2) that a union of two regions results in a space containing everything in the first region plus everything in the second region; this includes everything common to both regions. Superimposing the region shown in Figure 4.1b and the region to the left of surface 1 results in Figure 4.1f. Our geometry now includes everything hatched plus everything crosshatched and has added part of the tunnel which is interior to surface 2.

By the same method we will deal with surface 4. To the quantity 2 −3 : −1 we will add everything with a positive sense with respect to surface 4, written as 2 −3 : −1 : 4. Figure 4.1g shows our new geometry. It includes everything in Figure 4.1f plus everything outside surface 4.

Our final step is to block off the large tunnel extending to infinity to the right by adding the region with a positive sense with respect to surface 5 to the region shown in Figure 4.1g. The final expression that defines cell 3 of Figure 4.1a is 2 −3 : −1 : 4 : 5.

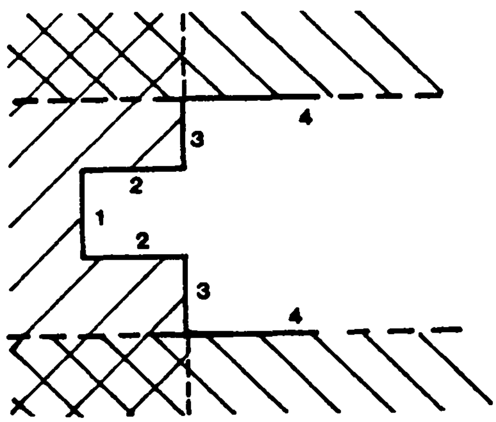


Figure 4.1g

There is more than one way to define cell 3. Starting with surface 1, we can add the region to the left of 1 to the region outside surface 2 or −1 : 2, which is illustrated in Figure 4.1h. We wish to intersect this

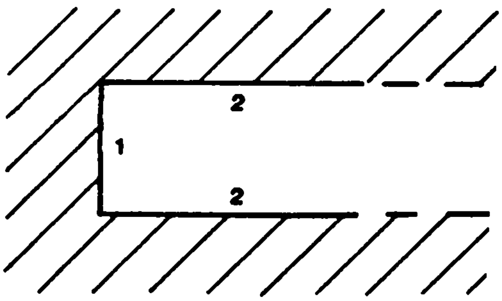


Figure 4.1h

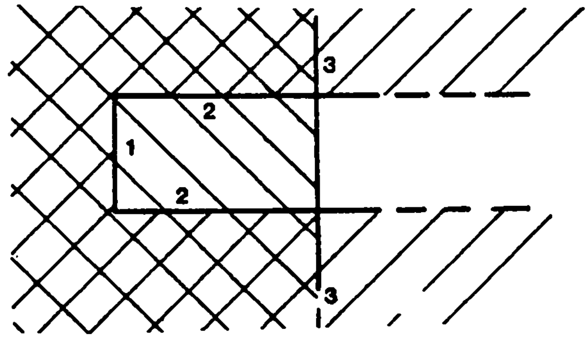


Figure 4.1i

space with the space having a negative sense with respect to surface 3. Superimposing Figure 4.1h and the region to the left of surface 3 results in Figure 4.1i. The cross-hatched area indicates the area common to both regions and is the result of the intersection. Note that the cross-hatched area of Figure 4.1i is identical to the entire hatched plus crosshatched area of Figure 4.1f. Therefore, we have defined the same geometry in both figures but have used two different approaches to the problem. To ensure that the intersection of −3 is with the quantity −1 : 2 as we have illustrated, we must use parentheses giving the expression (−1 : 2) −3. Remember the order in which the operations are performed. Intersections are done before unions unless parentheses alter the order. The final expression is (−1 : 2) −3 : 4 : 5.

Another tactic uses a somewhat different approach. Rather than defining a small region of the geometry as a starting point and adding other regions until we get the final product, we shall start by defining a block of space and adding to or subtracting from that block as necessary. We arbitrarily choose our initial block to be represented by 4 : −1 : 5, illustrated in Figure 4.1j. To this block we need to add the space in the

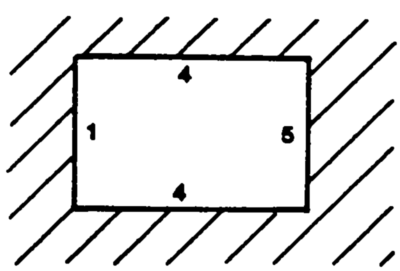


Figure 4.1j

upper and lower left corners. The expression 2 −3 isolates the space we need to add. Adding 2 −3 to our original block, we have 4 : −1 : 5 : (2 −3). The parentheses are not required for correctness in this case but help to illustrate the path our reasoning has followed. Figure 4.1k depicts the union of 2 −3 with the block of space we originally chose.

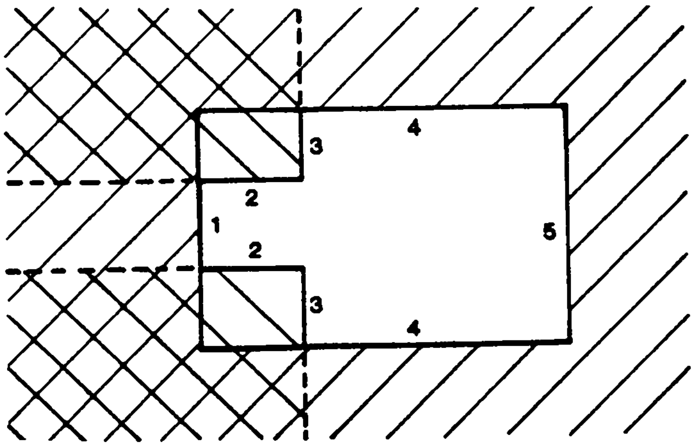


Figure 4.1k

Let us arbitrarily choose a different initial block, 4 : 5 : −3, all the universe except cell 2. From this region we need to subtract cell 1. If we intersect the region (2 : −1) with (4 : 5 : −3), as shown in Figure 4.1*l*, we will have introduced an undefined tunnel to the right of surface 5. To correct this error, define an area (2 : −1 : 3) or (2 : −1 : 5) and intersect this region with the initial block.

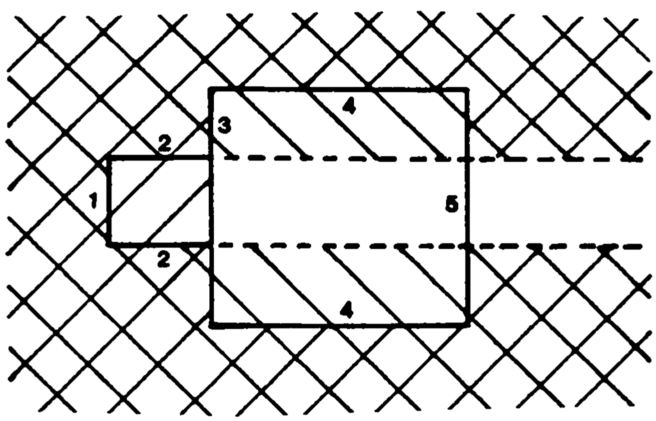


Figure 4.1*l*

Another approach is to intersect the two regions −1 : 2 and −3 : 4, then add that to the region to the right of surface 5 by (−1 : 2)(−3 : 4) : 5. In the above paragraph the expression

(4 : 5 : −3)(2 : −1 : 5) can have the common quantity : 5 factored out, also resulting in (−1 : 2)(−3 : 4) : 5.

Finally, another approach is to forget about the reality of the geometry and for cell 3 take the inverse (or complement) of all the cells bounding cell 3, which is cells 1 and 2. This says that cell 3 is all of the universe excluding that which has already been defined to be in cells 1 and 2. The advantage of this is that cells 1 and 2 are easy to specify and you don't get bogged down in details for cell 3. Cell 3 thus becomes (−1 : 2 : 3)(−3 : 4 : 5). Note that the specifications for cells 1 and 2 are simply reversed. Intersections become unions. positive senses become negative. Then each piece is intersected with the other. There is a complement operator in MCNP which simply is a shorthand notation for the above expression; it is the symbol #, which can be thought of as meaning *not in*. Therefore, cell 3 is specified by #1 #2, translated as everything in the universe that is not in cell 1 and not in cell 2.

Example 2:

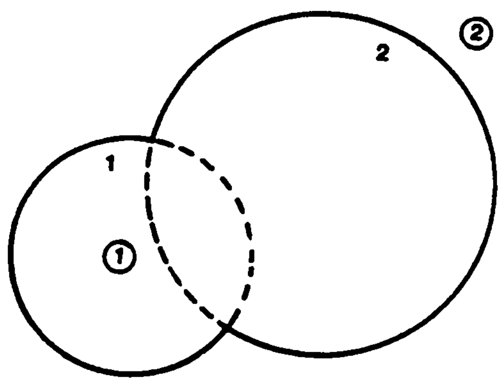


Figure 4.2

Cell 1 is everything interior to the surfaces 1 and 2:

```
1    0  -1  :  -2
2    0   1      2
```

Example 3: In this geometry of four cells defined by three spheres, cell 3 is disconnected. Cell 3 is the region inside surface 3 but outside surfaces 1 and 2 plus the region enclosed between surfaces 1 and 2:

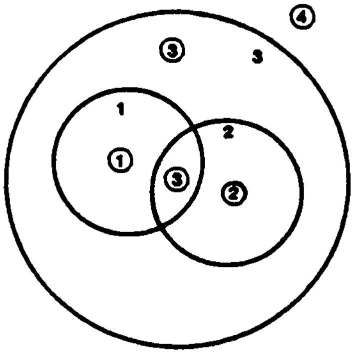


Figure 4.3

1	0	−1	2				
2	0	−2	1				
3	0	−3	1	2	:	−2	−1
4	0	3					

Cell 3 could also be written as

3 0 (−3 1 2) : (−2 −1)

The parentheses are not required.

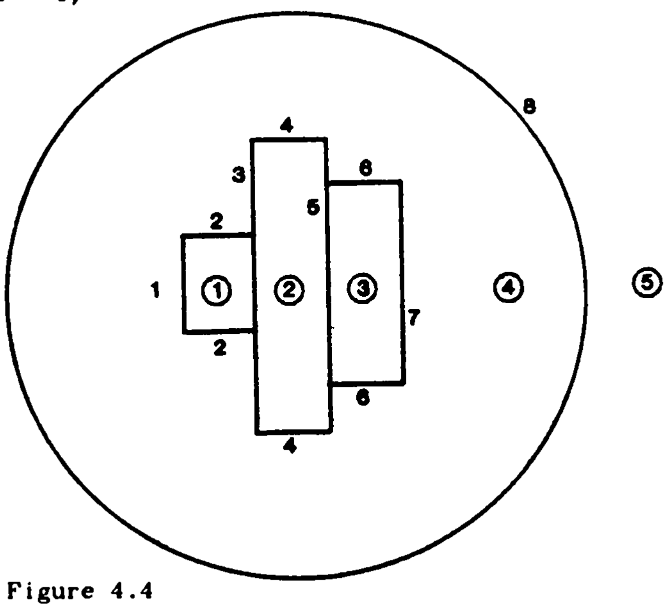


Figure 4.4

Example 4: In this example all vertical lines are planes with their positive sides to the right and all horizontal lines are cylinders. Cells 1, 2, and 3 are simple; they are defined by 1 −2 −3, 3 −4 −5, and 5 −6 −7, respectively. Cell 4 is simple if the complement operator is used; it is #1 #2 #3 #5 or #1 #2 #3 −8. Cell 5 is also simple; it is no more than 8

(or verbally, everything in the universe with a positive sense with respect to surface 8).

If cell 5 were defined as just #4, it would be incorrect. That says cell 5 is everything in the universe not in cell 4, which includes cells 1, 2, and 3. The specification #4 #1 #2 #3 is correct but should not be used since it tells MCNP that cell 5 is bounded by surfaces 1 through 7 in addition to surface 8. This will cause MCNP to run significantly more slowly than it should because anytime a particle enters cell 5 or has a collision in it, the intersection of the particle's trajectory with each bounding surface has to be calculated.

Specifying cell 4 exclusively with the complement operator is very convenient and computationally efficient in this case. However, it will be instructive to set up cell 4 explicitly without complements. There are many different ways to specify cell 4; the following approach should not be considered to be *the* way.

First consider cell 4 to be everything outside the big cylinder of surface 4 that is bounded on each end by surfaces 1 and 7. This is specified by (−1:4:7). The parentheses are not necessary but may add clarity. Now all that remains is to add the corners outside cylinders 2 and 6. The corner outside cylinder 2 is (2 −3), whereas it is (5 6) outside cylinder 6. Again the parentheses are optional. These corners are then added to what we already have outside cylinder 4 to get

(−1:4:7):(2 −3):(5 6)

The region described so far does not include cells 1, 2, or 3 but extends to infinity in all directions. This region needs to be terminated at the spherical surface 8. In other words, cell 4 is everything we have defined so far that is also common with everything inside surface 8 (that is, everything so far intersected with −8). So as a final result,

((−1:4:7):(2 −3):(5 6)) −8

The inner parentheses can be removed, but the outer ones are necessary (remember the hierarchy of operations) to give us

(−1:4:7:2 −3:5 6) −8

If the outer parentheses are removed, the intersection of −8 will occur only with 5 and 6, an event that is clearly incorrect.

Example 5:

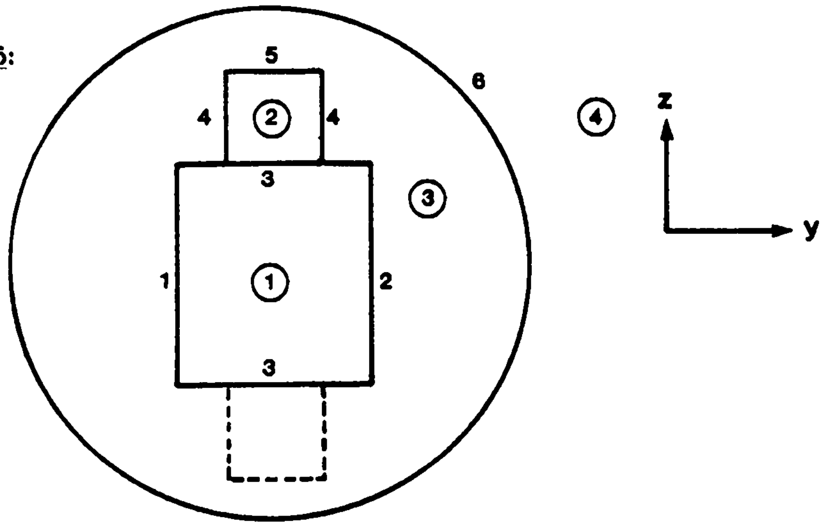


Figure 4.5

This example is similar to the previous one except that a vertical cylinder (surface 4) is added to one side of the horizontal cylinder (surface 3).

Cell 1 is (1 −3 −2), cell 3 is #1 #2 #4, and cell 4 is just 6.

Cell 2 is more than might initially meet the eye. It might appear to be simply (−5 −4 3), but this causes a mirror image of the cell 2 we want to show up on the bottom half of cell 1, as represented by the dashed lines in Figure 4.5. We need to add an ambiguity surface to keep cell 2 above the y−axis. Let surface 7 be an ambiguity surface that is a plane at z=0. This surface appears in the MCNP input file as any other surface. Then cell 2 becomes (−5 −4 3 7) for the final result. You should convince yourself that the region above surface 7 intersected with the region defined by −5 −4 3 is cell 2 (don't even think of surface 7 as an ambiguity surface but just another surface defining some region in space).

Example 6:

This is three concentric spheres with a box cut out of cell 3. Surface 8 is the front of the box and 9 is the back of the box. The cell cards are

```
1      0  -1
2      0  -2  1
3      0  -3  2 (-4:5:-6:7:8:-9)
4      0   3
5      0   4 -5 6 -7 -8 9
```

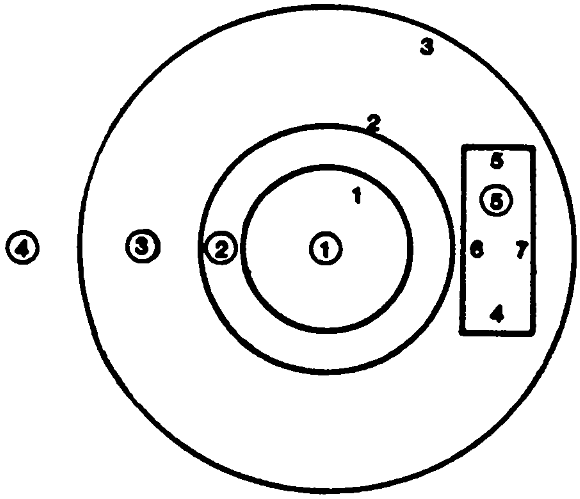


Figure 4.6

Note that the parentheses are required.

Cell 3 is everything inside surface 3 intersected with everything outside surface 2 but not in cell 5. Therefore, cell 3 could be written as

```
3    0  -3 2 #(4 -5 6 -7 -8 9)
```

```
or        3    0  -3 2 #5
or        3    0  -3 2 (-4:5:-6:7:8:-9)
```

Example 7:

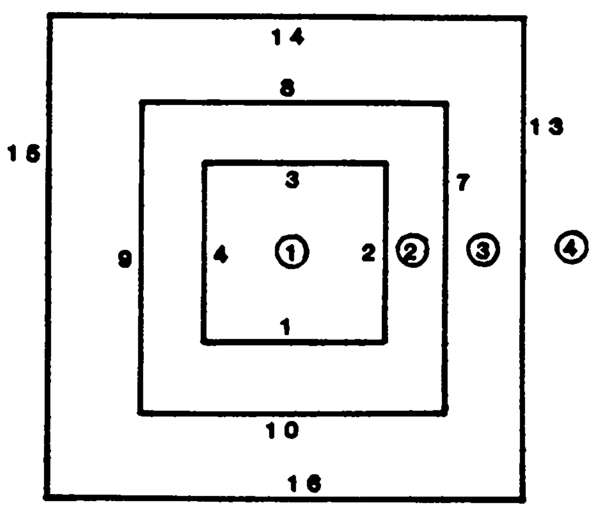


Figure 4.7

This is three concentric boxes, a geometry very challenging to set up using only intersections; it takes a minimum of 14 cells to get around the problem of convex corners when using only intersections. Surfaces 5, 11, and 17 are the back sides of the boxes (smaller to larger, respectively); 6, 12, and 18 are the fronts:

```
1     0  -2     -3     4     1       5     -6
2     0  -7     -8     9    10      11    -12
        (2  :    3  : -4  : -1   :  -5  :   6)
3     0 -13    -14    15    16      17    -18
        (7  :    8  : -9  : -10  : -11  :  12)
4     0  13  :  14  : -15 : -16  : -17  :  18
```

This same example set up entirely with intersections can be seen by looking at the input file BOX, which is available at Los Alamos from CFS from the /X6CODE/CTSS/MCNPEX/UNION node.

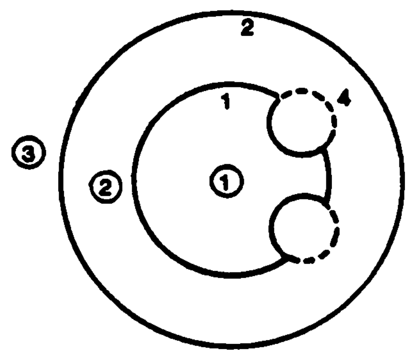


Figure 4.8

Example 8: This is two concentric spheres with a torus attached to cell 2 and cut out of cell 1:

```
1     0  -1   4
2     0  -2  (1  :  -4)
3     0   2
```

If the torus were attached to cell 1 and cut out of cell 2, this bug-eyed geometry would be:

```
1     0  -1  :  -4
2     0  -2   1   4
3     0   2
```

Example 9: Cell 9 is a box cut out of the left part of spherical cell 17; surface 9 is the front of the box and 8 is the rear. Cell 17 is disconnected; the right part is the space interior to the spheres 6 and 7. An F4 tally in cell 17 would be the average flux in all parts of cell 17.

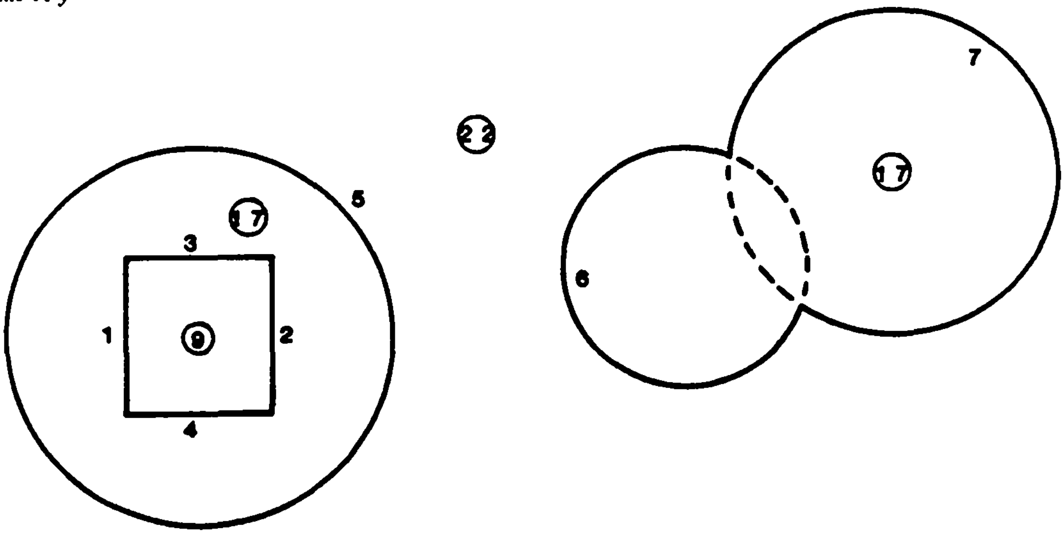


Figure 4.9

An F2 surface tally on surface 7 would be the flux across only the solid portion of surface 7 in the figure. The cell specifications are given by

```
9     0   -3 -2 4 1 8 -9
17    0   -5 (3 : -4 : -1 : 2 : 9 : -8) : -6 : -7
22    0    5 6 7
```

A variation on this problem is for the right portion of cell 17 to be the intersection of the interiors of surfaces 6 and 7 (the region bounded by the dashed lines in the above figure):

```
9     0   -3 -2 4 1 8 -9
17    0   -5 (3 : -4 : -1 : 2 :  9 : -8) : -6 -7
22    0    5 (6 : 7)
```

Example 10: This is a box with a cone sitting on top of it. Surface 6 is the front of the box and 7 is the rear. You should understand this example before going on to the next one.

```
1     0    1    2   -3 (-4 : -5)   -6     7
2     0   -1 : -2 :  3 : 4     5    : 6 : -7
```

Example 11:

Surfaces 15 and 16 are cones, surface 17 is a sphere, and cell 2 is

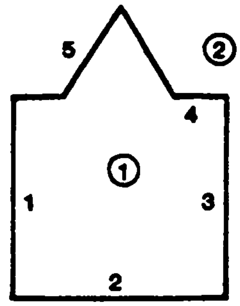


Figure 4.10

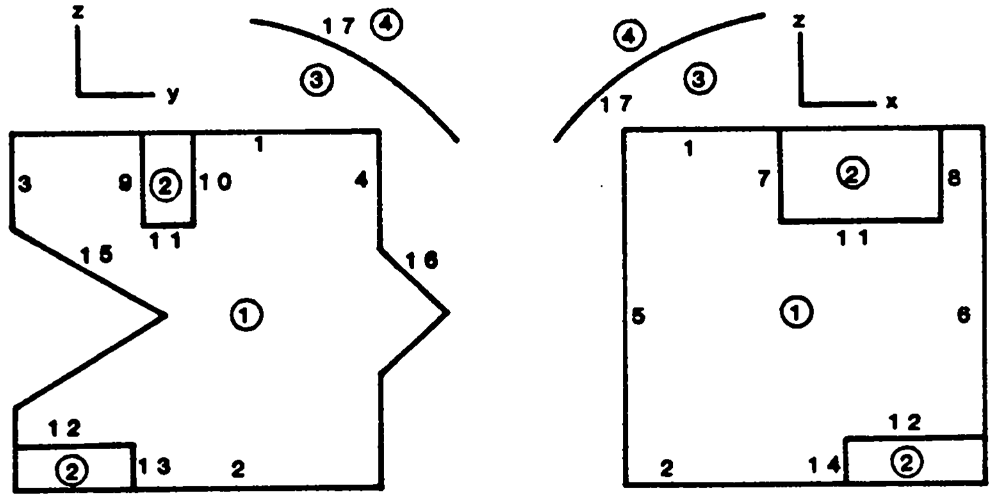


Figure 4.11

disconnected.

1	0	−1 2 3 (−4:−16) 5 −6 (12:13:−14) (10:−9:−11:−7:8) 15
2	0	−10 9 11 7 −8 −1:2 −12 14 −6 −13 3
3	0	−17 (1:−2:−5:6:−3:−15:16 4)
4	0	17

<u>Example 12</u>: Cell 1 consists of two cylinders joined at a 45° angle. Cell 2 is a disk consisting of a cylinder (surface 8) bounded by two planes. Surface 5 is a diagonal plane representing the intersection of the two cylinders. The problem is to specify the disk (cell 2) in one cell formed by the two cylinders (cell 1). A conflict arises in specifying cell 1 since, from the outside cell 3, corner *a* between surfaces 1 and 3 is convex, but on the other side of the cell the same two surfaces form a

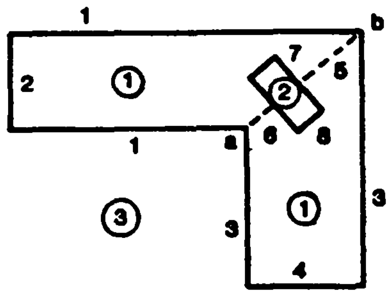


Figure 4.12

concave corner at *b*. The dilemma is solved by composing cell 1 of two disconnected cells, each bounded by surface 5 between the corners *a* and *b*. Surface 5 must be included in the list of surface cards in the MCNP input file. When the two parts are joined to make cell 1, surface 5 does not appear. Convince yourself by plotting it using an origin of 0 0 24 and basis vectors 0 1 1 0 −1 1. See Appendix B for an explanation of plotting commands.

```
1    0  (2 -1 -5 (7:8:-6)):(4 -3 5(-6:8:7))
2    0  -8  6 -7
3    0  (-2:1:5)(-4:3:-5)
```

A more efficient expression for cell 1 is

```
1    0  (2 -1 -5:4 -3 5)(-6:8:7)
```

Example 13: This example has the most complicated geometry so far, but it can be described very simply. The input file is called ANTARES and is available from the /X6CODE/CTSS/MCNPEX/UNION node.

You can see that Example 13 is similar to Example 1. There is just a lot more of it. It is possible to set this geometry up by any of the ways mentioned in Example 1. However, going around the outer surfaces of the cells inside cell 10 is tedious. There is a problem of visualization and also the problem of coming up with undefined tunnels going off to infinity as in Example 1.

The way to handle this geometry is by the last method in Example 1. Set up the cell/surface relations for each interior cell, then just take the complement for cell 10. For the interior cells,

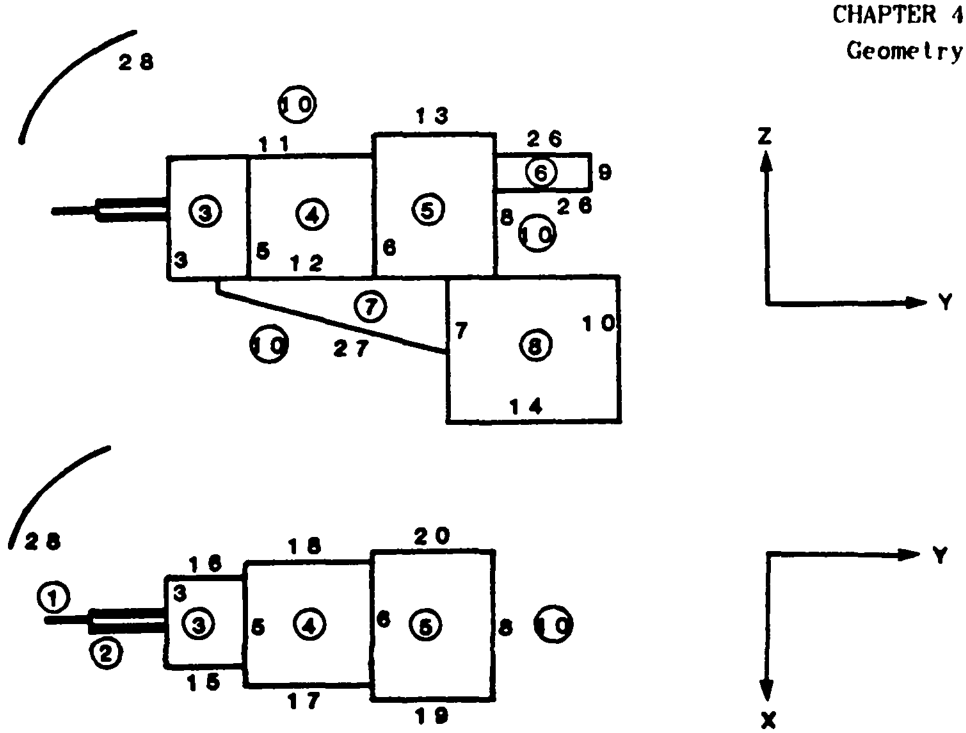


Figure 4.13

```
1    0    1 -2 -23
2    0   -3 25 -24 2
3    0    3 -5 12 -15 16 -11
4    0    5 -6 12 -17 18 -11
5    0    6 -8 12 -13 -19 20
6    0    8 -9 -26
7    0   -12 4 -7 -27
8    0   -12 7 -10 14 -21 22
9    0    2 -3 -25
```

Cell 10 is surrounded by the spherical surface 28. Considering cell 10 to be everything outside cells 1 through 9 but inside surface 28, one can reverse the senses and replace all intersections with unions to produce

```
10   0    (-1:2:23)(3:-25:24:-2)
          (-3:5:-12:15:-16:11)
          (-5:6:-12:17:-18:11)
          (-6:8:-12:13:19:-20)
          (-8:9:26)(12:-4:7:27)
          (12:-7:10:-14:21:-22)
          (-2:3:25)  -28
```

Note how easy cell 10 becomes when the complement operator is used:

10 0 #1 #2 #3 #4 #5 #6 #7 #8 #9 -28

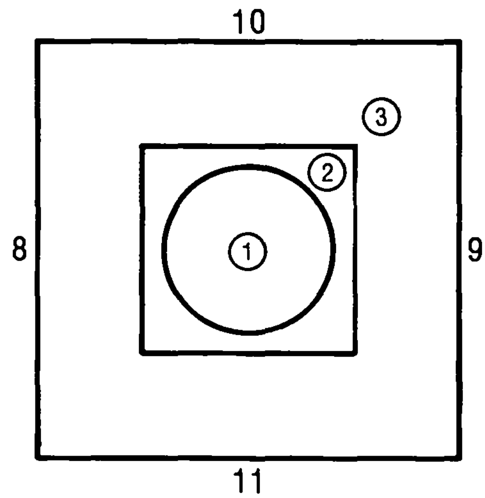


Figure 4.14

Example 14: This example illustrates some necessary conditions for volume and area calculations. The geometry has three cells, an outer cube, an inner cube, and a sphere at the center. If cell 3 is described as

3 0 8 -9 -10 11 -12 13 #2 #1

(and #1 must be included to be correct), the volume of cell 3 cannot be calculated. As described, it is not bounded by all planes so it is not a polyhedron, nor is it rotationally symmetric. If cell 3 is described by listing all 12 bounding surfaces explicitly, the volume can be calculated.

II. COORDINATE TRANSFORMATIONS

In most problems, the surface transformation feature of the TRn card will be used with the default value of 1 for M. When M = 1 applies, most of the geometry can be set up easily in an (x,y,z) coordinate system and only a small part of the total geometry will be difficult to specify. For example, a box with sides parallel to the (x,y,z) coordinate system is simple to describe, but inside might be a tilted object consisting of a cylinder bounded by two planes. Since the axis of the cylinder is neither parallel to nor on the x, y, or z axis, a general quadratic must be used to describe the surface of the cylinder. The GQ surface card has 10 entries that are usually difficult to determine. On the other hand, it is simple

to specify the entries for the surface card for a cylinder centered on the y-axis. Therefore, we define an auxiliary coordinate system (x′,y′,z′) so the axis of the cylinder is one of the primed axes, y′ for example. Now we will use the TRn card to describe the relationship between one coordinate system and the other. M = 1 requires that the coordinates of a vector from the (x,y,z) origin to the (x′,y′,z′) origin be given in terms of (x,y,z).

Only in rare instances will M = −1 be needed. Some unusual circumstances may require that a small item of the geometry must be described in a certain system which we will call (x,y,z), and the remainder of the surfaces would be easily described in an auxiliary system (x′,y′,z′). The O_i entries on the TRn card are then the coordinates of a vector from the (x′,y′,z′) origin to the (x,y,z) origin given in terms of the primed system.

Example 1: The following example consists of a can whose axis is in the yz plane and is tilted 30° from the y-axis and whose center is at (0,10,15) in the (x,y,z) coordinate system. The can is bounded by two planes and a cylinder, as shown in Figure 4.15.

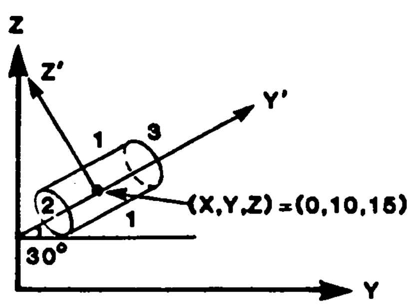


Figure 4.15

The surface cards that describe the can in the simple (x′,y′,z′) system are:

1	1	CY 4
2	1	PY −7
3	1	PY 7

The 1 before the surface mnemonics on the cards is the *n* that identifies which TRn card (up to 10 different ones may be used in the same input file) is to be associated with these surface cards. The TRn card indicates the relationship of the primed coordinate system to the basic coordinate system.

We will specify the origin vector as the location of the origin of the (x′,y′,z′) coordinate system with respect to the (x,y,z) system; therefore,

M = 1. Since we wanted the center of the cylinder at (0,10,15), the O_i entries are simply 0 10 15. If, however, we had wanted surface 2 to be located at (x,y,z) = (0,10,15), a different set of surface cards would accomplish it. If surface 2 were at y′ = 0 and surface 3 at y′ = 14 the O_i entries would remain the same. The significant fact to remember about the origin vector entries is that they describe one *origin* with respect to the other *origin*. The user must locate the surfaces about the auxiliary origin so that they will be properly located in the main coordinate system.

The B_i entries on the TRn card are the cosines of the angles between the axes as listed on page 198 in Chapter 3. In this example, the x-axis is parallel to the x′-axis. Therefore, the cosine of the angle between them is 1. The angle between y and x′ is 90° with a cosine of 0. The angle between z and x′ and also between x and y′ is 90° with a cosine of 0. The angle between y and y′ is 30° with a cosine of 0.866. The angle between z and y′ is 60° with 0.5 being the cosine. Similarly, 90° is between x and z′; 120° is between y and z′; and 30° is between z and z′. The complete TRn card is

```
TR1    0 10 15 1 0 0 0 .866 .5 0 -.5 .866
```

An asterisk preceding TRn indicates that the B_i entries are the angles in degrees between the appropriate axes. The entries using the *TRn mnemonic are

```
*TR1   0 10 15 0 90 90 90 30 60 90 120 30
```

The default value of 1 for M, the thirteenth entry, has been used and is therefore not explicitly specified.

The user need not enter values for all of the B_i. As shown on page 198, B_i may be specified in any of five patterns. Pattern #1 was used above, but the simplest form for this example is pattern #4 since all the skew surfaces are surfaces of revolution about some axis. The complete input card then becomes

```
*TR1    0 10 15 3J 90 30 60
```

<u>Example 2</u>: The following example illustrates another use of the TRn card. The first part of the example uses the TR1 card and an M = 1 transformation; the second part with the TR2 card uses an M = −1 transformation. Both parts and transformations are used in the following input file.

EXAMPLE OF SURFACE TRANSFORMATIONS

```
2     0  -4 3 -5
6     0  -14 -13 : -15 41 -42

3     1  PX -14
4     1  X  -14 10 0 12 14 10
5     1  PX  14
13    2  SX -15 70
14    2  CX  30
15    2  Y   75 0 30 16
41    2  PY  0
42    2  PY  75

TR1    20 31 37 .223954 .358401 .906308
TR2  -250 -100 -65 .675849 .669131 .309017
       J J .918650 J J -.246152 -1
```

A. TR1 and M = 1 Case: Cell 2 is bounded by the plane surfaces 3 and 5 and the spheroid surface 4, which is a surface of revolution about the skew axis x′ in Figure 4.16.

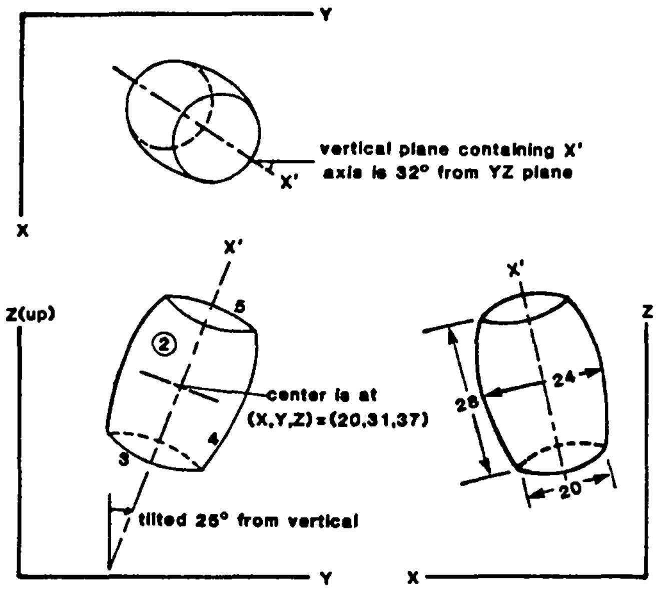
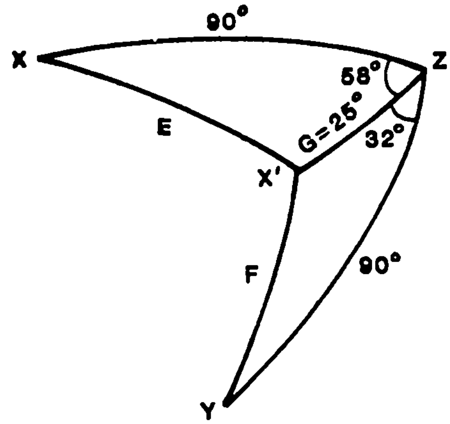


Figure 4.16

To get the coefficients of surfaces 3, 4, and 5, define the x′ axis as shown in the drawings (since the surfaces are surfaces of revolution about the x′ axis, the orientation of the y′ and z′ axes does not matter), then set up cell 2 and its surfaces, with coefficients defined in the x′y′z′ coordinate system.

On the TR1 card, the origin vector is the location of the origin of the x'y'z' coordinate system with respect to the main xyz system of the problem. The pattern #4 on page 198 in Chapter 3 is appropriate since the surfaces are all surfaces of revolution about the x' axis. The components of one vector of the transformation matrix are the cosines of the angles between x' and the x, y, and z axes. They are obtained from spherical trigonometry:

$\cos E = \cos 58^\circ \times \sin 25^\circ = .223954$
$\cos F = \cos 32^\circ \times \sin 25^\circ = .358401$
$\cos G = \cos 25^\circ = .906308$



B. TR2 and M = −1 Case: Cell 6 is the union of a can bounded by spherical surface 13 and cylindrical surface 14 and a conical piece bounded by conical surface 15 and ambiguity surfaces 41 and 42, which are planes. (Surface 42 is required because when surface 15 is defined in x'y'z' it is as a type Y surface, which becomes a cone of one sheet; when it is transformed into the xyz system it becomes a type GQ surface, which in this case is a cone of two sheets. Weird, but that's the way it has to be.) Surfaces 13 and 14 are surfaces of revolution about one axis, and surfaces 15, 41, and 42 are surfaces of revolution about an axis perpendicular to the first axis. Both axes are skewed with respect to the xyz coordinate system of the rest of the geometry.

Define the auxiliary x'y'z' coordinate system as shown in Figure 4.17. Set up cell 6 with its surfaces specified in the x'y'z' coordinate system as part of the input file and add a second transformation card, TR2.

Since the location of the origin of the xyz coordinate system is known relative to the x'y'z' system (rather than the other way around, as in the first part of the example), it is necessary to use the reverse mapping. This is indicated by setting M = −1. In this reverse mapping the origin vector (−250,−100,−65) is the location of the origin of the xyz system with respect to the x'y'z' system. For the components of the transformation matrix, pattern #3 out of the four possible choices from Chapter 3 is most convenient here. The xyz components of z' and the x'y'z' components of z are easy to get. The components of x and of y are not. The whole

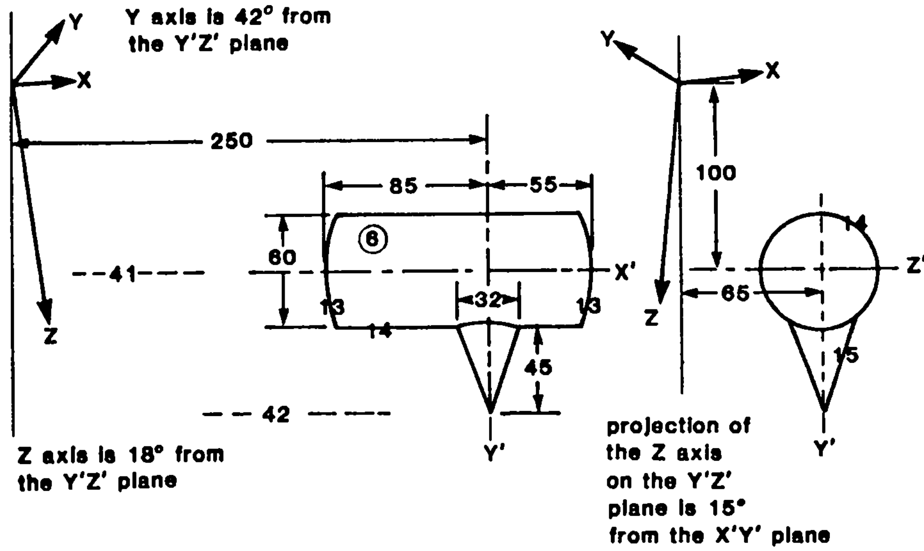


Figure 4.17

transformation matrix is shown here with the components that are obtained from Figure 4.17 written in:

	x	y	z
x′	.675849	cos 48° = .669131	cos 72° = .309017
y′			cos 15° x cos 18° = .918650
z′			−.246152

The zz′ component is −SQRT(1. − .309107**2 − .918650**2) = −.246152, and the xx′ component is SQRT(1. − .669131**2 − .309017**2) = .675849, with the signs determined by inspection of the figure.

III. TALLY EXAMPLES

This section contains several examples of the FMn and FSn tally cards plus the TALLYX subroutine. Refer also to page 228 for the FMn card, to page 236 for the FSn card, and to page 240 for TALLYX before trying to understand these examples.

A. FMn Examples (Simple Form)

Example 1: Consider the following input cards.

```
F4:N    10
FM4     0.04786   999     102
M999    92238.13  1
```

The F4 neutron tally is the track length estimate of the average fluence in cell 10. Material 999 is ^{238}U with an atomic fraction of 100%.

C	= 0.04786	normalization factor (such as atom/barn·cm)
M	= 999	material number for ^{238}U as defined on the material card (with an atom density of 0.04786 atom/barn·cm)
R_1	= 102	ENDF reaction number for radiative capture cross section (microscopic)

The average fluence is multiplied by the microscopic (n,γ) cross section of ^{238}U (with an atomic fraction of 1.0) and then by the constant 0.04786 (atom/barn·cm). Thus the tally 4 printout will indicate the number of ^{239}U atoms/cm^3 produced as a result of (n,γ) capture with ^{238}U.

Standard F6 and F7 tallies can be duplicated by F4 tallies with appropriate FM4 cards. The FM4 card to duplicate F6 is

```
FM4    C  M  1  -4.
```

For F7 it is

```
FM4    C  M  -6  -8.
```

C	= 10^{-24} x	number of atoms per gram
R_1	= 1	ENDF reaction number for total cross section (barns)
R_2	= −4	reaction number for average heating number (MeV/collision)
R_1	= −6	reaction number for total fission cross section (barns)
R_2	= −8	reaction number for fission Q (MeV/fission)

This technique applied to F2 tallies can be used to estimate the average heating over a surface rather than over a volume. It provides the surface equivalents of F6 and F7 tallies, which are not available as standard tallies in MCNP.

<u>Example 2</u>: Consider a point detector.

```
F25:N    0  0  0  0
FM25     0.00253  1001  -6  -8
M1001    92238.13  .9  92235.19  .1
```

This F25 neutron tally is the fission heating per unit volume of material 1001 at the origin. Material 1001 does not actually have to be in a cell at the origin. The FM25 card constants are:

C	= 0.00253	atoms per barn·cm (atomic density) of material 1001
M	= 1001	material number for material being heated
R_1	= −6	reaction number for total fission cross section (barn)
R_2	= −8	reaction number for fission Q (MeV/fission)

More examples: (Remember C = −1 = ρ for type 4 tally)

F5:N	0	0	0	0	Neutron heating per cm^3 with an atom density of
FM5	ρ	M	−1	−4	ρ of material M at a point detector
F5Y:P	10	5	0		Photon heating per cm^3 of material M
FM5	ρ	M	−5	−6	with an atom density ρ at a ring detector
F1:N	1	2	3		Number of neutron tracks crossing
FM1	1	0			surfaces 1, 2, and 3 per neutron started
F35	0	0	0	0	Number of photon collisions per source
FM35:P	1	0			particle that contribute to point detector
M99	3007	1			^{7}Li tritium production per cm^3 in cell 10
F4:N	10				
FM4	−1	99	91		
F104:N	8				Number of reactions per cm^3 of type R in cell 8
FM104	−1	M	R		of material M of atom density ρ

B. FMn Examples (General Form)

Remember that the hierarchy of operation is multiply first and then add and that this hierarchy can not be superseded by the use of parentheses.

Example 1:

```
F4:N   1
FM4    (ρ 1 (−1 −4)(−2)) (ρ 1 −1)    where C = ρ = atomic
M1     6012.10 1                     density (atom/barn·cm)
```

In this example there are three different tallies, namely

(a) ρ 1 −1 −4
(b) ρ 1 −2
(c) ρ 1 −1

Thus tally (a) will yield the neutron heating in MeV/cm^3 from ^{12}C in cell 1. The advantage in performing the multiplication −1 −4 in tally (a) is that the correct statistics are determined for the desired product. This would not be true if tally (a) were to be done as two separate tallies and the product formed by hand after the calculation.

Example 2:
```
F4:N   1
FM4    (0.04635 1 (105:91))
M1     3006.50 0.0742    3007.50 0.9258
```

In this example we obtain the total tritium production per cm^3 from natural lithium (ENDF/B-V evaluation) in cell 1. The constant C on the FM4 card is the atomic density of natural lithium. The subtle point is that R=105 for ^6Li contains (n,t)α and nothing for ^7Li whereas R=91 contains nothing for ^6Li and (n,n')t for ^7Li. Thus the four reactions (two for ^6Li and two for ^7Li) are multiplied by the respective atomic fractions, the neutron flux, and finally the atomic density (atoms/barn·cm).

Example 3: Suppose we have three reactions--R_1, R_2, and R_3--and wish to add R_2 and R_3 and multiply the result by R_1. The following would *not* be valid: FMn (C m R_1 (R_2:R_3)). The correct card is:
FMn (C m (R_1 R_2 : R_1 R_3)).

C. FSn Examples

The FSn card allows you to subdivide your tally into geometry segments without having to overspecify the problem geometry with unnecessary cells.

The entries on the FS card are the names and senses of surfaces that define how to segment any surface or cell tally.

Example 1: Consider a 1-MeV point isotropic source at the center of a 2 cm cube of carbon. We wish to calculate the flux through a 1-cm^2 window in the center of one face on the cube. The input file calculating the flux across one entire face is shown in Figure 4.18.

```
EXAMPLE 1, SIMPLE CUBE
1        1 -2.22   1 2 -3 -4 -5 6     IMP:N=1
2        0         #1                 IMP:N=0  2

1        PY  0
2        PZ -1
3        PY  2
4        PZ  1
5        PX  1
6        PX -1

SDEF     POS=0 1 0   ERG=1
M1       6012.10 -1
F2:N     3
```

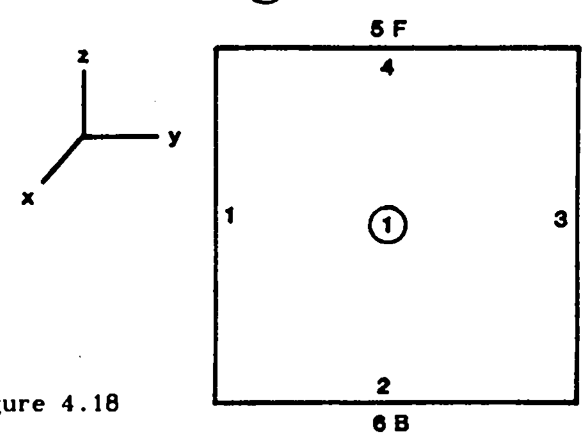


Figure 4.18

Using the FS card, the simple cube geometry is retained, but five more input cards are required,

```
 7    PX    .5
 8    PX   -.5
 9    PZ    .5
10    PZ   -.5
FS2  7  -10  -8  9
```

The four segmenting surface cards are listed with the other surface cards, but they are not part of the actual geometry and hence do not complicate the cell-surface relationships.

In the above example, the F2 tally is subdivided into five separate tallies as shown in Figure 4.19: (1) the first tally is the flux of particles crossing surface 3 but with a positive sense to surface 7; (2) the second is the remaining flux with negative sense to surface 7 crossing surface 3 but with a negative sense to surface 10; (3) the third is the remaining flux (negative sense to 7 and positive sense to 10) crossing 3 but with a negative sense to 8; (4) the remaining flux with positive sense to 9; and (5) everything else. In this example, the desired flux in the window is in the fifth subtally – the "everything else" portion.

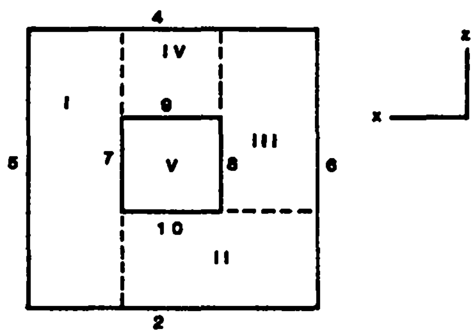


Figure 4.19

The FS segmenting card could have been set up other ways. For example:

```
FS2   -10   7     9   -8     and
FS2    -8   9   -10    7
```

Each works, but the order of the subtallies is changed. A way to avoid the five subtallies and to get only the window is to use the TALLYX subroutine described later.

Example 2: Consider a source at the center of a 10-cm radius sphere called cell 1. We want to determine the fission heating in a segment of the sphere defined by the intersection of the 10-cm sphere, an 8-cm inner sphere, and a 20° cone whose vertex is at the source and is about the Y-axis. This is accomplished by using

```
F7   1
FS7  -2  -3
```

where surface 2 is the 8-cm surface and surface 3 is the cone. This breaks the F7 tally up into three portions: (1) the heating inside the 8-cm sphere; (2) the heating outside the 8-cm sphere but within the cone - this is the desired portion; and (3) everything else, which is a 2-cm shell just inside the 10-cm sphere but outside the cone.

D. TALLYX Subroutine Examples

An explanation of the TALLYX subroutine arguments can be found on page 240 in Chapter 3. Only examples illustrating some uses of TALLYX will be found here.

Example 1: In the example of the FSn card to get the flux through a window on the face of a cube, instead of using the FS2 card, which established five subtallies, TALLYX could have been used to get only the desired window tally. Two input cards are used:

```
FU2    1
RDUM  -.5  .5  -.5  .5
```

The following subroutine (which is implemented just like a user-provided SOURCE subroutine in Appendix A by replacing lines TX.2 through TX.18) does the job. Note that IB=0 and IBU=1 upon entry into TALLYX.

```
      SUBROUTINE TALLYX(T,IB)
*CALL CM
*CALL RC
      IF(X.LT.RDUM(1).OR.X.GT.RDUM(2))IB = -1
      IF(Z.LT.RDUM(3).OR.Z.GT.RDUM(4))IB = -1
      RETURN
      END
```

The subroutine was generalized a bit by using the RDUM input card,

although the card could have been avoided by hardwiring the dimensions of the window into TALLYX.

Example 2: Dump the first 19 words of the PBL array to a BCD file called UOUT each time a neutron crosses surface 15. The cards in the input deck are

```
F2      15
FU2     1
FILES   7   UOUT
```

The user-provided subroutine is

```
      SUBROUTINE TALLYX(T,IB)
*CALL CM
*CALL RC
      WRITE(7,20) (PBL(I),I=1,10),ICL,JSU,IPT,NPA,IEX,NODE,IDX,NCP,KRN
NPA,IEX,NFCH,IPT,NODE,IDX,NCP
   20 FORMAT(5E14.6/5E14.6/7I10/2I10)
      RETURN
      END
```

Every time surface 15 is crossed and the F2 tally is scored, TALLYX is called and part of the PBL array is written to the file UOUT. If more discrimination is desired, such as dump the PBL array only for neutrons with energy between 2.5 and 4.5 MeV and crossing surface 15 at less than 30° with respect to the normal (assume surface 15 has been defined by a PY card), add the following two lines before the WRITE statement:

```
IF(VVV.LT.0.866)RETURN
IF(ERG.LT.2.5.OR.ERG.GT.4.5)RETURN
```

To write a binary file, the FILES card entries are 7 UOUT S U and the WRITE statement in TALLYX is unformatted:

```
      SUBROUTINE TALLYX(T,IB)
*CALL CM
*CALL RC
      WRITE(7) (PBL(I),I=1,10),ICL,JSU,IPT,NPA,IEX,NODE,IDX,NCP,KRN
      RETURN
      END
```

The advantage of a BCD file is that it is easy to look at and manipulate, but it requires more I/O time and a larger file. A binary file

is more compact than a BCD file and requires less I/O time to write; however, it may be more difficult to use.

Example 3: Calculate the number of neutron tracks exiting cell 20 per source neutron. This is also done in Chapter 5 with the TEST1 example using the FMn card. The input cards are

```
F4:N    20
FU4     1
SD4     1
```

and TALLYX becomes

```
      SUBROUTINE TALLYX(T,IB)
*CALL CM
*CALL RC
      T=1.0
      IF(PMF.LT.DLS) IB = -1
      RETURN
      END
```

The quantity T=1.0 is scored every time a track exits cell 20. The variables used in this subroutine, PMF (the distance to collision) and DLS (distance to the boundary), are available to TALLYX from the MCNP COMMON.

Example 4: Divide the point detector scores into separate tallies (that is, user bins) depending upon which of the 20 cells in a problem geometry caused the contributions. The input cards are

```
F5:N    0  0  0  0
FU5     1  18I  20
```

and TALLYX is

```
      SUBROUTINE TALLYX(T,IB)
*CALL CM
*CALL RC
      IBU=ICL9
      RETURN
      END
```

The FU5 card establishes 20 separate user bins, one for each cell in the problem.

<u>Example 5</u>: Determine the quantity $\int \phi(E)\, f(E)\, dE$ in cell 14 where $f(E) = e^{\alpha t}$. The input cards are

```
F4:N    14
FU4     α
```

where α is a numerical value and TALLYX is

```
      SUBROUTINE TALLYX(T,IB)
*CALL CM
*CALL RC
      T=T*EXP(TDS(IPTAL(LIPT+3,1,ITAL)+1)*TME)
      RETURN
      END
```

The FU4 card establishes a single user bin, and the value of α is stored in TDS(IPTAL(LIPT+3,1,ITAL)+1) and used for the tally label.

<u>Example 6</u>: Tally the number of neutrons passing through cell 16 which have had 0,1,2,3, or 4 collisions. The input cards are

```
F4:N    16
FU4     0 1 2 3 4
SD4     1
```

and TALLYX is

```
      SUBROUTINE TALLYX(T,IB)
*CALL CM
      IBU=INT(SPARE1)+1
      IF(IBU.GT.5)IB = -1
      T=WGT
      RETURN
      END
```

The subroutine can be generalized by replacing the 5 in the IF statement with IPTAL(LIPT+3,3,ITAL), which is the number of entries on the FU4 card.

If no tracks were put into the bank (from fission, geometry splitting, etc.), INT(SPARE1) could be replaced by NCH(1), the number of neutron collisions per history. However, to be general, use the quantity SPARE1, which goes into the bank with the rest of the banked track parameters. SPARE1 must be set to 0 when a source particle is started and then incremented at collisions. This is done with the following patch in the STARTP and HSTORY subroutines:

```
*I,SP.30
      SPARE1=0.
*I,HS.196
      SPARE1=SPARE1+1.
```

If the IF statement in this TALLYX is omitted, a count will be made of the cases of five or more collisions, and in these cases no score will be tallied but a count will be printed of the times that the tally was unable to be made because IBU was a value where no bin existed.

In the five user bins, T is the number of neutrons per source neutron passing through cell 16 that has undergone 0, 1, 2, 3, or 4 collisions, respectively. Note that the FU4 card has five entries to establish the five user bins and provide labels. Note also that in this example, the neutrons are calculated so that T=T*renormalization factor (which preserves the weight associated with the tracks), where in Example 3 the neutron tracks are calculated so that T=1. Again the value of SPARE1 is available from COMMON. Finally, note that if SPARE1 > 5 (six or more collisions) no tally is made because IB is set to be less than zero. If an E4 card were added, the neutrons would be tallied as a function of energy for each user bin.

IV. SOURCE EXAMPLES

Some examples of the general source are given here to illustrate the power and complexity of this feature. Refer to Chapter 3 for the more complete explanation and other examples.

Example 1:

```
SDEF ERG D1  DIR FERG D2
     SUR=1  CEL=2  POS=X Y Z  RAD D5  VEC=U V W
SI1  H  10^-7 10^-5 ... 13.5  14 ... 20      $ Level 1
SP1  D  0   10^-4 ... 10^-2  10^-1 ... .3
DS2  S  3   3 ...   3     4 ...  4
SI3     0  .2 ...  1                        $ Level 2
SP3  D  0  10^-4 ...  .1
SI4     0  .1 ...  1
SP4  D  0  10^-2 ...  .1
SI5    37
SP5   -21  1                                $ Optional card
```

This example of the general source illustrates two levels of dependency. Let us assume a duct streaming problem where the source at the duct opening has been obtained from a reactor calculation. Energies above 13.5 MeV have one angular distribution and energies below 13.5 MeV have a

different angular distribution. The source has a uniform spatial distribution on a circular disk of radius 37 cm centered at x,y,z on planar surface 1 going into cell 2.

This example can be expanded by having the source in two ducts instead of one (with the same energy and angular distribution as before). The SI1, SP1, DS2, SI3, SP3, SI4, and SP4 cards remain unchanged. The SDEF card is changed as shown below and the other cards are added.

```
SDEF ERG D1  DIR FERG D2
     SUR D6  CEL FSUR D7  POS FSUR D8
     RAD FSUR D9  VEC FSUR D10
SI6  L     1          7
SP6  D    .6         .4
DS7  L     2          8
DS8  L  x1 y1 z1   x2 y2 z2
DS9  S    11         12
DS10 L  u1 v1 w1   u2 v2 w2
SI11       0  37
SP11     -21   1
SI12       0  25
SP12     -21   1
```

Example 2: This example is a two-source-cell problem where the material in one cell is uranium and in the other is thorium. The uranium cell has two isotopes, ^{235}U and ^{238}U, and the thorium has one, ^{232}Th. Each isotope has many photon lines from radioactive decay. The following input cards describe this source.

```
SDEF  CEL D1  ERG FCEL D2 ...
SC1   source cells             $ Level 1
SI1   L  1  2
SP1   D  2  1
SC2   source "spectra"
DS2   S  3  4
SC3   uranium nuclides         $ Level 2
SI3   S  5  6
SP3   D  1  3
SC4   thorium nuclide
SI4   S  7
SP4   D  1
SC5   U235 photon lines        $ Level 3
SI5   L  E1  ...  EI
SP5   D  I1  ...  II
SC6   U238 photon lines
```

```
SI6   L  E1  ...  EJ
SP6   D  I1  ...  IJ
SC7   Th232 photon lines
SI7   L  E1  ...  EK
SP7   D  I1  ...  IK
```

Example 3:

```
SDEF   SUR D1  CEL FSUR D2  ERG FSUR D6
       X FSUR D3  Y FSUR D4  Z FSUR D5
SI1      L    10   0
SP1     .8    .2
DS2      0    88
DS6      S    61   62
SP61    -3   .98    2.2
SP62    -3  1.05    2.7
DS3      S     0   31
SI31    20    30
SP31     0     1
DS4      S     0   41
SI41   -17    36
SP41     0     1
DS5      S     0   51
SI51   -10    10
SP51     0     1
```

Of the particles from this source, 80% start on surface 10, and the rest start in cell 88. When a particle starts in cell 88, its position is sampled, with rejection, in the rectangular polyhedron bounded by x=20 to 30, y=−17 to 36, and z=−10 to 10. When a particle starts on surface 10, its cell is found from its position and direction. The energy spectrum of the particles from surface 10 is different from the energy spectrum of the particles from cell 88. Notice how a zero under the S option invokes the default value for the variable.

Example 4:

```
SDEF   ERG=D1  DIR FERG D2  SUR=m
SI1      E1   E2 ... Ek
SP1      0    P2 ... Pk
DS2      Q    .3 21   .8 22   1.7 23   20. 24
SP21    -21   1
SP22    -21   1.1
SP23    -21   1.3
SP24    -21   1.8
```

This is an example of using the Q option. The low-energy particles from surface m come out with a cosine distribution of direction, but the

higher-energy particles have a more nearly radial distribution. The energy values on the DS2 card need not be the same as any of the E_i on the SI1 card.

V. *SOURCE SUBROUTINE*

When possible, you should take advantage of the standard sources provided by the code rather than write a source subroutine. When you write your own source subroutine, you lose features such as sampling from multiple distributions, using dependent distributions, and having frequency prints for each tabular distribution. Also, subroutine SRCDX is not then needed, even for anisotropic distributions.

The standard sources, however, cannot handle all problems. If the general source (SDEF card), surface source (SSR), or criticality source (KCODE card) is unsuitable for a particular application, MCNP provides a mechanism to furnish your own source-modeling capability. The absence of SDEF, SSR, or KCODE cards causes MCNP to call subroutine SOURCE, which you must supply. Subroutine SOURCE specifies the coordinates, direction, weight, energy, and time of source particles as listed and defined on page 200. If the value of IPT (particle type) set by STARTP, which calls SOURCE, is not satisfactory, SOURCE must also specify IPT. STARTP sets IPT=1 (neutron) for MODE N and MODE N P and sets IPT=2 (photon) for MODE P. MCNP checks the user's source for consistency of cell, surface, direction, and position. If the source direction is anisotropic and there are point detectors or DXTRAN spheres, an SRCDX subroutine is also required (see Section V of this chapter). Appendix A shows how to get the SOURCE subroutine into MCNP with HISTORIAN.

The following example of a subroutine SOURCE uses SIn, SPn, and SBn cards and demonstrates the use of MCNP subroutines SMPSRC, ROTAS, CHKCEL, and the function NAMCHG. The geometry is a 5-cm-long cylinder centered about the y-axis, divided into 5 cells by PY planes at 1-cm intervals. The 1-MeV monoenergetic source is a biased isotropic distribution that is also biased along the y-axis. The input distribution cards for this source follow.

```
SI1  -1 0 1           $ These 3 cards
SP1   0 1 1           $ represent a biased
SB1   0 1 2           $ isotropic distribution.
SI2   0 1 2 3 4 5     $ These 3 cards
SP2   0 4 2 2 1 1     $ represent a biased
SB2   0 1 1 2 2 4     $ distribution in y.
RDUM  1               $ cylindrical radius
IDUM  2 4 6 8 10      $ source cells
```

This problem can be run with the general source by removing the RDUM and IDUM cards and adding:

```
SDEF  ERG=1  VEC=0 1 0  AXS=0 1 0  DIR=D1  EXT=D2  RAD=D3
 SI3  0 1       $ represents a covering surface of radius 1
 SP3  -21 1     $ samples from the power law with k=1
```

```
*IDENT SRCEX
*D,SO.13
      WGT=1.
C    RDUM(1)--RADIUS OF SOURCE CYLINDER.
C    SAMPLE RADIUS UNIFORM IN AREA.
      R=RDUM(1)*SQRT(RANG())
C    Y COORDINATE POSITION, PROBABILITY AND BIAS ARE
C    DEFINED IN DISTRIBUTION 2 BY THE SI2, SP2, SB2 CARDS.
C    SAMPLE FOR Y.
C    IX RETURNS THE INDEX SAMPLED, AND
C    FR THE INTERPOLATED FRACTION.
C    NEITHER ARE USED IN THIS EXAMPLE.
      CALL SMPSRC(YYY,2,IX,FR)
C    SAMPLE FOR X AND Z.
      TH=2.*PIE*RANG()
      XXX=-R*SIN(TH)
      ZZZ=R*COS(TH)
C    DIRECTION IS ISOTROPIC BUT BIASED IN CONE ALONG Y-AXIS
C    DEFINED AS DISTRIBUTION 1 BY THE SI1, SP1, SB1 CARDS.
C    SAMPLE FOR CONE OPENING C=COS(NU).
C    ROTAS SAMPLES A DIRECTION U,V,W at AN ANGLE ARCCOS(C)
C    FROM THE REFERENCE VECTOR UOLD,VOLD,WOLD
C    AND AT AN AZIMUTHAL ANGLE SAMPLED UNIFORMLY.
      CALL SMPSRC(C,1,IX,FR)
      UOLD=0.
      VOLD=1.
      WOLD=0.
      CALL ROTAS(C)
C    CELL SOURCE - FIND STARTING CELL.
C    IDUM(1)-IDUM(5)--LIST OF SOURCE CELLS (PROGRAM NAME).
      JSU=0
      DO 10 I=1,5
      ICL=NAMCHG(1,IDUM(I))
      CALL CHKCEL(ICL,2,J)
      IF(J.EQ.0) GO TO 20
   10 CONTINUE
```

```
      CALL EXPIRE(1,'SOURCE',
     1 'SOURCE IS NOT IN ANY CELLS ON THE IDUM CARD.')
   20 ERG=1.
      TME=0.
```

VI. SRCDX SUBROUTINE

If a user has supplied a subroutine SOURCE that does not emit particles isotropically (uniform emission in all directions) and is using either a detector tally or DXTRAN in the calculations, then subroutine SRCDX must also be supplied to MCNP. The structure of this subroutine is the same as for subroutine SOURCE, except that usually only a single parameter, PSC, needs to be specified for each detector or set of DXTRAN spheres. PSC as defined in SRCDX is used to calculate the direct contribution from the source to a point detector, to the point selected for the ring detector, or DXTRAN sphere. Other parameters may also be specified in SRCDX. For example, if a quantity such as particle energy and/or weight is directionally dependent, its value must be specified in both subroutine SOURCE and SRCDX. When using detectors and a subroutine SOURCE with an anisotropic distribution, check the direct source contribution to the detectors carefully to see if it is close to the expected result.

In general, it is best to have as many parameters as possible not directionally dependent in subroutine SOURCE. Directionally dependent parameters must also be dealt with in subroutine SRCDX.

The most general function for emitting a particle from the source in the laboratory system can be expressed as $p(\mu,\varphi)$, where μ is the cosine of the polar angle and φ is the azimuthal angle in the coordinate system of the problem. Most anisotropic sources are azimuthally symmetric and $p(\mu,\varphi) = p(\mu)/2\pi$. The quantity $p(\mu)$ is the probability density function for the μ variable only (that is, $\int p(\mu)\, d\mu = 1$, $p(\mu) \geq 0$). PSC is $p(\mu_o)$, where μ_o is the cosine of the angle between the direction defining the polar angle for the source and the direction to a detector or DXTRAN sphere point in the laboratory system. (MCNP includes the 2π in the calculation automatically.) Note that $p(\mu_o)$ and hence PSC may have a value greater than unity and must be non-negative. It is valuable to point out that every source must have a cumulative distribution function based on $p(\mu,\varphi)$ from which to sample angular dependence. The probability density function $p(\mu,\varphi)$ needs only to be considered explicitly for those problems with detectors or DXTRAN.

Table 4.1 gives the equations for PSC for six continuous source probability density functions. More discussion of probability density functions is given in the detector theory section of Chapter 2 (see

Table 4.1

Continuous Source Distributions
and their Associated PSC's

	Description	Source Distribution	PSC	Range of μ_o
1.	Isotropic	Uniform	0.5	$-1 \leq \mu_o \leq 1$
2.	Surface Cosine	μ	$2\lvert\mu_o\rvert$	$0 \leq \mu_o \leq 1$ (or $-1 \leq \mu_o \leq 0$)
			0	$-1 \leq \mu_o < 0$ (or $0 < \mu \leq 1$)
3.	Point Cosine	$\lvert\mu\rvert$	$\lvert\mu_o\rvert$	$-1 \leq \mu_o \leq 1$
4.	Point Cosine*	$a+b\mu$	$\frac{2(a+b\mu_o)}{2a+b}$	$0 \leq \mu_o \leq 1$
			$\left(\frac{2(a+b\mu_o)}{2a-b}\right)$	$(-1 \leq \mu_o \leq 0)$
			0	$-1 \leq \mu_o < 0$ (or $0 < \mu_o \leq 1$)
5.	Point Cosine*	$a+b\mu, a\neq 0$	$\frac{a+b\mu_o}{2a}$	$-1 \leq \mu_o \leq 1$
6.	Point Cosine*	$a+b\lvert\mu\rvert$	$\frac{a+b\lvert\mu_o\rvert}{2a+b}$	$-1 \leq \mu_o \leq 1$

*The quantities a and b must have values such that PSC is always non-negative and finite over the range of μ_o.

page 87). The isotropic case is assumed in MCNP; therefore SRCDX is required only for the anisotropic case.

As an example of calculating μ_o, consider a spherical surface cosine source (type 2 in Table 4.1) with several point detectors in the problem. Assume that a point on the spherical surface has been selected at which to start a particle. The value of μ_o for a detector is given by the scalar (or dot) product of the two directions; that is,

$$\mu_o = uu' + vv' + ww' , \tag{4.1}$$

where u, v, and w are the direction cosines of the line from the source point to the point detector location and u', v', and w' are the direction

cosines for either the outward normal if the surface source is outward or the inward normal if the source is inward.

If u = u′, v = v′, and w = w′, then μ_o = 1, indicating that the point detector lies on the normal line. The value of PSC for the detector point is

$$
\begin{aligned}
PSC &= 2|\mu_o|, & \mu_o &> 0 \\
& & (\mu_o &< 0) \\
&= 0, & \mu_o &\le 0 \\
& & (\mu_o &\ge 0)
\end{aligned}
$$

where the parenthetical values of μ_o are for the inward-directed cosine distribution.

For $|\mu_0|$ less than 0.25, PSC is less than 0.5, which is the value for an isotropic source. This means that source emissions for these values of $|\mu_0|$ are less probable than the isotropic case for this source distribution. The converse is also true. Note that if $|\mu_0|$ is greater than 0.5, PSC is greater than one, which is perfectly valid.

An example of a subroutine SRCDX with appropriate HISTORIAN lines for a surface outward cosine distribution is shown in Figure 4.20. This is basically the technique that is used in MCNP to calculate PSC for a spherical surface source in a cosine distribution; the only difference is that MCNP uses the cosines of the direction from the center of the sphere used to select the source point because this is the normal to the spherical surface. The primed direction cosines were calculated in Figure 4.20 to aid in illustrating this example. The direction cosines u, v, and w as defined in Equation (4.1) have already been calculated in subroutine DDDET when SRCDX is called and are available through COMMON.

```
*DELETE SX3A.1
C     CALCULATE PSC FOR AN OUTWARD SURFACE (SPHERE) COSINE
C     DISTRIBUTION.
C     FIND THE DIRECTION COSINES FOR THIS EXAMPLE BASED
C     ON THE SOURCE POINT ON THE SPHERE (X,Y,Z).
      UP = (XXX - RDUM(1))/RDUM(4)
      VP = (YYY - RDUM(2))/RDUM(4)
      WP = (ZZZ - RDUM(3))/RDUM(4)
C     (RDUM(1),RDUM(2),RDUM(3)) ARE THE COORDINATES OF THE CENTER
C     OF THE SPHERE FROM THE RDUM CARD.  RDUM(4) IS THE
C     SPHERE RADIUS.
C     U, V, AND W HAVE BEEN CALCULATED FOR THE CURRENT
C     POINT DETECTOR IN SUBROUTINE DDDET
      PSC=2.*MAX(0,UUU*UP+VVV*VP+WWW*WP)
```

Figure 4.20

The HISTORIAN cards in Figure 4.20 and in the following two examples are the recommended procedure for replacing the existing dummy SRCDX subroutine. Note that line SX3A.1 must always be deleted if a user provides an SRCDX.

For many sources, a discrete probability density function will be used. In this situation, a cumulative distribution function $P(\mu)$ is available and is defined as

$$P(\mu) = \int_{-1}^{\mu} p(\mu')d\mu' \text{ and } P_{i+1} = \sum_{j=1,i} p_j \Delta\mu_j \quad .$$

where p_j is an average value of the probability density function in the interval $\Delta\mu_j$. Thus, the probability density function is a constant p_j in the interval $\Delta\mu_j$. For this case, there are N values of P_i with $P_1 = 0$, $P_{N+1} = 1.0$, and $P_{i-1} < P_i$. Each value of P_i has an associated value of μ_i. Since PSC is the derivative of $P(\mu_o)$, then

$$PSC = \frac{P_i - P_{i-1}}{\mu_i - \mu_{i-1}} , \mu_{i-1} \le \mu_o < \mu_i \quad . \qquad (4.2)$$

This is an average PSC between μ_{i-1} and μ_i and is also an average value of $p(\mu)$ in the specified range of μ.

Frequently, the cumulative distribution function is divided into N equally probable intervals. For this case,

$$PSC = \frac{1}{N} \frac{1}{\mu_i - \mu_{i-1}} \quad .$$

This is precisely the form used in MCNP for calculating contributions to the point detector for elastic scattering with N = 32.

An example of a subroutine SRCDX for a discrete probability density function is shown in Figure 4.21. This subroutine would work with the subroutine SOURCE example on page 303, and would calculate PSC=1/2 for the isotropic distribution.

A biased anisotropic distribution can also be represented by

SIn μ_o μ_1 ... μ_n
SPn 0 p_1 ... p_n
SBn 0 q_1 ... q_n

A reference vector u',v',w' for this distribution is also needed.

The subroutine SOURCE input cards can be modified for this case by changing the SI1, SP1, SB1, and RDUM cards as follows:

```
    SI1   -1 0 1      $ These 3 cards
    SP1    0 2 1      $ represent a biased
    SB1    0 1 2      $ anisotropic distribution.
   RDUM    1 0 1 0    $ cylindrical radius and reference vector
```

SOURCE would sample this anisotropic distribution and SRCDX would calculate the appropriate PSC.

```
*D,SX3A.1
C    THE VARIABLY DIMENSIONED BLOCK SPF HOLDS THE SI, SP, SB ARRAYS.
C    THE KSD ARRAY IS A POINTER BLOCK TO THE SPF ARRAY.
C    THE FOLLOWING STATEMENT FUNCTION IS DEFINED.
       K(I,J)=KSD(LKSD+I,J)
C    RDUM(2),RDUM(3),RDUM(4)--DIRECTION COSINES
C    FOR THE SOURCE REFERENCE DIRECTION.
       AM=UUU*RDUM(2)+VVV*RDUM(3)+WWW*RDUM(4)
C    K(4,1) IS THE LENGTH OF THE DISTRIBUTION.
C    K(13,1) IS THE OFFSET INTO THE SPF BLOCK.
       DO 10 I=1,K(4,1)-1
    10 IF(SPF(K(13,1)+1,I).LE.AM.AND.SPF(K(13,1)+1,I+1).GE.AM)
      1 GO TO 20
       GO TO 30
    20 PSC=(SPF(K(13,1)+2,I+1)-SPF(K(13,1)+2,I))/
      1 (SPF(K(13,1)+1,I+1)-SPF(K(13,1)+1,I))
       PSC=PSC*SPF(K(13,1)+3,I+1)
       RETURN
    30 PSC=0.
```

Figure 4.21

It is extremely important to note that the above case applies only when the source is anisotropic with azimuthal symmetry. For the general case,

$$PSC = 2\pi p(\mu_o,\varphi_o).$$

The 2π factor must be applied by the user since MCNP assumes azimuthal symmetry and in effect divides the user-defined PSC by 2π.

For a continuous $p(\mu,\varphi)$ function, PSC is calculated as above. In the case of a discrete probability density function,

$$\text{PSC} = 2\pi \cdot \overline{p(\mu_o,\varphi_o)} = \frac{2\pi(P_i - P_{i-1})}{(\mu_i - \mu_{i-1})(\varphi_i - \varphi_{i-1})}$$

$$= \frac{2\pi(P_i - P_{i-1})}{\Delta\mu_i \Delta\varphi_i} \quad ,$$

where $\mu_{i-1} \leq \mu_o < \mu_i$, $\varphi_{i-1} \leq \varphi_o < \varphi_i$ and $\overline{p(\mu_o,\varphi_o)}$ is an average probability density function in the specified values of μ_o and φ_o and $P_i - P_{i-1}$ is the probability of selecting μ_o and φ_o in these intervals. For N equally probable bins and n equally spaced $\Delta\varphi$'s, each $2\pi/n$ wide,

$$\text{PSC} = \frac{n}{N}\frac{1}{\Delta\mu_i} \quad .$$

Another way to view this general case is by considering solid angles on the unit sphere. For an isotropic source, the probability $(P_i - P_{i-1})$ of being emitted into a specified solid angle is the ratio of the total solid angle (4π) to the specified solid angle $(\Delta\varphi\Delta\mu)$. Then, PSC≡0.5. Thus, for the general case (normed to PSC≡0.5 for an isotropic source)

$$\text{PSC} = \frac{(0.5)(P_i - P_{i-1})4\pi}{\Delta\mu_i\Delta\varphi_i} = \frac{2\pi(P_i - P_{i-1})}{\Delta\mu_i\Delta\varphi_i} \quad .$$

Note that PSC is greater than 0.5 if the specified solid angle $\Delta\mu\ \Delta\varphi_i$ is less than $(P_i - P_{i-1})4\pi$. This is the same as the previous general expression.

CAUTIONS:

You are cautioned to be extremely careful when using your own subroutine SOURCE with either detectors or DXTRAN. This caution applies to the calculation of the direct contribution from the source to a point detector, point on a ring, or point on a DXTRAN sphere. Not only is there the calculation of the correct value of PSC for an anisotropic source, but there may also be problems with a biased source.

For example, if an isotropic source is biased to start only in a cone of a specified angle (for example, ψ), the starting weight of each particle should be WGT*(1 − cos ψ)/2, where WGT is the weight of the unbiased source (that is, WGT is the expected weight from a total source). Note that the weight in SRCDX must be changed to the expected weight WGT to calculate the direct contribution to a point detector correctly if PSC is defined to be 0.5.

This example can be viewed in a different way. The probability density function for the above biased source is

$$p(\mu) = \frac{1}{1 - \cos \psi}, \quad \text{for } \cos \psi \le \mu \le 1$$

$$= 0 \qquad \text{for } -1 \le \mu < \cos \psi.$$

Thus, PSC is this constant everywhere in the cone and zero elsewhere. Multiplying this PSC and biased starting weight gives

$$WGT*(1 - \cos \psi)*0.5/(1 - \cos \psi)$$

or WGT*0.5, which is the expected result for an isotropic source.

Another source type that requires caution is for a user supplied source that is energy-angle correlated. For example, assume a source has a Gaussian distribution in energy where the mean of the Gaussian is correlated in some manner with μ. In subroutine SRCDX, the μ_o to a point detector must be calculated and the energy of the starting particle must be sampled from the Gaussian based on this μ_o. This must be done for each point detector in the problem, thus guaranteeing that the direct source contribution to each detector will be from the proper energy spectrum. The original energy of the starting particle, as well as all of the other starting parameters, selected in subroutine SOURCE are automatically restored after the direct source contribution to detectors is made. Thus, the subroutine SOURCE is still sampled correctly.

CHAPTER 5

OUTPUT

This Chapter contains four test problem input files that are available for the CRAY machines at Los Alamos from the /X6CODE/CTSS/MCNPEX node on the CFS. DEMO is a simple file illustrating the flexibility of the tally structure. The complete output is not shown. TEST1 illustrates a complete problem output and parts of it are discussed. The third problem, SHELL, illustrates detectors and detector diagnostics. The fourth problem is a brief discussion of the KCODE criticality output (problem GODIVA).

The chapter ends with an illustration of the event-log print and debug print, which can help you find errors if you modify the code or set up a geometry improperly. The DBCN input card is also useful when finding errors, but only the event-log feature is discussed in this chapter.

MCNP prints out warning messages if needed. Do not ignore these warning messages. Look up the pertinent section in the manual if you need explanation to help you understand what you are being warned about.

I. DEMO PROBLEM AND OUTPUT

DEMO has a point isotropic neutron source in the center of a void cube, with energy uniformly distributed from 0.1 to 1.0 MeV. Flux is calculated across each surface of the cube (F2), across the sum of all surfaces (F22), and across the sum of some of the surfaces (F12). Selected pages of the output file follow.

Note how the FQ card in the DEMO input file changes the order of printing. Depending on what you are interested in, the tally output can be made more readable. FQ2 causes energy to be printed as a function of surface. FQ22 causes surface to be printed as a function of energy. FQ62 prints multiplier bins as a function of energy for the two surfaces desired. In tally 22 the letter g is defined to be the equivalent of character string (1 2 3 4 5 6), which is too long for a label. The NT and T features also are illustrated in tallies 62 and 22, respectively. The generalized FM62 card used with the F62:N tally is a useful feature for normalization, unit conversion, reaction rate, etc., and has three multipliers instead of one. Finally, note that the TF62 card causes the tally fluctuation chart for the second surface, the first multiplier bin, and the second energy bin to be printed. By default the fluctuation chart for tally 62 would contain information for the first surface, the first multiplier bin, and the last energy bin.

```
mcnp     version 3a     ld=04/16/86          9406cc14nl  y 04/30/86 08:00:41          probid = y 04/30/86 08:00:41
*****************************************************************************************
inp=demo

    1-       demo: a box with flux across surfaces in various combinations
    2-       1     0  -1 -2 3 4 -5 6
    3-       2     0  #1
    4-
    5-       1     py  1
    6-       2     pz  1
    7-       3     py -1
    8-       4     pz -1
    9-       5     px  1
   10-       6     px -1
   11-
   12-       sdef  pos=0 0 0 cel=1 wgt=1 erg=d1
   13-       si1    .1 1
   14-       sp1   0 1
   15-       imp:n 1 0
   16-       e0     .2 .4 .6 .8 1
   17-       f2:n  1 2 3 4 5 6
   18-       fq2   e f
   19-       f12:n (1 2) (3 4) (5 6)
   20-       f22:n 1 2 3 4 5 6 t
   21-       fq22  f e
   22-       m1    6000.50c  1
warning.  material   1 is used only for tallying.
   23-       m2    74000.55c 1
warning.  material   2 is used only for tallying.
   24-       f62:n 1 3
   25-       fm62  (1 2(1 -4)(-2))(1 1 1)
   26-       e62    .2 .4 .6 .8 1 nt
   27-       fq62  m e
   28-       tf62  2 5j 2
   29-       nps   104000
   30-
```

```
tally   2        nps =    104000
          neutron tally type 2:   particle flux averaged over a surface.       units   1/cm**2

          areas
              surface:         1              2              3              4              5              6
                         4.00000e+00    4.00000e+00    4.00000e+00    4.00000e+00    4.00000e+00    4.00000e+00

      surface:             1                         2                         3                         4                         5
      energy
    2.0000e-01     5.70327e-03 0.0226      5.89204e-03 0.0227      5.81796e-03 0.0225      5.61231e-03 0.0228      5.69858e-03 0.0227
    4.0000e-01     1.15868e-02 0.0157      1.13071e-02 0.0159      1.11485e-02 0.0160      1.13627e-02 0.0159      1.11244e-02 0.0161
    6.0000e-01     1.10237e-02 0.0162      1.11704e-02 0.0160      1.11234e-02 0.0161      1.14028e-02 0.0159      1.12232e-02 0.0160
    8.0000e-01     1.13520e-02 0.0159      1.11780e-02 0.0160      1.15286e-02 0.0158      1.13603e-02 0.0159      1.12088e-02 0.0160
    1.0000e+00     1.13741e-02 0.0159      1.13787e-02 0.0159      1.12370e-02 0.0160      1.12680e-02 0.0160      1.11077e-02 0.0161
      total        5.10398e-02 0.0070      5.07282e-02 0.0070      5.08555e-02 0.0070      5.10061e-02 0.0070      5.03627e-02 0.0070

      surface:             6
      energy
    2.0000e-01     5.67523e-03 0.0227
    4.0000e-01     1.13847e-02 0.0159
    6.0000e-01     1.14088e-02 0.0159
    8.0000e-01     1.13862e-02 0.0159
    1.0000e+00     1.14430e-02 0.0158
      total        5.12979e-02 0.0070

 for the tally-fluctuation-chart bin:
 average tally per history = 2.04159e-01        largest tally = 1.72511e+00
 largest/average = 8.44983e+00                  nps of largest =     67465

tally  12        nps =    104000
          neutron tally type 2:   particle flux averaged over a surface.       units   1/cm**2

          areas
              surface:       (1 2)          (3 4)          (5 6)
                         8.00000e+00    8.00000e+00    8.00000e+00

 surface (1 2)
         energy
    2.0000e-01     5.69765e-03 0.0159
    4.0000e-01     1.14469e-02 0.0110
    6.0000e-01     1.10970e-02 0.0112
    8.0000e-01     1.12650e-02 0.0111
    1.0000e+00     1.13764e-02 0.0110
      total        5.08830e-02 0.0044

 surface (3 4)
         energy
    2.0000e-01     5.71514e-03 0.0159
    4.0000e-01     1.12558e-02 0.0111
    6.0000e-01     1.12631e-02 0.0111
    8.0000e-01     1.14444e-02 0.0110
    1.0000e+00     1.12525e-02 0.0111
      total        5.09308e-02 0.0044

 surface (5 8)
         energy
    2.0000e-01     5.68691e-03 0.0159
    4.0000e-01     1.12545e-02 0.0111
    6.0000e-01     1.13160e-02 0.0111
    8.0000e-01     1.12975e-02 0.0111
    1.0000e+00     1.12754e-02 0.0111
      total        5.08303e-02 0.0044

 for the tally-fluctuation-chart bin:
 average tally per history = 4.07064e-01        largest tally = 1.72662e+00
 largest/average = 4.24165e+00                  nps of largest =     29676
```

```
tally  22          nps =    104000
          neutron tally type 2:   particle flux averaged over a surface.        units   1/cm**2
          surface  g is (1 2 3 4 5 6)

          areas
              surface:        1              2              3              4              5              6              g
                       4.00000e+00    4.00000e+00    4.00000e+00    4.00000e+00    4.00000e+00    4.00000e+00    2.40000e+01

     energy:       2.0000e-01              4.0000e-01              6.0000e-01              8.0000e-01              1.0000e+00
   surface
      1         5.70327e-03 0.0226    1.15868e-02 0.0157    1.10237e-02 0.0162    1.13520e-02 0.0159    1.13741e-02 0.0159
      2         5.69204e-03 0.0227    1.13071e-02 0.0159    1.11704e-02 0.0160    1.11780e-02 0.0160    1.13787e-02 0.0159
      3         5.81796e-03 0.0225    1.11485e-02 0.0160    1.11234e-02 0.0161    1.15286e-02 0.0158    1.12370e-02 0.0160
      4         5.61231e-03 0.0226    1.13627e-02 0.0159    1.14028e-02 0.0159    1.13603e-02 0.0159    1.12680e-02 0.0160
      5         5.69858e-03 0.0227    1.11244e-02 0.0161    1.12232e-02 0.0160    1.12088e-02 0.0160    1.11077e-02 0.0161
      6         5.67523e-03 0.0227    1.13847e-02 0.0159    1.14088e-02 0.0159    1.13862e-02 0.0159    1.14430e-02 0.0158
      g         5.69990e-03 0.0068    1.13190e-02 0.0059    1.12254e-02 0.0059    1.13356e-02 0.0059    1.13014e-02 0.0059

     energy:         total
   surface
      1         5.10398e-02 0.0070
      2         5.07262e-02 0.0070
      3         5.08555e-02 0.0070
      4         5.10061e-02 0.0070
      5         5.03627e-02 0.0070
      6         5.12979e-02 0.0070
      g         5.08814e-02 0.0004

 for the tally-fluctuation-chart bin:
 average tally per history = 2.04159e-01         largest tally = 1.72511e+00
 largest/average = 8.44983e+00                   nps of largest =     67465

tally  62          nps =    104000
          neutron tally type 2:   particle flux averaged over a surface.
          multiplier bin  1:   1.00000e+00    2          1       -4
          multiplier bin  2:   1.00000e+00    2         -2
          multiplier bin  3:   1.00000e+00    1          1

          areas
              surface:        1              3
                       4.00000e+00    4.00000e+00

 surface  1
     energy:       2.0000e-01              4.0000e-01              6.0000e-01              8.0000e-01              1.0000e+00
   mult bin
      1         7.14479e-05 0.0228    2.28961e-04 0.0159    3.13574e-04 0.0162    4.17174e-04 0.0159    5.21860e-04 0.0159
      2         7.22534e-04 0.0232    8.28765e-04 0.0158    6.60869e-04 0.0162    6.93075e-04 0.0159    7.24504e-04 0.0159
      3         2.42866e-02 0.0226    4.46971e-02 0.0158    3.75746e-02 0.0162    3.43937e-02 0.0159    3.09242e-02 0.0159

 surface  3
     energy:       2.0000e-01              4.0000e-01              6.0000e-01              8.0000e-01              1.0000e+00
   mult bin
      1         7.24898e-05 0.0227    2.19258e-04 0.0162    3.15736e-04 0.0161    4.22173e-04 0.0158    5.15286e-04 0.0160
      2         7.44332e-04 0.0230    8.00197e-04 0.0162    6.67135e-04 0.0161    7.03265e-04 0.0158    7.15717e-04 0.0160
      3         2.47957e-02 0.0225    4.30660e-02 0.0161    3.79504e-02 0.0161    3.49961e-02 0.0158    3.05628e-02 0.0160

 for the tally-fluctuation-chart bin:
 average tally per history = 8.77031e-04         largest tally = 4.04150e-02
 largest/average = 4.60816e+01                   nps of largest =     89977
```

tally fluctuation charts

	tally 2			tally 12			tally 22		
nps	mean	error	fom	mean	error	fom	mean	error	fom
8000	5.19490e-02	0.0249	38302	5.09259e-02	0.0180	92856	5.19490e-02	0.0249	38302
16000	5.06051e-02	0.0179	37117	5.02708e-02	0.0114	91300	5.06051e-02	0.0179	37117
24000	5.07911e-02	0.0146	37259	5.05497e-02	0.0093	92079	5.07911e-02	0.0146	37259
32000	5.07186e-02	0.0126	37202	5.06438e-02	0.0080	92358	5.07186e-02	0.0126	37202
40000	5.11064e-02	0.0112	37444	5.06609e-02	0.0072	92162	5.11064e-02	0.0112	37444
48000	5.11324e-02	0.0103	37474	5.09819e-02	0.0065	93133	5.11324e-02	0.0103	37474
56000	5.09241e-02	0.0095	37310	5.09112e-02	0.0060	92970	5.09241e-02	0.0095	37310
64000	5.08531e-02	0.0089	37229	5.08429e-02	0.0057	92728	5.08531e-02	0.0089	37229
72000	5.06920e-02	0.0084	37101	5.07082e-02	0.0053	92361	5.06920e-02	0.0084	37101
80000	5.10515e-02	0.0080	37367	5.08491e-02	0.0051	92816	5.10515e-02	0.0080	37367
88000	5.12325e-02	0.0076	37511	5.09264e-02	0.0048	92782	5.12325e-02	0.0076	37511
96000	5.11367e-02	0.0073	37454	5.08526e-02	0.0046	92621	5.11367e-02	0.0073	37454
104000	5.10398e-02	0.0070	37372	5.08830e-02	0.0044	92705	5.10398e-02	0.0070	37372

	tally 62		
nps	mean	error	fom
8000	2.04009e-04	0.0605	6468
16000	2.20747e-04	0.0412	6971
24000	2.17390e-04	0.0339	6884
32000	2.18704e-04	0.0293	6922
40000	2.20101e-04	0.0261	6952
48000	2.17706e-04	0.0239	6885
56000	2.19005e-04	0.0221	6949
64000	2.21268e-04	0.0205	7014
72000	2.20577e-04	0.0194	6996
80000	2.20094e-04	0.0184	6968
88000	2.18824e-04	0.0176	6936
96000	2.20139e-04	0.0168	6982
104000	2.19258e-04	0.0162	6959

```
*******************************************************************************
dump no.   2 on file   runtpe          nps =   104000          ctm =    0.55

    3 warning messages so far.
```

II. TEST1 PROBLEM AND OUTPUT

TEST1 defines a disk of concrete 100 cm thick, having a 75-cm radius. A 14.19 MeV neutron source is incident at a point in the center of a face of the disk and normal to it. Several neutron and photon tallies are made on surface 18 and in cell 17. There is no energy cutoff and the simple physics treatment that includes implicit capture is used for photons with energy greater than 0.001 MeV.

The disk is divided into 16 slabs, each 6.25 cm thick, as seen in Fig. 5.1. The neutron importance of each slab, or cell, varies from 0 in cell 1 to 32 in cell 17. Photon importances are set equal to neutron importances. The problem ran 10,000 particles and the tally means, errors, and FOMs shown in the tally fluctuation charts seem to be stable.

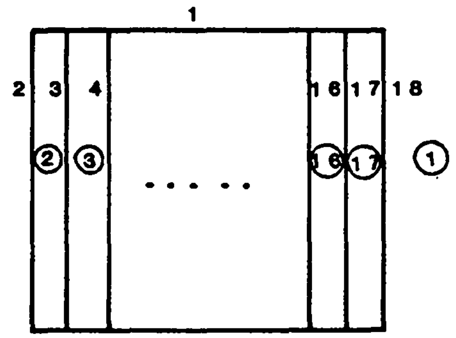


Figure 5.1

The weight window generator was used to generate a better importance function for subsequent runs. The resulting cards are printed at the end of the TEST1 output file and can be copied into an input file to be run a second time. Generation of weight windows did not affect the results of TEST1 but did slow down the calculation by 16%. When the importances in TEST1 were replaced by the generated weight windows (WWP and WWN cards), the problem took 10 minutes to run 9639 particles vs 8.74 minutes for 10,000 particles in TEST1. However, the photon FOMs increased by a factor of 2 to 3 and the errors decreased by half, while the means appeared to stay stable. The neutron means, errors, and FOMs stayed approximately the same, indicating that they were already well chosen to optimize tally 12.

Following is the complete output from TEST1. If a page has the symbol X on it, that page will not appear unless the PRINT option or card has been used. If $\mathcal{N}$n, where n is an integer, appears before an item on a page, that item is explained or discussed in Note $\mathcal{N}$n in the text following the output.

```
mcnp     version 3a    ld=04/16/86        9406cc14n1  y 04/30/86 11:23:49
**********************************************************************          probid = y 04/30/86 11:23:49
inp=test1
    1-        test1:  100 cm thick concrete disk with 15 splitting surfaces
    2-        c
    3-         1       0             1 : -2 : 18
    4-         2       1 -2.2505   -1   -3    2
    5-         3       1 -2.2505   -1   -4    3
    6-         4       1 -2.2505   -1   -5    4
    7-         5       1 -2.2505   -1   -6    5
    8-         6       1 -2.2505   -1   -7    6
    9-         7       1 -2.2505   -1   -8    7
   10-         8       1 -2.2505   -1   -9    8
   11-         9       1 -2.2505   -1   -10   9
   12-        10       1 -2.2505   -1   -11   10
   13-        11       1 -2.2505   -1   -12   11
   14-        12       1 -2.2505   -1   -13   12
   15-        13       1 -2.2505   -1   -14   13
   16-        14       1 -2.2505   -1   -15   14
   17-        15       1 -2.2505   -1   -16   15
   18-        16       1 -2.2505   -1   -17   16
   19-        17       1 -2.2505   -1   -18   17
   20-
   21-         1       cy  75
   22-         2       py  0
   23-         3       py  6.25
   24-         4       py  12.50
   25-         5       py  18.75
   26-         6       py  25.00
   27-         7       py  31.25
   28-         8       py  37.50
   29-         9       py  43.75
   30-        10       py  50.00
   31-        11       py  56.25
   32-        12       py  62.50
   33-        13       py  68.75
   34-        14       py  75.00
   35-        15       py  81.25
   36-        16       py  87.50
   37-        17       py  93.75
   38-        18       py  100.00
   39-
   40-        mode     n p
   41-        c          the following is los alamos concrete
   42-        m1        1001.50 -.004532
   43-                  8016.50 -.512597
   44-                 11023.51 -.011553
   45-                 12000.51 -.003866
   46-                 13027.50 -.035548
   47-                 14000.51 -.360364
   48-                 19000.51 -.014219
   49-                 20000.51 -.043546
   50-                 26000.55 -.013775
   51-        sdef     pos=0 0 0 cel=2 wgt=1 vec=0 1 0 sur=2 dir=1 erg=14.19
   52-        imp:n 0  1 5r 2 2 2 4 4 4 8 8 16 32
   53-        imp:p 0  1 5r 2 2 2 4 4 4 8 8 16 32
   54-        pwt   0 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1.7 -1.4 -1.0 -0.7 -0.4 -0.3 -0.2
   55-        f1:p     18
   56-        f11:n    18
```

```
fc12     optimize weight window generator on tally 12
f12:p    18
e12      20
wwg      12 2 .5 0 1e8 0
f8:n,p   17
e8       .00001 .0001 .001 .01 .05 .1 .5 1 13i 15 20
f18:n    17
f28:p    17
f34:n    17
fm34     -1 1 1 -4
e0       .0001 .001 .01 .05 .1 .5 1 13i 15 20
phys:n    15  0
phys:p    .001
nps      10000
print
ctme     10
```

.17 values of defaulted or explicitly defined source variables

cel	2.0000e+00		
sur	2.0000e+00		
erg	1.4190e+01		
tme	0.0000e+00		
dir	1.0000e+00		
pos	0.0000e+00	0.0000e+00	0.0000e+00
x	0.0000e+00		
y	0.0000e+00		
z	0.0000e+00		
rad	0.0000e+00		
ext	0.0000e+00		
axs	0.0000e+00	0.0000e+00	0.0000e+00
vec	0.0000e+00	1.0000e+00	0.0000e+00
ccc	0.0000e+00		
nrm	1.0000e+00		
ara	1.0000e+00		
wgt	1.0000e+00		
eff	1.0000e-02		

order of sampling source variables.
cel sur pos vec dir erg tme

```
      tally  11                                                                          print table 30
18                 neutron tally type 1:   number of particles crossing a surface.

.19   warning.  energy bin limits adjusted for tally  11

      surfaces       18

.110  energy bins
          0.00000e+00  to  1.00000e-04 mev
          1.00000e-04  to  1.00000e-03 mev
          1.00000e-03  to  1.00000e-02 mev
          1.00000e-02  to  5.00000e-02 mev
          5.00000e-02  to  1.00000e-01 mev
          1.00000e-01  to  5.00000e-01 mev
          5.00000e-01  to  1.00000e+00 mev
          1.00000e+00  to  2.00000e+00 mev
          2.00000e+00  to  3.00000e+00 mev
          3.00000e+00  to  4.00000e+00 mev
          4.00000e+00  to  5.00000e+00 mev
          5.00000e+00  to  6.00000e+00 mev
          6.00000e+00  to  7.00000e+00 mev
          7.00000e+00  to  8.00000e+00 mev
          8.00000e+00  to  9.00000e+00 mev
          9.00000e+00  to  1.00000e+01 mev
          1.00000e+01  to  1.10000e+01 mev
          1.10000e+01  to  1.20000e+01 mev
          1.20000e+01  to  1.30000e+01 mev
          1.30000e+01  to  1.40000e+01 mev
          1.40000e+01  to  1.50000e+01 mev
          total bin
```

```
tally  16                                                                      print table 30
          neutron tally type 6:   track length estimate of heating.

cells        17

energy bins
     0.00000e+00  to  1.00000e-04 mev
     1.00000e-04  to  1.00000e-03 mev
     1.00000e-03  to  1.00000e-02 mev
     1.00000e-02  to  5.00000e-02 mev
     5.00000e-02  to  1.00000e-01 mev
     1.00000e-01  to  5.00000e-01 mev
     5.00000e-01  to  1.00000e+00 mev
     1.00000e+00  to  2.00000e+00 mev
     2.00000e+00  to  3.00000e+00 mev
     3.00000e+00  to  4.00000e+00 mev
     4.00000e+00  to  5.00000e+00 mev
     5.00000e+00  to  6.00000e+00 mev
     6.00000e+00  to  7.00000e+00 mev
     7.00000e+00  to  8.00000e+00 mev
     8.00000e+00  to  9.00000e+00 mev
     9.00000e+00  to  1.00000e+01 mev
     1.00000e+01  to  1.10000e+01 mev
     1.10000e+01  to  1.20000e+01 mev
     1.20000e+01  to  1.30000e+01 mev
     1.30000e+01  to  1.40000e+01 mev
     1.40000e+01  to  1.50000e+01 mev
     total bin
```

```
tally  34                                                                                       print table 30
          neutron tally type 4:   track length estimate of particle flux.

cells        17

multiplier bins
att  constant    material    reactions or material-rho*x pairs
   -1.00000e+00       1

energy bins
      0.00000e+00   to   1.00000e-04 mev
      1.00000e-04   to   1.00000e-03 mev
      1.00000e-03   to   1.00000e-02 mev
      1.00000e-02   to   5.00000e-02 mev
      5.00000e-02   to   1.00000e-01 mev
      1.00000e-01   to   5.00000e-01 mev
      5.00000e-01   to   1.00000e+00 mev
      1.00000e+00   to   2.00000e+00 mev
      2.00000e+00   to   3.00000e+00 mev
      3.00000e+00   to   4.00000e+00 mev
      4.00000e+00   to   5.00000e+00 mev
      5.00000e+00   to   6.00000e+00 mev
      6.00000e+00   to   7.00000e+00 mev
      7.00000e+00   to   8.00000e+00 mev
      8.00000e+00   to   9.00000e+00 mev
      9.00000e+00   to   1.00000e+01 mev
      1.00000e+01   to   1.10000e+01 mev
      1.10000e+01   to   1.20000e+01 mev
      1.20000e+01   to   1.30000e+01 mev
      1.30000e+01   to   1.40000e+01 mev
      1.40000e+01   to   1.50000e+01 mev
      total bin
```

```
tally    1                                                                                   print table 30
                photon tally type 1:   number of particles crossing a surface.

[illegible]  warning.     2 energy bins of tally   1 are below energy cutoff.

      surfaces       18

      energy bins
         1.00000e-03  to  1.00000e-04 mev
         1.00000e-04  to  1.00000e-03 mev
         1.00000e-03  to  1.00000e-02 mev
         1.00000e-02  to  5.00000e-02 mev
         5.00000e-02  to  1.00000e-01 mev
         1.00000e-01  to  5.00000e-01 mev
         5.00000e-01  to  1.00000e+00 mev
         1.00000e+00  to  2.00000e+00 mev
         2.00000e+00  to  3.00000e+00 mev
         3.00000e+00  to  4.00000e+00 mev
         4.00000e+00  to  5.00000e+00 mev
         5.00000e+00  to  6.00000e+00 mev
         6.00000e+00  to  7.00000e+00 mev
         7.00000e+00  to  8.00000e+00 mev
         8.00000e+00  to  9.00000e+00 mev
         9.00000e+00  to  1.00000e+01 mev
         1.00000e+01  to  1.10000e+01 mev
         1.10000e+01  to  1.20000e+01 mev
         1.20000e+01  to  1.30000e+01 mev
         1.30000e+01  to  1.40000e+01 mev
         1.40000e+01  to  1.50000e+01 mev
         1.50000e+01  to  2.00000e+01 mev
         total bin
```

-325-

```
tally  12
             photon tally type 2:   optimize weight window generator on tally 12          print table 30
                                    particle flux averaged over a surface.
surfaces       18

energy bins
     1.00000e-03  to  2.00000e+01 mev
```

```
tally  26                                                                                        print table 30
          photon tally type 6:   track length estimate of heating.

cells          17

energy bins
     1.00000e-03  to  1.00000e-04 mev
     1.00000e-04  to  1.00000e-03 mev
     1.00000e-03  to  1.00000e-02 mev
     1.00000e-02  to  5.00000e-02 mev
     5.00000e-02  to  1.00000e-01 mev
     1.00000e-01  to  5.00000e-01 mev
     5.00000e-01  to  1.00000e+00 mev
     1.00000e+00  to  2.00000e+00 mev
     2.00000e+00  to  3.00000e+00 mev
     3.00000e+00  to  4.00000e+00 mev
     4.00000e+00  to  5.00000e+00 mev
     5.00000e+00  to  6.00000e+00 mev
     6.00000e+00  to  7.00000e+00 mev
     7.00000e+00  to  8.00000e+00 mev
     8.00000e+00  to  9.00000e+00 mev
     9.00000e+00  to  1.00000e+01 mev
     1.00000e+01  to  1.10000e+01 mev
     1.10000e+01  to  1.20000e+01 mev
     1.20000e+01  to  1.30000e+01 mev
     1.30000e+01  to  1.40000e+01 mev
     1.40000e+01  to  1.50000e+01 mev
     1.50000e+01  to  2.00000e+01 mev
     total bin
```

```
tally   6                                                                                            print table 30
        combined tally type 6:   track length estimate of heating.

cells        17

energy bins
     0.00000e+00  to  1.00000e-05 mev
     1.00000e-05  to  1.00000e-04 mev
     1.00000e-04  to  1.00000e-03 mev
     1.00000e-03  to  1.00000e-02 mev
     1.00000e-02  to  5.00000e-02 mev
     5.00000e-02  to  1.00000e-01 mev
     1.00000e-01  to  5.00000e-01 mev
     5.00000e-01  to  1.00000e+00 mev
     1.00000e+00  to  2.00000e+00 mev
     2.00000e+00  to  3.00000e+00 mev
     3.00000e+00  to  4.00000e+00 mev
     4.00000e+00  to  5.00000e+00 mev
     5.00000e+00  to  6.00000e+00 mev
     6.00000e+00  to  7.00000e+00 mev
     7.00000e+00  to  8.00000e+00 mev
     8.00000e+00  to  9.00000e+00 mev
     9.00000e+00  to  1.00000e+01 mev
     1.00000e+01  to  1.10000e+01 mev
     1.10000e+01  to  1.20000e+01 mev
     1.20000e+01  to  1.30000e+01 mev
     1.30000e+01  to  1.40000e+01 mev
     1.40000e+01  to  1.50000e+01 mev
     1.50000e+01  to  2.00000e+01 mev
     total bin
```

material composition

print table 40

material number	component nuclide, atom fraction							
1	1001.50	,0.08476	8016.50	,0.60408	11023.51	,0.00947	12000.51	,0.00300
	13027.50	,0.02483	14000.51	,0.24187	19000.51	,0.00685	20000.51	,0.02048
	26000.55	,0.00465						

material number	component nuclide, mass fraction							
1	1001.50	,0.00453	8016.50	,0.51260	11023.51	,0.01155	12000.51	,0.00387
	13027.50	,0.03555	14000.51	,0.36036	19000.51	,0.01422	20000.51	,0.04355
	26000.55	,0.01378						

cell volumes and masses

print table 50

progr name	probl name	atom density	gram density	input volume	calculated volume	mass	pieces	reason volume not calculated
1	1	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0	infinite
2	2	7.18969e-02	2.25050e+00	0.00000e+00	1.10447e+05	2.48560e+05	1	
3	3	7.18969e-02	2.25050e+00	0.00000e+00	1.10447e+05	2.48560e+05	1	
4	4	7.18969e-02	2.25050e+00	0.00000e+00	1.10447e+05	2.48560e+05	1	
5	5	7.18969e-02	2.25050e+00	0.00000e+00	1.10447e+05	2.48560e+05	1	
6	6	7.18969e-02	2.25050e+00	0.00000e+00	1.10447e+05	2.48560e+05	1	
7	7	7.18969e-02	2.25050e+00	0.00000e+00	1.10447e+05	2.48560e+05	1	
8	8	7.18969e-02	2.25050e+00	0.00000e+00	1.10447e+05	2.48560e+05	1	
9	9	7.18969e-02	2.25050e+00	0.00000e+00	1.10447e+05	2.48560e+05	1	
10	10	7.18969e-02	2.25050e+00	0.00000e+00	1.10447e+05	2.48560e+05	1	
11	11	7.18969e-02	2.25050e+00	0.00000e+00	1.10447e+05	2.48560e+05	1	
12	12	7.18969e-02	2.25050e+00	0.00000e+00	1.10447e+05	2.48560e+05	1	
13	13	7.18969e-02	2.25050e+00	0.00000e+00	1.10447e+05	2.48560e+05	1	
14	14	7.18969e-02	2.25050e+00	0.00000e+00	1.10447e+05	2.48560e+05	1	
15	15	7.18969e-02	2.25050e+00	0.00000e+00	1.10447e+05	2.48560e+05	1	
16	16	7.18969e-02	2.25050e+00	0.00000e+00	1.10447e+05	2.48560e+05	1	
17	17	7.18969e-02	2.25050e+00	0.00000e+00	1.10447e+05	2.48560e+05	1	

N12

surface areas

print table 50

progr name	probl name	input area	calculated area	reason area not calculated
1	1	0.00000e+00	4.71239e+04	
2	2	0.00000e+00	1.76715e+04	
3	3	0.00000e+00	1.76715e+04	
4	4	0.00000e+00	1.76715e+04	
5	5	0.00000e+00	1.76715e+04	
6	6	0.00000e+00	1.76715e+04	
7	7	0.00000e+00	1.76715e+04	
8	8	0.00000e+00	1.76715e+04	
9	9	0.00000e+00	1.76715e+04	
10	10	0.00000e+00	1.76715e+04	
11	11	0.00000e+00	1.76715e+04	
12	12	0.00000e+00	1.76715e+04	
13	13	0.00000e+00	1.76715e+04	
14	14	0.00000e+00	1.76715e+04	
15	15	0.00000e+00	1.76715e+04	
16	16	0.00000e+00	1.76715e+04	
17	17	0.00000e+00	1.76715e+04	
18	18	0.00000e+00	1.76715e+04	

cells

print table 60

	cell	mat	atom density	gram density	volume	mass	pieces	neutron importance	photon importance	photon wt generation
1	1	0	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0	0.00000e+00	0.00000e+00	0.00000e+00
2	2	1	7.18969e-02	2.25050e+00	1.10447e+05	2.48560e+05	1	1.00000e+00	1.00000e+00	-1.00000e+01
3	3	1	7.18969e-02	2.25050e+00	1.10447e+05	2.48560e+05	1	1.00000e+00	1.00000e+00	-9.00000e+00
4	4	1	7.18969e-02	2.25050e+00	1.10447e+05	2.48560e+05	1	1.00000e+00	1.00000e+00	-8.00000e+00
5	5	1	7.18969e-02	2.25050e+00	1.10447e+05	2.48560e+05	1	1.00000e+00	1.00000e+00	-7.00000e+00
6	6	1	7.18969e-02	2.25050e+00	1.10447e+05	2.48560e+05	1	1.00000e+00	1.00000e+00	-6.00000e+00
7	7	1	7.18969e-02	2.25050e+00	1.10447e+05	2.48560e+05	1	1.00000e+00	1.00000e+00	-5.00000e+00
8	8	1	7.18969e-02	2.25050e+00	1.10447e+05	2.48560e+05	1	2.00000e+00	2.00000e+00	-4.00000e+00
9	9	1	7.18969e-02	2.25050e+00	1.10447e+05	2.48560e+05	1	2.00000e+00	2.00000e+00	-3.00000e+00
10	10	1	7.18969e-02	2.25050e+00	1.10447e+05	2.48560e+05	1	2.00000e+00	2.00000e+00	-2.00000e+00
11	11	1	7.18969e-02	2.25050e+00	1.10447e+05	2.48560e+05	1	4.00000e+00	4.00000e+00	-1.70000e+00
12	12	1	7.18969e-02	2.25050e+00	1.10447e+05	2.48560e+05	1	4.00000e+00	4.00000e+00	-1.40000e+00
13	13	1	7.18969e-02	2.25050e+00	1.10447e+05	2.48560e+05	1	4.00000e+00	4.00000e+00	-1.00000e+00
14	14	1	7.18969e-02	2.25050e+00	1.10447e+05	2.48560e+05	1	8.00000e+00	8.00000e+00	-7.00000e-01
15	15	1	7.18969e-02	2.25050e+00	1.10447e+05	2.48560e+05	1	8.00000e+00	8.00000e+00	-4.00000e-01
16	16	1	7.18969e-02	2.25050e+00	1.10447e+05	2.48560e+05	1	1.60000e+01	1.60000e+01	-3.00000e-01
17	17	1	7.18969e-02	2.25050e+00	1.10447e+05	2.48560e+05	1	3.20000e+01	3.20000e+01	-2.00000e-01
A13 total					1.76715e+06	3.97696e+06				

18 surfaces

print table 70

.414	progr name	probl name	trans	type	surface coefficients
	1	1		cy	7.5000000e+01
	2	2		py	0.0000000e+00
	3	3		py	6.2500000e+00
	4	4		py	1.2500000e+01
	5	5		py	1.8750000e+01
	6	6		py	2.5000000e+01
	7	7		py	3.1250000e+01
	8	8		py	3.7500000e+01
	9	9		py	4.3750000e+01
	10	10		py	5.0000000e+01
	11	11		py	5.6250000e+01
	12	12		py	6.2500000e+01
	13	13		py	6.8750000e+01
	14	14		py	7.5000000e+01
	15	15		py	8.1250000e+01
	16	16		py	8.7500000e+01
	17	17		py	9.3750000e+01
	18	18		py	1.0000000e+02

print table 72

115 temperatures of the cells.

time: 0.0000e+00

cell	tmp 1
2	2.5300e-08
3	2.5300e-08
4	2.5300e-08
5	2.5300e-08
6	2.5300e-08
7	2.5300e-08
8	2.5300e-08
9	2.5300e-08
10	2.5300e-08
11	2.5300e-08
12	2.5300e-08
13	2.5300e-08
14	2.5300e-08
15	2.5300e-08
16	2.5300e-08
17	2.5300e-08

116 minimum source weight = 1.0000e+00 maximum source weight = 1.0000e+00

2 warning messages so far.

```
117   cross-section tables                                                                              print table 100
          table     length
                                  tables from file rmccs

       1001.50c      1851
       8016.50c     30229
      11023.51c     46156
      12000.51c     44732
      13027.50c     46655
      14000.51c     82689
      19000.51c     15662
      20000.51c     45036
      26000.55c    141728
                                  tables from file mcplib

       1000.01p       389
       8000.01p       389
      11000.01p       401
      12000.01p       409
      13000.01p       409
      14000.01p       409
      19000.01p       409
      20000.01p       417
      26000.01p       417

      total        458387
N18   any neutrons with energy greater than emax = 1.50000e+01 from the source or from a collision will be resampled.
N19   neutron cross sections outside the range from 0.0000e+00 to 1.5000e+01 mev are expunged.
N20   warning.  neutron energy cutoff is below some cross-section tables.


      use of dynamically allocated storage

          general                10082
          tallies                  411
          cross sections        458387

          total                 468880

      ***************************************************************************
N21   dump no.  1 on file   runtpe           nps =        0           ctm =   0.00

              3 warning messages so far.
```

starting mcrun.   field length = 648468   cp0 = 0.13   print table 110

test1: 100 cm thick concrete disk with 15 splitting surfaces

nps	x	y	z	cell	surf	u	v	w	time	weight	energy
1	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
2	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
3	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
4	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
5	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
6	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
7	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
8	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
9	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
10	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
11	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
12	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
13	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
14	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
15	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
16	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
17	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
18	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
19	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
20	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
21	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
22	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
23	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
24	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
25	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
26	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
27	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
28	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
29	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
30	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
31	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
32	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
33	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
34	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
35	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
36	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
37	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
38	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
39	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
40	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
41	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
42	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
43	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
44	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
45	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
46	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
47	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
48	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
49	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01
50	0.000e+00	0.000e+00	0.000e+00	2	2	0.000e+00	1.000e+00	0.000e+00	0.000e+00	1.000e+00	1.419e+01

```
     run terminated when     10000 particle histories were done.                                          y 04/30/86 11:54:18
     test1:  100 cm thick concrete disk with 15 splitting surfaces                                probid = y 04/30/86 11:23:49
N23                        ledger of net neutron creation and loss (for accounting only)

                      tracks      weight        energy                                   tracks      weight        energy
                                  (per source particle)                                              (per source particle)

     source            10000    1.0000e+00    1.4190e+01   N26  escape                 13575    4.1407e-01    9.9847e-01
     scattering            0    0.            1.1659e-07   N27  scattering                 0    0.            8.7732e+00
     fission               0    0.            0.           N28  capture                    0    5.9746e-01    4.4429e+00
     (n,xn)              271    1.5857e-02    3.8012e-02        energy cutoff              0    0.            0.
     forced collision      0    0.            0.                time cutoff                0    0.            0.
N24  weight cutoff         0    8.0713e-02    3.8588e-02   N29  weight cutoff           8528    8.1090e-02    4.4977e-02
     weight window         0    0.            0.                weight window              0    0.            0.
N25  cell importance   36474    2.6982e-01    9.8527e-02   N30  cell importance        24642    2.7377e-01    1.0562e-01
     energy import.        0    0.            0.                energy import.             0    0.            0.
     dxtran                0    0.            0.                dxtran                     0    0.            0.
     exp. transform        0    0.            0.                exp. transform             0    0.            0.
                                                                dead fission               0    0.            0.
        total          46745    1.3664e+00    1.4365e+01   N31  total                  46745    1.3664e+00    1.4365e+01

                                               N32    average lifetime, shakes              cutoffs
       neutron tracks per source particle     4.6745e+00   escape             6.3329e+03      tco    1.0000+120
       neutron collisions per source particle 1.1687e+02   capture            1.6003e+04      eco    0.0000e+00
       total neutron collisions                 1168723    capture or escape  1.3226e+04      wc1   -5.0000e-01
 N33   net multiplication             1.0159e+00 0.0010    any termination    1.8736e+04      wc2   -2.5000e-01

                           ledger of net photon  creation and loss (for accounting only)

                      tracks      weight        energy                                   tracks      weight        energy
                                  (per source particle)                                              (per source particle)

     source                0    0.            0.                escape                  6147    4.5643e-01    8.9965e-01
N34  p-annihilation        0    1.2575e-01    0.                fluorescence               0    0.            0.
     from neutrons     14156    1.9007e+00    5.6815e+00        capture                    0    1.5581e+00    1.2052e-01
     2x fluorescence       0    0.            0.                energy cutoff              0    0.            0.
     forced collision      0    0.            0.                time cutoff                0    0.            0.
     weight cutoff         0    3.4476e-02    4.2340e-03        weight cutoff          11240    3.6768e-02    4.2111e-03
     weight window         0    0.            0.                weight window              0    0.            0.
     cell importance    8575    1.2667e-01    1.3579e-01        cell importance         5344    1.3631e-01    1.4624e-01
     energy import.        0    0.            0.                energy import.             0    0.            0.
     dxtran                0    0.            0.                dxtran                     0    0.            0.
     exp. transform        0    0.            0.                exp. transform             0    0.            0.
                                                                scattering                 0    0.            4.1229e+00
                                                                pair production            0    0.            5.2608e-01
        total          22731    2.1876e+00    5.8216e+00           total               22731    2.1876e+00    5.8216e+00

                                               N35    average lifetime, shakes              cutoffs
       photon tracks per source particle      2.2731e+00   escape             7.6713e+03      tco    1.0000+120
       photon collisions per source particle  1.2711e+01   capture            1.2500e+04      eco    1.0000e-03
       total photon collisions                   127106    capture or escape  1.1406e+04      wc1   -5.0000e-01
                                                           any termination    1.2461e+04      wc2   -2.5000e-01

     computer time so far in this run     8.88 minutes               total neutrons banked             36745
     computer time in mcrun (4c0)         8.74 minutes                  per source particle       3.6745e+00
     source particles per minute            1.1436e+03               total photons banked              22731
     field length                               648468                  per source particle       2.2731e+00
     random numbers generated                 19984567               maximum number ever in bank          21
     last starting random number       241688579059069               bank overflows to backup file         0
     next starting random number       153035362075497

     range of sampled source weights = 1.0000e+00 to 1.0000e+00
```

neutron activity in each cell   print table 126

cell progr name	cell probl name	tracks entering	population	collisions	collisions * weight (per history)	number weighted energy	flux weighted energy	average track weight (relative)	average track mfp (cm)
2	2	18371	10475	43141	2.9070e+00	4.1218e-03	6.8232e+00	7.4199e-01	6.3710e+00
3	3	22030	9141	64977	4.0247e+00	1.7030e-03	4.3054e+00	6.6850e-01	5.4641e+00
4	4	22766	8177	75787	4.4346e+00	9.5810e-04	3.0588e+00	6.2239e-01	4.8829e+00
5	5	21621	7449	77865	4.3368e+00	6.0345e-04	2.2878e+00	5.8703e-01	4.4780e+00
6	6	19439	7195	73448	3.9113e+00	4.0443e-04	1.7723e+00	5.5804e-01	4.1526e+00
7	7	16693	7961	65193	3.3354e+00	2.9482e-04	1.4523e+00	5.3326e-01	3.9315e+00
8	8	27634	13548	110599	2.7469e+00	2.3179e-04	1.1865e+00	5.1552e-01	3.7803e+00
9	9	21960	9150	89226	2.1587e+00	1.7868e-04	9.5858e-01	4.9894e-01	3.6085e+00
10	10	16945	8505	69606	1.6442e+00	1.4681e-04	8.2381e-01	4.8592e-01	3.5083e+00
11	11	25436	12980	106231	1.2196e+00	1.2706e-04	7.3087e-01	4.7256e-01	3.4408e+00
12	12	18449	7930	76392	8.5732e-01	1.1166e-04	6.5640e-01	4.5990e-01	3.3850e+00
13	13	13215	6884	55560	6.1106e-01	1.0058e-04	5.9893e-01	4.4972e-01	3.3353e+00
14	14	18735	10525	79795	4.2994e-01	9.0151e-05	5.3892e-01	4.4051e-01	3.2715e+00
15	15	12696	7266	54637	2.9158e-01	8.2338e-05	4.9328e-01	4.3628e-01	3.2254e+00
16	16	15654	11159	67057	1.7804e-01	8.6521e-05	5.1426e-01	4.3225e-01	3.2623e+00
17	17	14306	11977	59209	7.8115e-02	1.0190e-04	6.0666e-01	4.3116e-01	3.3747e+00
total		305950	150322	1168723	3.3165e+01				
		$\mathcal{N}$36	$\mathcal{N}$37	$\mathcal{N}$38	$\mathcal{N}$39	$\mathcal{N}$40	$\mathcal{N}$41	$\mathcal{N}$42	$\mathcal{N}$43

photon activity in each cell

print table 126

cell progr name	cell probl name	tracks entering	population	collisions	collisions * weight (per history)	number weighted energy	flux weighted energy	average track weight (relative)	average track mfp (cm)
2	2	572	885	2421	1.6864e+00	1.6790e+00	1.6790e+00	8.5156e+00	7.4637e+00
3	3	970	975	3934	2.5670e+00	1.4798e+00	1.4798e+00	7.8959e+00	6.9347e+00
4	4	1094	1002	4175	2.4349e+00	1.4318e+00	1.4316e+00	7.1603e+00	6.7851e+00
5	5	1039	968	4001	2.0991e+00	1.3151e+00	1.3151e+00	6.3433e+00	6.5511e+00
6	6	931	853	3341	1.6062e+00	1.3153e+00	1.3153e+00	5.6401e+00	6.4905e+00
7	7	878	888	2824	1.0971e+00	1.4664e+00	1.4684e+00	4.6521e+00	6.8745e+00
8	8	1775	1812	6071	9.9963e-01	1.3533e+00	1.3533e+00	3.8828e+00	6.6232e+00
9	9	1660	1572	5436	7.4090e-01	1.4244e+00	1.4244e+00	3.1409e+00	6.7883e+00
10	10	1578	1643	5367	5.7896e-01	1.5201e+00	1.5201e+00	2.4720e+00	6.9912e+00
11	11	2926	3043	9675	4.5070e-01	1.4333e+00	1.4333e+00	2.1049e+00	6.8033e+00
12	12	2657	2557	8293	3.0299e-01	1.4571e+00	1.4571e+00	1.6800e+00	6.9239e+00
13	13	2502	2562	7778	2.4850e-01	1.3784e+00	1.3784e+00	1.3743e+00	6.6694e+00
14	14	4516	4740	13820	1.7521e-01	1.3693e+00	1.3693e+00	1.0444e+00	6.6844e+00
15	15	3767	4384	11664	1.1588e-01	1.4056e+00	1.4056e+00	8.0192e-01	6.7758e+00
16	18	5951	7588	17591	7.7171e-02	1.4284e+00	1.4284e+00	6.8625e-01	6.8372e+00
17	17	7554	10271	20719	4.4718e-02	1.4553e+00	1.4553e+00	6.6068e-01	6.9579e+00
total		40370	45743	127108	1.5225e+01				

N4

.115 neutron weight balance in each cell -- external events — print table 130

cell progr	cell probl	source	entering	time cutoff	energy cutoff	exiting	total
2	2	1.0000e+00	5.1855e-01	0.0000e+00	0.0000e+00	-1.4057e+00	1.1284e-01
3	3	0.0000e+00	1.5875e+00	0.0000e+00	0.0000e+00	-1.4916e+00	9.5857e-02
4	4	0.0000e+00	1.4975e+00	0.0000e+00	0.0000e+00	-1.4166e+00	8.0893e-02
5	5	0.0000e+00	1.3212e+00	0.0000e+00	0.0000e+00	-1.2558e+00	6.5387e-02
6	6	0.0000e+00	1.1210e+00	0.0000e+00	0.0000e+00	-1.0671e+00	5.3816e-02
7	7	0.0000e+00	9.2012e-01	0.0000e+00	0.0000e+00	-8.7575e-01	4.4367e-02
8	8	0.0000e+00	7.2906e-01	0.0000e+00	0.0000e+00	-6.9385e-01	3.5210e-02
9	9	0.0000e+00	5.5921e-01	0.0000e+00	0.0000e+00	-5.3146e-01	2.7744e-02
10	10	0.0000e+00	4.2013e-01	0.0000e+00	0.0000e+00	-3.9863e-01	2.1500e-02
11	11	0.0000e+00	3.0500e-01	0.0000e+00	0.0000e+00	-2.8917e-01	1.5827e-02
12	12	0.0000e+00	2.1480e-01	0.0000e+00	0.0000e+00	-2.0366e-01	1.1140e-02
13	13	0.0000e+00	1.5078e-01	0.0000e+00	0.0000e+00	-1.4194e-01	8.8427e-03
14	14	0.0000e+00	1.0454e-01	0.0000e+00	0.0000e+00	-9.9136e-02	5.4081e-03
15	15	0.0000e+00	7.0146e-02	0.0000e+00	0.0000e+00	-6.6327e-02	3.8162e-03
16	16	0.0000e+00	4.2645e-02	0.0000e+00	0.0000e+00	-4.0381e-02	2.2644e-03
17	17	0.0000e+00	1.9486e-02	0.0000e+00	0.0000e+00	-1.8464e-02	1.0219e-03
total		1.0000e+00	9.5816e+00	0.0000e+00	0.0000e+00	-9.9957e+00	5.8593e-01

neutron weight balance in each cell -- physical events

print table 130

cell progr	cell probl	fission loss	fission neutrons	(n,xn)	capture	total
2	2	0.0000e+00	0.0000e+00	5.4689e-03	-1.1893e-01	-1.1346e-01
3	3	0.0000e+00	0.0000e+00	3.8286e-03	-9.9448e-02	-9.5619e-02
4	4	0.0000e+00	0.0000e+00	2.5551e-03	-8.3065e-02	-8.0510e-02
5	5	0.0000e+00	0.0000e+00	1.2684e-03	-6.6545e-02	-6.5276e-02
6	6	0.0000e+00	0.0000e+00	9.8792e-04	-5.4163e-02	-5.3175e-02
7	7	0.0000e+00	0.0000e+00	6.8926e-04	-4.4663e-02	-4.3973e-02
8	8	0.0000e+00	0.0000e+00	4.6461e-04	-3.5907e-02	-3.5443e-02
9	9	0.0000e+00	0.0000e+00	2.0297e-04	-2.7127e-02	-2.6924e-02
10	10	0.0000e+00	0.0000e+00	1.2740e-04	-2.0520e-02	-2.0393e-02
11	11	0.0000e+00	0.0000e+00	1.2898e-04	-1.5542e-02	-1.5413e-02
12	12	0.0000e+00	0.0000e+00	6.9778e-05	-1.1065e-02	-1.0996e-02
13	13	0.0000e+00	0.0000e+00	2.6023e-05	-8.0104e-03	-7.9844e-03
14	14	0.0000e+00	0.0000e+00	5.7811e-06	-5.4223e-03	-5.4165e-03
15	15	0.0000e+00	0.0000e+00	1.0209e-05	-3.6934e-03	-3.6832e-03
16	16	0.0000e+00	0.0000e+00	1.5178e-05	-2.3217e-03	-2.3065e-03
17	17	0.0000e+00	0.0000e+00	7.4875e-06	-1.0374e-03	-1.0300e-03
total		0.0000e+00	0.0000e+00	1.5857e-02	-5.9746e-01	-5.8161e-01

neutron weight balance in each cell -- variance reduction events

print table 130

cell progr	cell probl	weight cutoff	cell importance	exponential transform	dxtran	energy importance	weight window	total
2	2	6.2829e-04	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	6.2829e-04
3	3	-2.3833e-04	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-2.3833e-04
4	4	-3.8324e-04	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-3.8324e-04
5	5	-1.1040e-04	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-1.1040e-04
6	6	-6.4037e-04	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-6.4037e-04
7	7	1.6613e-03	-2.0553e-03	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-3.9399e-04
8	8	2.3320e-04	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	2.3320e-04
9	9	-8.2020e-04	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-8.2020e-04
10	10	-1.9774e-05	-1.0873e-03	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-1.1071e-03
11	11	-4.1392e-04	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-4.1392e-04
12	12	-1.4423e-04	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-1.4423e-04
13	13	-2.1796e-04	-6.4030e-04	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-8.5828e-04
14	14	8.3420e-06	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	8.3420e-06
15	15	1.1178e-04	-2.4672e-04	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-1.3494e-04
16	16	-3.9880e-05	8.2011e-05	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	4.2131e-05
17	17	8.0098e-06	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	8.0098e-06
total		-3.7733e-04	-3.9477e-03	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-4.3250e-03

photon weight balance in each cell -- external events

print table 130

cell progr	cell probl	source	entering	time cutoff	energy cutoff	exiting	total
2	2	0.0000e+00	4.3757e-01	0.0000e+00	0.0000e+00	-7.7028e-01	-3.3271e-01
3	3	0.0000e+00	7.4168e-01	0.0000e+00	0.0000e+00	-8.8507e-01	-1.4340e-01
4	4	0.0000e+00	7.8118e-01	0.0000e+00	0.0000e+00	-8.0165e-01	-2.0487e-02
5	5	0.0000e+00	6.6455e-01	0.0000e+00	0.0000e+00	-6.8126e-01	-1.6708e-02
6	6	0.0000e+00	5.1654e-01	0.0000e+00	0.0000e+00	-5.1240e-01	4.1385e-03
7	7	0.0000e+00	4.2216e-01	0.0000e+00	0.0000e+00	-4.1273e-01	9.4291e-03
8	8	0.0000e+00	3.4380e-01	0.0000e+00	0.0000e+00	-3.4243e-01	1.3692e-03
9	9	0.0000e+00	2.6907e-01	0.0000e+00	0.0000e+00	-2.6822e-01	8.4820e-04
10	10	0.0000e+00	2.1815e-01	0.0000e+00	0.0000e+00	-2.0796e-01	1.0189e-02
11	11	0.0000e+00	1.6544e-01	0.0000e+00	0.0000e+00	-1.5877e-01	6.6713e-03
12	12	0.0000e+00	1.1972e-01	0.0000e+00	0.0000e+00	-1.1689e-01	2.8249e-03
13	13	0.0000e+00	9.3404e-02	0.0000e+00	0.0000e+00	-8.4990e-02	8.4141e-03
14	14	0.0000e+00	6.8196e-02	0.0000e+00	0.0000e+00	-6.2033e-02	6.1631e-03
15	15	0.0000e+00	4.6030e-02	0.0000e+00	0.0000e+00	-4.2757e-02	3.2724e-03
16	16	0.0000e+00	3.1321e-02	0.0000e+00	0.0000e+00	-2.9302e-02	2.0184e-03
17	17	0.0000e+00	1.8727e-02	0.0000e+00	0.0000e+00	-1.7188e-02	1.5391e-03
total		0.0000e+00	4.9375e+00	0.0000e+00	0.0000e+00	-5.3939e+00	-4.5643e-01

photon weight balance in each cell -- physical events

print table 130

cell progr	cell probl	from neutrons	pair production	fluorescence	capture	total
2	2	4.7370e-01	2.0595e-02	0.0000e+00	-1.6122e-01	3.3307e-01
3	3	3.6605e-01	2.4093e-02	0.0000e+00	-2.4661e-01	1.4353e-01
4	4	2.5744e-01	1.7193e-02	0.0000e+00	-2.5389e-01	2.0742e-02
5	5	2.3154e-01	1.4294e-02	0.0000e+00	-2.2870e-01	1.7131e-02
6	6	1.4090e-01	1.5028e-02	0.0000e+00	-1.5956e-01	-3.6377e-03
7	7	1.1192e-01	6.5265e-03	0.0000e+00	-1.1658e-01	-1.2874e-04
8	8	9.6728e-02	7.7932e-03	0.0000e+00	-1.0621e-01	-1.6886e-03
9	9	7.0601e-02	6.0709e-03	0.0000e+00	-7.7286e-02	-6.1368e-04
10	10	4.6872e-02	3.6243e-03	0.0000e+00	-6.0935e-02	-1.0438e-02
11	11	3.7248e-02	3.5918e-03	0.0000e+00	-4.7365e-02	-6.5252e-03
12	12	2.4834e-02	2.4412e-03	0.0000e+00	-3.0103e-02	-2.8276e-03
13	13	1.6266e-02	1.7224e-03	0.0000e+00	-2.5624e-02	-7.6354e-03
14	14	1.1466e-02	9.3843e-04	0.0000e+00	-1.8226e-02	-5.8221e-03
15	15	7.8238e-03	7.6230e-04	0.0000e+00	-1.1836e-02	-3.2494e-03
16	16	5.0573e-03	7.0181e-04	0.0000e+00	-7.8251e-03	-2.0660e-03
17	17	2.2716e-03	3.7546e-04	0.0000e+00	-4.1270e-03	-1.4799e-03
total		1.9007e+00	1.2575e-01	0.0000e+00	-1.5581e+00	4.6836e-01

photon weight balance in each cell -- variance reduction events   print table 130

cell progr	cell probl	weight cutoff	cell importance	exponential transform	dxtran	energy importance	weight window	total
2	2	-3.5828e-04	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-3.5828e-04
3	3	-1.3597e-04	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-1.3597e-04
4	4	-2.5495e-04	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-2.5495e-04
5	5	-4.2311e-04	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-4.2311e-04
6	6	-5.0084e-04	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-5.0084e-04
7	7	-5.2441e-04	-8.7760e-03	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-9.3004e-03
8	8	3.1938e-04	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	3.1938e-04
9	9	-2.3452e-04	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-2.3452e-04
10	10	2.1416e-04	3.5490e-05	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	2.4965e-04
11	11	-1.4613e-04	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-1.4613e-04
12	12	2.7016e-06	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	2.7016e-06
13	13	2.7376e-05	-8.0611e-04	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-7.7873e-04
14	14	-3.4092e-04	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-3.4092e-04
15	15	9.8843e-05	-1.2177e-04	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-2.2927e-05
16	16	2.4065e-05	2.3578e-05	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	4.7643e-05
17	17	-5.9223e-05	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-5.9223e-05
total		-2.2918e-03	-9.6448e-03	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	-1.1937e-02

1 .1t6 neutron activity of each nuclide in each cell, per source particle   print table 140

cell progr name	cell probl name	nuclides	atom fraction	total collisions	collisions * weight	weight lost to capture	weight gain by fission	weight gain by (n,xn)
2	2	1001.50c	8.4765e-02	10731	6.2194e-01	2.3697e-03	0.0000e+00	0.0000e+00
		8016.50c	6.0408e-01	21940	1.5275e+00	5.1661e-02	0.0000e+00	0.0000e+00
		11023.51c	9.4728e-03	457	3.1111e-02	1.2619e-03	0.0000e+00	0.0000e+00
		12000.51c	2.9983e-03	124	8.9465e-03	5.0709e-04	0.0000e+00	8.0808e-05
		13027.50c	2.4835e-02	825	6.0647e-02	3.0993e-03	0.0000e+00	1.7405e-04
		14000.51c	2.4187e-01	7713	5.8114e-01	4.9679e-02	0.0000e+00	4.1939e-03
		19000.51c	6.8546e-03	234	1.7263e-02	2.7491e-03	0.0000e+00	0.0000e+00
		20000.51c	2.0480e-02	766	5.5159e-02	6.2841e-03	0.0000e+00	0.0000e+00
		26000.55c	4.6497e-03	351	2.3192e-02	1.3220e-03	0.0000e+00	1.0202e-03
3	3	1001.50c	8.4785e-02	19161	1.0757e+00	4.6633e-03	0.0000e+00	0.0000e+00
		8016.50c	6.0408e-01	31646	2.0120e+00	3.7679e-02	0.0000e+00	0.0000e+00
		11023.51c	9.4728e-03	684	4.4918e-02	1.4659e-03	0.0000e+00	0.0000e+00
		12000.51c	2.9983e-03	177	1.1862e-02	3.7971e-04	0.0000e+00	0.0000e+00
		13027.50c	2.4835e-02	1060	7.1419e-02	2.6328e-03	0.0000e+00	2.6108e-04
		14000.51c	2.4187e-01	10381	6.9065e-01	4.0329e-02	0.0000e+00	3.0362e-03
		19000.51c	6.8546e-03	301	1.8815e-02	3.4506e-03	0.0000e+00	0.0000e+00
		20000.51c	2.0480e-02	1035	6.7660e-02	6.7648e-03	0.0000e+00	0.0000e+00
		26000.55c	4.6497e-03	532	3.1767e-02	2.0627e-03	0.0000e+00	5.3137e-04
4	4	1001.50c	8.4765e-02	23847	1.2928e+00	6.1985e-03	0.0000e+00	0.0000e+00
		8016.50c	6.0408e-01	36395	2.1801e+00	2.6326e-02	0.0000e+00	0.0000e+00
		11023.51c	9.4728e-03	855	5.2872e-02	1.8595e-03	0.0000e+00	0.0000e+00
		12000.51c	2.9983e-03	206	1.2710e-02	2.8745e-04	0.0000e+00	0.0000e+00
		13027.50c	2.4835e-02	1120	7.1481e-02	2.5271e-03	0.0000e+00	1.6402e-04
		14000.51c	2.4187e-01	11080	6.8892e-01	3.2485e-02	0.0000e+00	2.1819e-03
		19000.51c	6.8546e-03	400	2.3894e-02	4.4596e-03	0.0000e+00	0.0000e+00
		20000.51c	2.0480e-02	1168	7.1442e-02	6.1683e-03	0.0000e+00	0.0000e+00
		26000.55c	4.6497e-03	716	4.0388e-02	2.7533e-03	0.0000e+00	2.0921e-04
5	5	1001.50c	8.4765e-02	26106	1.3639e+00	7.1186e-03	0.0000e+00	0.0000e+00
		8016.50c	6.0408e-01	36557	2.0826e+00	1.7577e-02	0.0000e+00	0.0000e+00
		11023.51c	9.4728e-03	811	4.8365e-02	1.4378e-03	0.0000e+00	0.0000e+00
		12000.51c	2.9983e-03	237	1.3414e-02	1.5027e-04	0.0000e+00	0.0000e+00
		13027.50c	2.4835e-02	1064	6.5743e-02	2.4370e-03	0.0000e+00	0.0000e+00
		14000.51c	2.4187e-01	10836	6.3651e-01	2.5488e-02	0.0000e+00	1.0359e-03
		19000.51c	6.8546e-03	364	2.0436e-02	4.2575e-03	0.0000e+00	0.0000e+00
		20000.51c	2.0480e-02	1160	6.5766e-02	5.1183e-03	0.0000e+00	0.0000e+00
		26000.55c	4.6497e-03	730	4.0139e-02	2.9601e-03	0.0000e+00	2.3247e-04
6	6	1001.50c	8.4765e-02	26139	1.3159e+00	7.4626e-03	0.0000e+00	0.0000e+00
		8016.50c	6.0408e-01	33863	1.8452e+00	1.1412e-02	0.0000e+00	0.0000e+00
		11023.51c	9.4728e-03	741	4.2408e-02	1.3938e-03	0.0000e+00	0.0000e+00
		12000.51c	2.9983e-03	183	1.0692e-02	1.0830e-04	0.0000e+00	0.0000e+00
		13027.50c	2.4835e-02	901	5.1820e-02	1.9842e-03	0.0000e+00	0.0000e+00
		14000.51c	2.4187e-01	9551	5.3523e-01	1.9678e-02	0.0000e+00	9.8792e-04
		19000.51c	6.8546e-03	358	1.8893e-02	4.8644e-03	0.0000e+00	0.0000e+00
		20000.51c	2.0480e-02	1010	5.4817e-02	4.2618e-03	0.0000e+00	0.0000e+00
		26000.55c	4.6497e-03	702	3.6324e-02	3.0200e-03	0.0000e+00	0.0000e+00
7	7	1001.50c	8.4765e-02	23668	1.1559e+00	7.0037e-03	0.0000e+00	0.0000e+00
		8016.50c	6.0408e-01	29876	1.5566e+00	7.4980e-03	0.0000e+00	0.0000e+00
		11023.51c	9.4728e-03	646	3.5918e-02	1.3961e-03	0.0000e+00	7.8411e-05
		12000.51c	2.9983e-03	153	8.4617e-03	9.9841e-05	0.0000e+00	0.0000e+00
		13027.50c	2.4835e-02	798	4.3138e-02	1.7672e-03	0.0000e+00	0.0000e+00
		14000.51c	2.4187e-01	8236	4.4239e-01	1.6564e-02	0.0000e+00	5.4687e-04
		19000.51c	6.8546e-03	306	1.5693e-02	3.9464e-03	0.0000e+00	0.0000e+00
		20000.51c	2.0480e-02	887	4.6229e-02	3.6898e-03	0.0000e+00	0.0000e+00
		26000.55c	4.6497e-03	623	3.1090e-02	2.6978e-03	0.0000e+00	6.3987e-05
8	8	1001.50c	8.4765e-02	40978	9.7550e-01	6.1622e-03	0.0000e+00	0.0000e+00
		8016.50c	6.0408e-01	50111	1.2678e+00	4.5978e-03	0.0000e+00	0.0000e+00
		11023.51c	9.4728e-03	1083	2.8538e-02	1.1816e-03	0.0000e+00	0.0000e+00
		12000.51c	2.9983e-03	251	6.3700e-03	7.0271e-05	0.0000e+00	0.0000e+00

		13027.50c	2.4835e-02	1260	3.4221e-02	1.5802e-03	0.0000e+00	2.6051e-05
		14000.51c	2.4187e-01	13772	3.5666e-01	1.3374e-02	0.0000e+00	3.6981e-04
		19000.51c	6.8546e-03	508	1.2748e-02	3.3676e-03	0.0000e+00	0.0000e+00
		20000.51c	2.0480e-02	1488	3.7685e-02	2.8672e-03	0.0000e+00	0.0000e+00
		26000.55c	4.6497e-03	1148	2.7591e-02	2.7064e-03	0.0000e+00	6.8743e-05
9	9	1001.50c	6.4765e-02	33895	7.9003e-01	5.1786e-03	0.0000e+00	0.0000e+00
		8016.50c	6.0408e-01	40161	9.8953e-01	2.9706e-03	0.0000e+00	0.0000e+00
		11023.51c	9.4728e-03	865	2.2680e-02	9.4945e-04	0.0000e+00	0.0000e+00
		12000.51c	2.9983e-03	209	5.3299e-03	7.2815e-05	0.0000e+00	0.0000e+00
		13027.50c	2.4835e-02	976	2.5455e-02	1.2472e-03	0.0000e+00	4.3513e-05
		14000.51c	2.4187e-01	10681	2.6742e-01	9.5739e-03	0.0000e+00	1.2260e-04
		19000.51c	6.8546e-03	421	1.0053e-02	2.8968e-03	0.0000e+00	0.0000e+00
		20000.51c	2.0480e-02	1096	2.7101e-02	2.0227e-03	0.0000e+00	0.0000e+00
		26000.55c	4.6497e-03	922	2.1139e-02	2.2150e-03	0.0000e+00	3.6849e-05
10	10	1001.50c	6.4765e-02	26746	6.1260e-01	4.1311e-03	0.0000e+00	0.0000e+00
		8016.50c	6.0408e-01	31246	7.4975e-01	1.8203e-03	0.0000e+00	0.0000e+00
		11023.51c	9.4728e-03	681	1.6967e-02	7.8590e-04	0.0000e+00	0.0000e+00
		12000.51c	2.9983e-03	147	3.6757e-03	3.8933e-05	0.0000e+00	0.0000e+00
		13027.50c	2.4835e-02	678	1.6801e-02	8.6901e-04	0.0000e+00	0.0000e+00
		14000.51c	2.4187e-01	8171	1.9908e-01	7.2822e-03	0.0000e+00	1.2740e-04
		19000.51c	6.8546e-03	342	7.8490e-03	2.2657e-03	0.0000e+00	0.0000e+00
		20000.51c	2.0480e-02	904	2.1545e-02	1.5944e-03	0.0000e+00	0.0000e+00
		26000.55c	4.6497e-03	691	1.5933e-02	1.7327e-03	0.0000e+00	0.0000e+00
11	11	1001.50c	6.4765e-02	41211	4.5994e-01	3.2084e-03	0.0000e+00	0.0000e+00
		8016.50c	6.0408e-01	47376	5.5210e-01	1.0638e-03	0.0000e+00	0.0000e+00
		11023.51c	9.4728e-03	981	1.1858e-02	5.6414e-04	0.0000e+00	0.0000e+00
		12000.51c	2.9983e-03	263	3.1433e-03	3.0945e-05	0.0000e+00	0.0000e+00
		13027.50c	2.4835e-02	1121	1.3623e-02	7.7383e-04	0.0000e+00	0.0000e+00
		14000.51c	2.4187e-01	12208	1.4391e-01	5.3941e-03	0.0000e+00	1.0035e-04
		19000.51c	6.8546e-03	499	5.7168e-03	1.7297e-03	0.0000e+00	0.0000e+00
		20000.51c	2.0480e-02	1418	1.6387e-02	1.3140e-03	0.0000e+00	1.2624e-05
		26000.55c	4.6497e-03	1156	1.2957e-02	1.4652e-03	0.0000e+00	1.6011e-05
12	12	1001.50c	8.4765e-02	29568	3.2280e-01	2.3176e-03	0.0000e+00	0.0000e+00
		8016.50c	6.0408e-01	34219	3.8946e-01	7.4590e-04	0.0000e+00	0.0000e+00
		11023.51c	9.4728e-03	890	8.3614e-03	4.0574e-04	0.0000e+00	0.0000e+00
		12000.51c	2.9983e-03	169	1.8538e-03	2.0626e-05	0.0000e+00	0.0000e+00
		13027.50c	2.4835e-02	758	8.9741e-03	5.2655e-04	0.0000e+00	1.4848e-05
		14000.51c	2.4187e-01	8885	1.0230e-01	3.9404e-03	0.0000e+00	3.1504e-05
		19000.51c	6.8546e-03	362	3.9281e-03	1.2442e-03	0.0000e+00	0.0000e+00
		20000.51c	2.0480e-02	940	1.0727e-02	8.5396e-04	0.0000e+00	0.0000e+00
		26000.55c	4.6497e-03	801	8.9183e-03	1.0104e-03	0.0000e+00	2.3426e-05
13	13	1001.50c	6.4765e-02	21832	2.3549e-01	1.7148e-03	0.0000e+00	0.0000e+00
		8016.50c	6.0406e-01	24563	2.7234e-01	4.2212e-04	0.0000e+00	0.0000e+00
		11023.51c	9.4728e-03	488	5.8341e-03	2.9744e-04	0.0000e+00	0.0000e+00
		12000.51c	2.9983e-03	114	1.3201e-03	1.3248e-05	0.0000e+00	0.0000e+00
		13027.50c	2.4635e-02	520	6.0795e-03	3.5292e-04	0.0000e+00	0.0000e+00
		14000.51c	2.4187e-01	6446	7.2396e-02	2.8278e-03	0.0000e+00	2.6023e-05
		19000.51c	6.6546e-03	277	3.0312e-03	9.9041e-04	0.0000e+00	0.0000e+00
		20000.51c	2.0480e-02	726	8.0839e-03	6.4624e-04	0.0000e+00	0.0000e+00
		26000.55c	4.6497e-03	594	6.4841e-03	7.4565e-04	0.0000e+00	0.0000e+00
14	14	1001.50c	8.4765e-02	31690	1.6734e-01	1.2409e-03	0.0000e+00	0.0000e+00
		8016.50c	6.0408e-01	35402	1.9291e-01	2.6200e-04	0.0000e+00	0.0000e+00
		11023.51c	9.4728e-03	654	3.6799e-03	2.2705e-04	0.0000e+00	0.0000e+00
		12000.51c	2.9983e-03	151	8.0263e-04	8.1921e-06	0.0000e+00	0.0000e+00
		13027.50c	2.4835e-02	742	4.1570e-03	2.5803e-04	0.0000e+00	0.0000e+00
		14000.51c	2.4187e-01	8983	4.9355e-02	1.8888e-03	0.0000e+00	5.7811e-06
		19000.51c	6.6546e-03	348	1.8213e-03	5.9912e-04	0.0000e+00	0.0000e+00
		20000.51c	2.0480e-02	975	5.4098e-03	4.3148e-04	0.0000e+00	0.0000e+00
		26000.55c	4.6497e-03	850	4.4622e-03	5.0690e-04	0.0000e+00	0.0000e+00
15	15	1001.50c	8.4765e-02	21585	1.1215e-01	8.4259e-04	0.0000e+00	0.0000e+00
		8016.50c	6.0408e-01	24388	1.3197e-01	1.6702e-04	0.0000e+00	0.0000e+00
		11023.51c	9.4728e-03	446	2.4868e-03	1.4733e-04	0.0000e+00	0.0000e+00
		12000.51c	2.9983e-03	115	6.2164e-04	7.1794e-06	0.0000e+00	0.0000e+00

		13027.50c	2.4835e-02	556	3.1226e-03	1.9118e-04	0.0000e+00	0.0000e+00
		14000.51c	2.4187e-01	6039	3.3174e-02	1.2782e-03	0.0000e+00	1.0209e-05
		19000.51c	6.8546e-03	235	1.2570e-03	4.1639e-04	0.0000e+00	0.0000e+00
		20000.51c	2.0480e-02	686	3.7170e-03	2.9121e-04	0.0000e+00	0.0000e+00
		26000.55c	4.6497e-03	587	3.0818e-03	3.5039e-04	0.0000e+00	0.0000e+00
16	16	1001.50c	8.4765e-02	26467	6.8810e-02	5.1876e-04	0.0000e+00	0.0000e+00
		8016.50c	6.0408e-01	29602	7.9518e-02	1.0297e-04	0.0000e+00	0.0000e+00
		11023.51c	9.4728e-03	632	1.7772e-03	1.0363e-04	0.0000e+00	0.0000e+00
		12000.51c	2.9983e-03	153	4.0298e-04	3.9456e-06	0.0000e+00	0.0000e+00
		13027.50c	2.4835e-02	644	1.7815e-03	1.1107e-04	0.0000e+00	0.0000e+00
		14000.51c	2.4187e-01	7656	2.0752e-02	8.0907e-04	0.0000e+00	1.5178e-05
		19000.51c	6.8546e-03	313	8.1873e-04	2.6140e-04	0.0000e+00	0.0000e+00
		20000.51c	2.0480e-02	883	2.3431e-03	1.9373e-04	0.0000e+00	0.0000e+00
		26000.55c	4.6497e-03	707	1.8395e-03	2.1713e-04	0.0000e+00	0.0000e+00
17	17	1001.50c	8.4765e-02	23194	3.0026e-02	2.2225e-04	0.0000e+00	0.0000e+00
		8016.50c	6.0408e-01	26336	3.5048e-02	5.5039e-05	0.0000e+00	0.0000e+00
		11023.51c	9.4728e-03	499	6.9600e-04	3.8934e-05	0.0000e+00	0.0000e+00
		12000.51c	2.9983e-03	138	1.9834e-04	2.4715e-06	0.0000e+00	0.0000e+00
		13027.50c	2.4835e-02	581	8.1933e-04	5.1020e-05	0.0000e+00	0.0000e+00
		14000.51c	2.4187e-01	6719	9.0753e-03	3.5636e-04	0.0000e+00	7.4875e-06
		19000.51c	6.8546e-03	287	3.6414e-04	1.2808e-04	0.0000e+00	0.0000e+00
		20000.51c	2.0480e-02	814	1.0685e-03	9.1480e-05	0.0000e+00	0.0000e+00
		26000.55c	4.6497e-03	641	8.2148e-04	9.5782e-05	0.0000e+00	0.0000e+00
total				1168723	3.3185e+01	5.9746e-01	0.0000e+00	1.5857e-02

total over all cells for each nuclide

1001.50c	426818	1.0601e+01	6.0351e-02	0.0000e+00	0.0000e+00
8016.50c	533881	1.5864e+01	1.6436e-01	0.0000e+00	0.0000e+00
11023.51c	11213	3.5847e-01	1.3514e-02	0.0000e+00	7.8411e-05
12000.51c	2790	8.9803e-02	1.7993e-03	0.0000e+00	8.0808e-05
13027.50c	13604	4.7928e-01	2.0389e-02	0.0000e+00	6.8356e-04
14000.51c	147355	4.8090e+00	2.3095e-01	0.0000e+00	1.2799e-02
19000.51c	5555	1.6258e-01	3.7627e-02	0.0000e+00	0.0000e+00
20000.51c	15956	4.9514e-01	4.2614e-02	0.0000e+00	1.2624e-05
26000.55c	11751	3.0613e-01	2.5862e-02	0.0000e+00	2.2023e-03

photon activity of each nuclide in each cell, per source particle print table 140

cell progr name	probl name	nuclides	atom fraction	total collisions	collisions * weight	weight lost to capture	total from neutrons	weight from neutrons	avg photon energy
2	2	1000.01p	8.4765e-02	22	2.0954e-02	1.8332e-08	2	2.0000e-03	2.2246e+00
		8000.01p	6.0408e-01	1037	7.8777e-01	1.2471e-02	164	1.6400e-01	3.6956e+00
		11000.01p	9.4728e-03	26	2.0349e-02	7.3857e-04	7	7.0000e-03	9.1239e-01
		12000.01p	2.9983e-03	10	8.8207e-03	5.1782e-04	1	1.0000e-03	1.7617e+00
		13000.01p	2.4835e-02	90	5.9933e-02	7.8410e-03	22	2.2000e-02	3.1602e+00
		14000.01p	2.4187e-01	928	6.3652e-01	7.7254e-02	242	2.4175e-01	2.6897e+00
		19000.01p	6.8546e-03	47	3.0854e-02	8.1211e-03	7	7.0000e-03	1.9871e+00
		20000.01p	2.0480e-02	179	9.4374e-02	3.4992e-02	20	1.9961e-02	2.4650e+00
		26000.01p	4.6497e-03	82	2.6845e-02	1.9289e-02	9	8.9931e-03	1.8637e+00
3	3	1000.01p	8.4765e-02	27	1.8110e-02	4.3990e-08	3	2.7000e-03	2.2246e+00
		8000.01p	6.0408e-01	1727	1.2276e+00	2.0876e-02	104	9.3600e-02	4.1564e+00
		11000.01p	9.4728e-03	44	3.1720e-02	1.7062e-03	13	1.1700e-02	1.9630e+00
		12000.01p	2.9983e-03	16	1.2107e-02	1.4312e-03	0	0.0000e+00	0.0000e+00
		13000.01p	2.4835e-02	163	1.0887e-01	9.5835e-03	21	1.8900e-02	3.1552e+00
		14000.01p	2.4187e-01	1420	8.9526e-01	1.0966e-01	225	2.0231e-01	2.3996e+00
		19000.01p	6.8546e-03	83	4.8179e-02	1.3864e-02	8	5.4000e-03	2.6030e+00
		20000.01p	2.0480e-02	307	1.6158e-01	5.4631e-02	21	1.8859e-02	3.1684e+00
		26000.01p	4.6497e-03	147	6.3593e-02	3.4861e-02	14	1.2580e-02	3.0311e+00
4	4	1000.01p	8.4765e-02	30	1.9313e-02	7.6334e-08	6	4.8000e-03	2.2246e+00
		8000.01p	6.0408e-01	1740	1.1194e+00	2.0351e-02	74	5.9200e-02	3.6626e+00
		11000.01p	9.4728e-03	53	2.4331e-02	2.0238e-03	14	1.1200e-02	3.0461e+00
		12000.01p	2.9983e-03	15	1.0347e-02	8.8859e-04	1	8.0000e-04	4.6574e+00
		13000.01p	2.4835e-02	142	8.6924e-02	1.0046e-02	12	9.6000e-03	3.9757e+00
		14000.01p	2.4187e-01	1643	9.2179e-01	1.2462e-01	165	1.3189e-01	2.7054e+00
		19000.01p	6.8546e-03	102	4.3366e-02	1.3927e-02	13	1.0400e-02	2.8031e+00
		20000.01p	2.0480e-02	303	1.4708e-01	5.1654e-02	23	1.8376e-02	2.7187e+00
		26000.01p	4.6497e-03	147	6.2295e-02	3.0377e-02	14	1.1167e-02	1.7643e+00
5	5	1000.01p	8.4765e-02	34	2.2095e-02	7.1989e-08	6	4.2000e-03	2.2246e+00
		8000.01p	6.0408e-01	1697	9.8001e-01	1.8981e-02	79	5.5300e-02	3.5740e+00
		11000.01p	9.4728e-03	35	1.9861e-02	1.1256e-03	13	9.1000e-03	3.0536e+00
		12000.01p	2.9983e-03	21	1.0899e-02	1.0420e-03	0	0.0000e+00	0.0000e+00
		13000.01p	2.4835e-02	121	6.4513e-02	7.4985e-03	20	1.4000e-02	2.1145e+00
		14000.01p	2.4187e-01	1544	7.8159e-01	1.1036e-01	169	1.1819e-01	2.7300e+00
		19000.01p	6.8546e-03	88	3.6150e-02	1.3009e-02	14	9.8000e-03	2.3426e+00
		20000.01p	2.0480e-02	317	1.3039e-01	4.5068e-02	17	1.1886e-02	2.4823e+00
		26000.01p	4.6497e-03	144	5.3561e-02	3.1617e-02	13	9.0600e-03	2.4236e+00
6	6	1000.01p	8.4765e-02	25	1.2433e-02	5.7057e-08	9	5.4000e-03	2.2246e+00
		8000.01p	6.0408e-01	1461	7.7338e-01	1.4771e-02	42	2.5200e-02	4.1378e+00
		11000.01p	9.4728e-03	32	1.3531e-02	1.0041e-03	9	5.4000e-03	2.4557e+00
		12000.01p	2.9983e-03	21	1.0495e-02	5.4806e-04	0	0.0000e+00	0.0000e+00
		13000.01p	2.4835e-02	108	4.6120e-02	5.0193e-03	14	8.4000e-03	3.2852e+00
		14000.01p	2.4187e-01	1243	5.8203e-01	7.7298e-02	111	6.6556e-02	3.0471e+00
		19000.01p	6.8546e-03	71	2.6274e-02	8.8622e-03	21	1.2600e-02	3.0118e+00
		20000.01p	2.0480e-02	263	1.0435e-01	3.1294e-02	13	7.7879e-03	3.8208e+00
		26000.01p	4.6497e-03	117	3.7576e-02	2.0768e-02	16	9.5557e-03	2.8977e+00
7	7	1000.01p	8.4765e-02	27	1.3313e-02	3.2871e-08	9	4.5000e-03	2.2246e+00
		8000.01p	6.0408e-01	1221	5.1444e-01	1.0051e-02	34	1.7000e-02	4.8446e+00
		11000.01p	9.4728e-03	32	1.2154e-02	6.1297e-04	16	8.0000e-03	1.0447e+00
		12000.01p	2.9983e-03	11	3.1544e-03	3.8774e-04	2	1.0000e-03	1.3893e+00
		13000.01p	2.4835e-02	75	2.9266e-02	2.8311e-03	9	4.5000e-03	2.6286e+00
		14000.01p	2.4187e-01	1088	4.0445e-01	5.5460e-02	105	5.2475e-02	2.5294e+00
		19000.01p	6.8546e-03	58	2.0024e-02	5.0420e-03	19	9.5000e-03	2.8595e+00
		20000.01p	2.0480e-02	211	7.1691e-02	2.6563e-02	9	4.4927e-03	2.2602e+00
		26000.01p	4.6497e-03	101	2.8565e-02	1.7628e-02	21	1.0453e-02	3.9680e+00
8	8	1000.01p	8.4765e-02	42	8.2288e-03	3.0059e-08	26	5.2000e-03	2.2246e+00
		8000.01p	6.0408e-01	2658	4.7453e-01	9.3138e-03	46	9.2000e-03	4.1428e+00
		11000.01p	9.4728e-03	51	7.3036e-03	5.1716e-04	37	7.4000e-03	1.5232e+00
		12000.01p	2.9983e-03	21	2.8345e-03	1.8330e-04	1	2.0000e-04	1.3553e+00

		13000.01p	2.4835e-02	202	3.1521e-02	3.3561e-03	36	7.6000e-03	3.0805e+00
		14000.01p	2.4187e-01	2285	3.6688e-01	4.8138e-02	218	4.3573e-02	3.0516e+00
		19000.01p	6.8546e-03	128	1.9249e-02	5.6553e-03	35	7.0000e-03	2.8783e+00
		20000.01p	2.0480e-02	459	6.3724e-02	2.4839e-02	42	8.3923e-03	2.9990e+00
		26000.01p	4.6497e-03	225	2.5355e-02	1.4207e-02	41	8.1626e-03	2.8114e+00
9	9	1000.01p	8.4765e-02	35	5.1628e-03	1.5969e-08	30	4.5000e-03	2.2246e+00
		8000.01p	6.0408e-01	2419	3.5107e-01	6.4392e-03	50	7.5000e-03	3.8457e+00
		11000.01p	9.4728e-03	54	8.2074e-03	3.7480e-04	28	4.2000e-03	2.0690e+00
		12000.01p	2.9983e-03	34	5.7019e-03	5.8067e-04	2	3.0000e-04	2.0564e+00
		13000.01p	2.4835e-02	162	2.2522e-02	2.1319e-03	37	5.5500e-03	2.9419e+00
		14000.01p	2.4187e-01	1997	2.6532e-01	3.3543e-02	206	3.0886e-02	3.3274e+00
		19000.01p	6.8546e-03	115	1.5482e-02	4.1540e-03	50	7.5000e-03	2.4612e+00
		20000.01p	2.0480e-02	417	4.6832e-02	1.8221e-02	27	4.0455e-03	2.8956e+00
		26000.01p	4.6497e-03	203	2.0601e-02	1.1841e-02	41	6.1196e-03	3.7976e+00
10	10	1000.01p	8.4765e-02	29	3.8837e-03	2.5414e-08	29	2.9000e-03	2.2246e+00
		8000.01p	6.0408e-01	2381	2.7016e-01	5.0601e-03	46	4.6000e-03	3.6199e+00
		11000.01p	9.4728e-03	61	7.5677e-03	4.0759e-04	33	3.3000e-03	1.9278e+00
		12000.01p	2.9983e-03	21	2.1029e-03	1.3635e-04	2	2.0000e-04	4.8170e+00
		13000.01p	2.4835e-02	199	2.0493e-02	2.3989e-03	24	2.4000e-03	4.7474e+00
		14000.01p	2.4187e-01	2033	2.1582e-01	2.9344e-02	211	2.1091e-02	3.2476e+00
		19000.01p	6.8546e-03	99	9.2578e-03	2.8752e-03	46	4.6000e-03	3.0529e+00
		20000.01p	2.0480e-02	367	3.4609e-02	1.2290e-02	39	3.8978e-03	3.9135e+00
		26000.01p	4.6497e-03	177	1.5069e-02	8.4221e-03	39	3.8834e-03	2.8389e+00
11	11	1000.01p	8.4765e-02	67	3.6535e-03	8.6732e-09	54	2.2950e-03	2.2246e+00
		8000.01p	6.0408e-01	4376	2.1389e-01	4.2291e-03	60	2.5500e-03	3.1917e+00
		11000.01p	9.4728e-03	99	4.4195e-03	3.2361e-04	64	2.7200e-03	2.0097e+00
		12000.01p	2.9983e-03	30	1.8620e-03	1.1845e-04	4	1.7000e-04	2.8200e+00
		13000.01p	2.4835e-02	320	1.4628e-02	1.7137e-03	49	2.0825e-03	3.4870e+00
		14000.01p	2.4187e-01	3563	1.6640e-01	2.1608e-02	362	1.6230e-02	3.4684e+00
		19000.01p	6.8546e-03	194	7.7303e-03	2.6472e-03	89	3.7825e-03	3.2049e+00
		20000.01p	2.0480e-02	669	2.6494e-02	9.8646e-03	82	3.4840e-03	3.3982e+00
		26000.01p	4.6497e-03	357	1.1824e-02	6.8609e-03	93	3.9346e-03	3.2747e+00
12	12	1000.01p	8.4765e-02	77	3.3310e-03	8.5087e-09	52	1.8200e-03	2.2246e+00
		8000.01p	6.0408e-01	3741	1.4206e-01	2.4850e-03	51	1.7850e-03	4.1177e+00
		11000.01p	9.4728e-03	70	2.3558e-03	1.3218e-04	48	1.6600e-03	1.2338e+00
		12000.01p	2.9983e-03	31	1.0814e-03	9.2651e-05	2	7.0000e-05	1.5393e+00
		13000.01p	2.4835e-02	308	1.0192e-02	1.1355e-03	46	1.6100e-03	3.3118e+00
		14000.01p	2.4187e-01	3098	1.1425e-01	1.4551e-02	306	1.0706e-02	3.3283e+00
		19000.01p	6.8546e-03	148	4.9575e-03	1.5682e-03	73	2.5550e-03	2.9892e+00
		20000.01p	2.0480e-02	569	1.7855e-02	6.3923e-03	62	2.1685e-03	3.2047e+00
		26000.01p	4.6497e-03	251	6.9051e-03	3.7457e-03	70	2.4391e-03	2.9561e+00
13	13	1000.01p	8.4765e-02	58	2.3337e-03	5.9462e-09	42	1.0500e-03	2.2246e+00
		8000.01p	6.0408e-01	3502	1.1536e-01	2.2036e-03	25	6.2500e-04	4.0040e+00
		11000.01p	9.4728e-03	85	2.0212e-03	1.4962e-04	49	1.2399e-03	2.1913e+00
		12000.01p	2.9983e-03	28	7.3495e-04	5.1504e-05	2	5.0000e-05	2.0577e+00
		13000.01p	2.4835e-02	281	9.1172e-03	9.2749e-04	36	9.0000e-04	2.9998e+00
		14000.01p	2.4187e-01	2868	9.2797e-02	1.1923e-02	297	7.4302e-03	3.3838e+00
		19000.01p	6.8546e-03	165	4.5496e-03	1.1730e-03	61	2.0296e-03	2.9781e+00
		20000.01p	2.0480e-02	548	1.5211e-02	5.7416e-03	56	1.3994e-03	3.0338e+00
		26000.01p	4.6497e-03	241	6.3728e-03	3.4539e-03	62	1.5422e-03	2.6183e+00
14	14	1000.01p	8.4765e-02	126	1.8646e-03	5.2407e-09	79	6.9125e-04	2.2246e+00
		8000.01p	6.0408e-01	6335	8.2485e-02	1.5188e-03	70	6.1250e-04	3.7454e+00
		11000.01p	9.4728e-03	139	1.8586e-03	1.4893e-04	85	7.4375e-04	1.6849e+00
		12000.01p	2.9983e-03	49	7.5446e-04	5.2917e-05	5	4.3750e-05	2.4375e+00
		13000.01p	2.4835e-02	468	5.6141e-03	6.5851e-04	67	5.8713e-04	2.9281e+00
		14000.01p	2.4187e-01	5084	6.3780e-02	8.5480e-03	584	5.1530e-03	3.5124e+00
		19000.01p	6.8546e-03	276	3.2805e-03	1.0344e-03	142	1.2479e-03	2.8948e+00
		20000.01p	2.0480e-02	932	1.0968e-02	3.7160e-03	115	1.0059e-03	2.9778e+00
		26000.01p	4.6497e-03	411	4.6051e-03	2.5488e-03	158	1.3806e-03	3.5837e+00
15	15	1000.01p	8.4765e-02	94	1.0779e-03	2.9324e-09	119	5.9500e-04	2.2246e+00
		8000.01p	6.0408e-01	5331	5.4200e-02	9.4208e-04	72	3.6000e-04	4.2865e+00
		11000.01p	9.4728e-03	130	1.3133e-03	6.6609e-05	110	5.5686e-04	1.9773e+00
		12000.01p	2.9983e-03	46	4.1064e-04	2.3428e-05	3	1.5000e-05	2.3961e+00

		13000.01p	2.4835e-02	435	4.3732e-03	4.6690e-04	96	5.0015e-04	3.3811e+00
		14000.01p	2.4187e-01	4351	4.2880e-02	5.4774e-03	633	3.3514e-03	3.4753e+00
		19000.01p	6.8546e-03	236	2.0560e-03	6.4460e-04	146	8.2083e-04	2.9356e+00
		20000.01p	2.0480e-02	691	6.3758e-03	2.2825e-03	146	7.3563e-04	3.0634e+00
		26000.01p	4.6497e-03	350	3.1883e-03	1.9321e-03	177	8.8893e-04	3.3961e+00
16	16	1000.01p	8.4765e-02	115	4.8385e-04	1.4353e-09	190	3.5625e-04	2.2246e+00
		8000.01p	6.0408e-01	8138	3.6538e-02	6.4657e-04	110	2.0854e-04	3.6646e+00
		11000.01p	9.4728e-03	192	8.9621e-04	6.0542e-05	215	4.2049e-04	1.8404e+00
		12000.01p	2.9983e-03	66	2.9159e-04	2.5383e-05	8	1.5927e-05	3.0507e+00
		13000.01p	2.4835e-02	569	2.4251e-03	2.5069e-04	144	2.8019e-04	3.3159e+00
		14000.01p	2.4187e-01	6628	2.8704e-02	3.6953e-03	1126	2.2568e-03	3.5047e+00
		19000.01p	6.8546e-03	332	1.2569e-03	3.9763e-04	216	5.0821e-04	2.7861e+00
		20000.01p	2.0480e-02	1082	4.3775e-03	1.5289e-03	252	4.8976e-04	3.0774e+00
		26000.01p	4.6497e-03	469	2.1976e-03	1.2201e-03	273	5.2106e-04	3.1562e+00
17	17	1000.01p	8.4765e-02	174	4.2483e-04	1.2890e-09	252	1.5750e-04	2.2246e+00
		8000.01p	6.0408e-01	9639	2.0649e-02	3.5718e-04	180	1.2145e-04	3.8501e+00
		11000.01p	9.4728e-03	226	5.3911e-04	2.6669e-05	198	1.3820e-04	1.7626e+00
		12000.01p	2.9983e-03	77	1.7284e-04	1.0079e-05	14	1.0028e-05	2.1306e+00
		13000.01p	2.4835e-02	749	1.5416e-03	1.4201e-04	194	1.4253e-04	3.5203e+00
		14000.01p	2.4187e-01	7682	1.6748e-02	1.9365e-03	1434	1.0087e-03	3.4881e+00
		19000.01p	6.8546e-03	398	9.7050e-04	2.6121e-04	229	2.5196e-04	2.9480e+00
		20000.01p	2.0480e-02	1207	2.5901e-03	8.1982e-04	318	2.1338e-04	3.1598e+00
		26000.01p	4.6497e-03	567	1.0826e-03	5.7353e-04	342	2.2788e-04	3.2140e+00
total				127108	1.5225e+01	1.5581e+00	14156	1.9007e+00	2.9892e+00

total over all cells for each nuclide

1000.01p	982	1.3666e-01	4.0604e-07	908	4.3165e-02	2.2246e+00
8000.01p	57403	7.1635e+00	1.3070e-01	1207	4.4186e-01	3.8539e+00
11000.01p	1329	1.5843e-01	9.4189e-03	939	7.4799e-02	2.0432e+00
12000.01p	497	7.1569e-02	6.0901e-03	47	3.8747e-03	2.4856e+00
13000.01p	4392	5.1805e-01	5.6001e-02	829	9.9052e-02	3.1056e+00
14000.01p	47455	5.5952e+00	7.3342e-01	6414	9.5486e-01	2.7367e+00
19000.01p	2540	2.7364e-01	8.3236e-02	1187	8.4996e-02	2.7110e+00
20000.01p	8521	9.3851e-01	3.2990e-01	1242	1.0719e-01	2.9043e+00
26000.01p	3989	3.6964e-01	2.0934e-01	1383	9.0909e-02	2.8335e+00

N48

149 summary of photons produced in neutron collisions

progr name	probl name	number of photons	weight per source neut	energy per source neut	avg photon energy	mev/gm per source neut	weight/neut collision	energy/neut collision
1	1	0	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
2	2	474	4.73700e-01	1.41888e+00	2.99531e+00	5.70840e-06	1.62954e-01	4.88099e-01
3	3	407	3.66051e-01	1.07505e+00	2.93690e+00	4.32513e-06	9.09508e-02	2.67113e-01
4	4	322	2.57436e-01	7.59152e-01	2.94889e+00	3.05420e-06	5.80518e-02	1.71188e-01
5	5	331	2.31540e-01	6.61470e-01	2.85683e+00	2.66121e-06	5.33898e-02	1.52525e-01
6	6	235	1.40899e-01	4.55338e-01	3.23166e+00	1.83190e-06	3.60239e-02	1.16417e-01
7	7	224	1.11921e-01	3.25474e-01	2.90807e+00	1.30944e-06	3.35552e-02	9.75808e-02
8	8	484	9.67280e-02	2.84469e-01	2.94091e+00	1.14447e-06	3.52137e-02	1.03560e-01
9	9	471	7.06013e-02	2.20672e-01	3.12560e+00	8.87800e-07	3.27049e-02	1.02223e-01
10	10	469	4.68720e-02	1.50638e-01	3.21381e+00	6.06041e-07	2.85075e-02	9.16176e-02
11	11	677	3.72483e-02	1.19589e-01	3.21060e+00	4.81129e-07	3.05404e-02	9.80531e-02
12	12	710	2.48339e-02	7.63420e-02	3.07410e+00	3.07137e-07	2.89669e-02	8.90473e-02
13	13	650	1.62663e-02	4.98269e-02	3.06320e+00	2.00482e-07	2.66200e-02	8.15424e-02
14	14	1305	1.14657e-02	3.65656e-02	3.18912e+00	1.47110e-07	2.66680e-02	8.50476e-02
15	15	1502	7.82381e-03	2.50241e-02	3.19845e+00	1.00676e-07	2.68323e-02	8.58219e-02
16	16	2534	5.05725e-03	1.57854e-02	3.12133e+00	6.35072e-08	2.84047e-02	8.86604e-02
17	17	3161	2.27164e-03	7.25258e-03	3.19266e+00	2.91784e-08	2.90807e-02	9.28446e-02
	total	14156	1.90072e+00	5.68153e+00	2.98915e+00			

energy interval	number of photons	number frequency	cum number distribution	weight of photons	weight frequency	cum weight distribution
20.00	0	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
15.00	0	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
10.00	10	7.06414e-04	7.06414e-04	5.32799e-03	2.80315e-03	2.80315e-03
9.00	49	3.46143e-03	4.16784e-03	9.81676e-03	5.16477e-03	7.96791e-03
8.00	88	6.21645e-03	1.03843e-02	1.27078e-02	6.68578e-03	1.46537e-02
7.00	759	5.36168e-02	6.40011e-02	8.13888e-02	4.28201e-02	5.74737e-02
6.00	1126	7.95422e-02	1.43543e-01	2.06509e-01	1.08648e-01	1.66122e-01
5.00	457	3.22831e-02	1.75827e-01	6.36222e-02	3.34727e-02	1.99594e-01
4.00	1717	1.21291e-01	2.97118e-01	1.45599e-01	7.66021e-02	2.76197e-01
3.00	2265	1.60003e-01	4.57121e-01	2.48871e-01	1.30935e-01	4.07132e-01
2.00	2687	1.89814e-01	6.46934e-01	2.94282e-01	1.54827e-01	5.61959e-01
1.00	2716	1.91862e-01	8.38796e-01	4.24727e-01	2.23456e-01	7.85415e-01
0.50	1013	7.15598e-02	9.10356e-01	2.09161e-01	1.10043e-01	8.95458e-01
0.10	587	4.14665e-02	9.51823e-01	1.15685e-01	6.08638e-02	9.56322e-01
0.01	587	4.14665e-02	9.93289e-01	7.16400e-02	3.76911e-02	9.94013e-01
0.00	95	6.71094e-03	1.00000e+00	1.13797e-02	5.98703e-03	1.00000e+00
total	14156	1.00000e+00		1.90072e+00	1.00000e+00	

```
150    tally  11         nps =     10000
                  neutron tally type 1:   number of particles crossing a surface.

       surface  18
            energy
          1.0000e-04    5.03891e-03 0.0414
          1.0000e-03    2.80211e-04 0.0981
          1.0000e-02    2.89254e-04 0.1015
          5.0000e-02    2.73441e-04 0.0973
          1.0000e-01    1.46142e-04 0.1312
          5.0000e-01    3.62093e-04 0.1014
          1.0000e+00    3.10792e-04 0.1092
          2.0000e+00    3.43895e-04 0.0986
          3.0000e+00    5.46420e-04 0.1010
          4.0000e+00    1.39397e-04 0.1409
          5.0000e+00    1.57682e-04 0.1475
          6.0000e+00    1.07942e-04 0.2299
          7.0000e+00    1.03403e-04 0.2213
          8.0000e+00    5.83465e-05 0.2793
          9.0000e+00    3.00209e-05 0.5170
          1.0000e+01    2.07414e-05 0.3719
          1.1000e+01    1.73321e-05 0.4025
          1.2000e+01    2.72981e-05 0.3281
          1.3000e+01    5.83876e-05 0.1990
          1.4000e+01    1.84613e-04 0.1704
          1.5000e+01    8.04091e-05 0.2594
            total       8.57673e-03 0.0387

151    for the tally-fluctuation-chart bin:
       average tally per history = 8.57673e-03        largest tally = 7.04326e-01
       largest/average = 8.21205e+01                  nps of largest =      9653
```

```
tally  16         nps =    10000
           neutron tally type 6:   track length estimate of heating.           units   mev/gram

           masses
                   cell:       17
                         2.48580e+05

cell  17
     energy
   1.0000e-04    6.41421e-10 0.0440
   1.0000e-03    1.82880e-12 0.0896
   1.0000e-02    1.42739e-11 0.0943
   5.0000e-02    5.25274e-11 0.0837
   1.0000e-01    7.06196e-11 0.1005
   5.0000e-01    5.79175e-10 0.0830
   1.0000e+00    6.62457e-10 0.0923
   2.0000e+00    1.11314e-09 0.0901
   3.0000e+00    1.21398e-09 0.0926
   4.0000e+00    6.92051e-10 0.1266
   5.0000e+00    7.11761e-10 0.1218
   6.0000e+00    4.40133e-10 0.1625
   7.0000e+00    4.58018e-10 0.1877
   8.0000e+00    4.44805e-10 0.2309
   9.0000e+00    2.20339e-10 0.3869
   1.0000e+01    1.58852e-10 0.2926
   1.1000e+01    1.89422e-10 0.3405
   1.2000e+01    4.35150e-10 0.2402
   1.3000e+01    6.67359e-10 0.1894
   1.4000e+01    1.90263e-09 0.1582
   1.5000e+01    7.27929e-10 0.2391
     total       1.13979e-08 0.0663

for the tally-fluctuation-chart bin:
average tally per history = 2.83306e-03       largest tally = 3.93385e-01
largest/average = 1.38855e+02                 nps of largest =      1793
```

```
tally  34         nps =    10000
           neutron tally type 4:   track length estimate of particle flux.

           volumes
                   cell:       17
                         1.10447e+05

      cell  17
452   multiplier bin:  -1.00000e+00    1           1     -4
           energy
         1.0000e-04   1.44352e-09 0.0440
         1.0000e-03   4.11572e-12 0.0896
         1.0000e-02   3.21235e-11 0.0943
         5.0000e-02   1.18213e-10 0.0837
         1.0000e-01   1.58929e-10 0.1005
         5.0000e-01   1.30343e-09 0.0830
         1.0000e+00   1.49086e-09 0.0923
         2.0000e+00   2.50512e-09 0.0901
         3.0000e+00   2.73206e-09 0.0926
         4.0000e+00   1.55746e-09 0.1266
         5.0000e+00   1.60182e-09 0.1218
         6.0000e+00   9.90519e-10 0.1625
         7.0000e+00   1.03077e-09 0.1877
         8.0000e+00   1.00103e-09 0.2309
         9.0000e+00   4.95872e-10 0.3889
         1.0000e+01   3.57495e-10 0.2926
         1.1000e+01   4.26295e-10 0.3405
         1.2000e+01   9.79305e-10 0.2402
         1.3000e+01   1.50189e-09 0.1894
         1.4000e+01   4.28186e-09 0.1582
         1.5000e+01   1.63820e-09 0.2391
           total      2.56509e-08 0.0663

      for the tally-fluctuation-chart bin:
      average tally per history = 2.83306e-03       largest tally = 3.93385e-01
      largest/average = 1.38855e+02                 nps of largest =      1793
```

tally 1 nps = 10000
photon tally type 1: number of particles crossing a surface.

surface 18

energy		
1.0000e-04	0.00000e+00	0.0000
1.0000e-03	0.00000e+00	0.0000
1.0000e-02	0.00000e+00	0.0000
5.0000e-02	7.40393e-08	0.4263
1.0000e-01	7.39749e-04	0.1000
5.0000e-01	3.14540e-03	0.0631
1.0000e+00	1.49100e-03	0.0793
2.0000e+00	1.65354e-03	0.0807
3.0000e+00	1.29727e-03	0.1122
4.0000e+00	1.02357e-03	0.1151
5.0000e+00	8.27145e-04	0.1941
6.0000e+00	3.09210e-04	0.3138
7.0000e+00	4.56650e-04	0.1699
8.0000e+00	3.57336e-04	0.1705
9.0000e+00	1.59481e-05	0.3797
1.0000e+01	7.17776e-08	0.7650
1.1000e+01	0.00000e+00	0.0000
1.2000e+01	0.00000e+00	0.0000
1.3000e+01	0.00000e+00	0.0000
1.4000e+01	0.00000e+00	0.0000
1.5000e+01	0.00000e+00	0.0000
2.0000e+01	0.00000e+00	0.0000
total	1.13314e-02	0.0535

for the tally-fluctuation-chart bin:
average tally per history = 1.13314e-02 largest tally = 1.68717e+00
largest/average = 1.48894e+02 nps of largest = 5474

```
tally  12         nps =    10000      optimize weight window generator on tally 12
           photon tally type 2:   particle flux averaged over a surface.       units   1/cm**2

          areas
               surface:       18
                        1.76715e+04

surface  18
     energy
   2.0000e+01   1.00182e-06 0.0528

for the tally-fluctuation-chart bin:
average tally per history = 1.77036e-02       largest tally = 2.92887e+00
largest/average = 1.65315e+02                 nps of largest =     5474
```

tally 26 nps = 10000
photon tally type 6: track length estimate of heating. units mev/gram

masses
cell: 17
2.48560e+05

cell 17

energy		
1.0000e-04	0.00000e+00	0.0000
1.0000e-03	0.00000e+00	0.0000
1.0000e-02	2.60321e-13	0.2740
5.0000e-02	9.68889e-11	0.1030
1.0000e-01	1.14297e-09	0.0648
5.0000e-01	4.91832e-09	0.0581
1.0000e+00	4.35676e-09	0.0654
2.0000e+00	7.59310e-09	0.0664
3.0000e+00	7.35866e-09	0.0857
4.0000e+00	8.35214e-09	0.0918
5.0000e+00	8.00567e-09	0.1333
6.0000e+00	3.14033e-09	0.2824
7.0000e+00	5.99933e-09	0.1326
8.0000e+00	5.81891e-09	0.1496
9.0000e+00	3.09310e-10	0.2843
1.0000e+01	5.05803e-10	0.5651
1.1000e+01	0.00000e+00	0.0000
1.2000e+01	0.00000e+00	0.0000
1.3000e+01	0.00000e+00	0.0000
1.4000e+01	0.00000e+00	0.0000
1.5000e+01	0.00000e+00	0.0000
2.0000e+01	0.00000e+00	0.0000
total	5.75984e-08	0.0538

for the tally-fluctuation-chart bin:
average tally per history = 1.43167e-02 largest tally = 2.83462e+00
largest/average = 1.97994e+02 nps of largest = 286

```
tally   6          nps =     10000
         combined tally type 6:   track length estimate of heating.            units   mev/gram

          masses
                 cell:         17
                          2.48560e+05

cell  17
     energy
   1.0000e-05    6.39281e-10 0.0441
   1.0000e-04    2.14020e-12 0.0773
   1.0000e-03    1.82880e-12 0.0896
   1.0000e-02    1.45343e-11 0.0929
   5.0000e-02    1.49416e-10 0.0742
   1.0000e-01    1.21359e-09 0.0617
   5.0000e-01    5.49749e-09 0.0537
   1.0000e+00    5.01922e-09 0.0592
   2.0000e+00    8.70624e-09 0.0605
   3.0000e+00    8.57264e-09 0.0755
   4.0000e+00    9.04419e-09 0.0857
   5.0000e+00    8.71743e-09 0.1230
   6.0000e+00    3.58046e-09 0.2488
   7.0000e+00    6.45734e-09 0.1243
   8.0000e+00    6.26371e-09 0.1402
   9.0000e+00    5.29648e-10 0.2311
   1.0000e+01    6.84655e-10 0.4356
   1.1000e+01    1.89422e-10 0.3405
   1.2000e+01    4.35150e-10 0.2402
   1.3000e+01    6.67359e-10 0.1894
   1.4000e+01    1.90263e-09 0.1582
   1.5000e+01    7.27929e-10 0.2391
   2.0000e+01    0.00000e+00 0.0000
     total       6.89963e-08 0.0482

for the tally-fluctuation-chart bin:
average tally per history = 1.71497e-02        largest tally = 2.83610e+00
largest/average = 1.65373e+02                  nps of largest =      288
```

153 tally fluctuation charts

tally 11

nps	mean	error	fom
1000	8.71851e-03	0.1241	74
2000	9.03658e-03	0.0851	75
3000	8.82390e-03	0.0703	74
4000	8.54998e-03	0.0620	72
5000	8.59782e-03	0.0546	75
6000	8.63999e-03	0.0499	76
7000	8.57037e-03	0.0460	77
8000	8.51572e-03	0.0432	77
9000	8.37039e-03	0.0406	77
10000	8.57673e-03	0.0387	76

tally 16

nps	mean	error	fom
1000	9.70189e-09	0.2219	23
2000	1.04946e-08	0.1620	21
3000	1.06751e-08	0.1270	23
4000	1.00378e-08	0.1116	22
5000	1.01967e-08	0.0955	25
6000	1.07512e-08	0.0867	25
7000	1.06560e-08	0.0794	26
8000	1.11115e-08	0.0742	26
9000	1.09638e-08	0.0700	26
10000	1.13979e-08	0.0663	28

tally 34

nps	mean	error	fom
1000	2.18341e-08	0.2219	23
2000	2.36182e-08	0.1620	21
3000	2.40243e-08	0.1270	23
4000	2.25902e-08	0.1116	22
5000	2.29477e-08	0.0955	25
6000	2.41955e-08	0.0867	25
7000	2.39812e-08	0.0794	26
8000	2.50065e-08	0.0742	26
9000	2.46741e-08	0.0700	26
10000	2.56509e-08	0.0663	26

tally 1

nps	mean	error	fom
1000	1.28117e-02	0.1799	35
2000	1.48446e-02	0.1151	41
3000	1.35604e-02	0.0927	42
4000	1.23611e-02	0.0817	41
5000	1.19881e-02	0.0731	42
6000	1.22168e-02	0.0700	39
7000	1.20677e-02	0.0640	40
8000	1.19741e-02	0.0597	40
9000	1.16873e-02	0.0564	40
10000	1.13314e-02	0.0535	40

tally 12

nps	mean	error	fom
1000	1.15343e-08	0.1731	38
2000	1.30034e-06	0.1066	48
3000	1.17423e-08	0.0871	48
4000	1.08278e-06	0.0763	48
5000	1.05485e-06	0.0693	47
6000	1.07922e-06	0.0687	40
7000	1.05803e-06	0.0630	41
8000	1.04938e-06	0.0590	41
9000	1.02361e-06	0.0559	41
10000	1.00182e-06	0.0528	41

tally 26

nps	mean	error	fom
1000	6.50323e-08	0.2076	27
2000	7.45915e-08	0.1179	39
3000	6.70702e-08	0.0944	41
4000	6.14209e-08	0.0837	40
5000	6.02273e-08	0.0738	41
6000	6.11251e-08	0.0712	37
7000	6.01487e-08	0.0649	39
8000	5.96095e-08	0.0609	38
9000	5.78353e-08	0.0575	39
10000	5.75984e-08	0.0538	40

tally 6

nps	mean	error	fom
1000	7.47342e-08	0.1870	33
2000	8.50861e-08	0.1088	46
3000	7.77453e-08	0.0866	49
4000	7.14588e-08	0.0766	47
5000	7.04240e-08	0.0875	49
6000	7.18763e-08	0.0642	46
7000	7.08046e-08	0.0585	48
8000	7.07210e-08	0.0547	48
9000	6.87991e-08	0.0516	48
10000	6.89963e-08	0.0482	49

```
1 34   neutron weight-window lower bounds from the weight-window generator                    print table 190
        cell
           1     -1.000e+00
           2      5.000e-01
           3      3.697e-01
           4      3.186e-01
           5      2.592e-01
           6      2.167e-01
           7      1.736e-01
           8      1.112e-01
           9      1.057e-01
          10      7.193e-02
          11      6.298e-02
          12      4.346e-02
          13      3.437e-02
          14      2.568e-02
          15      1.997e-02
          16      1.492e-02
          17      1.555e-02
```

photon weight-window lower bounds from the weight-window generator print table 190

photon cell	
1	-1.000e+00
2	0.000e+00
3	2.090e+00
4	8.626e-01
5	8.416e-01
6	4.162e-01
7	2.509e-01
8	1.840e-01
9	8.668e-02
10	6.119e-02
11	4.462e-02
12	2.676e-02
13	1.714e-02
14	1.105e-02
15	7.414e-03
16	4.912e-03
17	3.218e-03

```
N35   weight-window cards from the weight-window generator                                              print table 200

                each card has ten leading blanks that must be removed by a text editor.

                wwp:n 5 3 5
                wwn 1:n   -1.0000e+00  5.0000e-01  3.8970e-01  3.1861e-01  2.5924e-01
                           2.1673e-01  1.7364e-01  1.1118e-01  1.0568e-01  7.1931e-02
                           6.2977e-02  4.3464e-02  3.4366e-02  2.5677e-02  1.9972e-02
                           1.4923e-02  1.5549e-02
                wwp:p 5 3 5
                wwn 1:p   -1.0000e+00  0.0000e+00  2.0899e+00  8.6263e-01  8.4175e-01
                           4.1617e-01  2.5085e-01  1.8402e-01  8.6684e-02  6.1191e-02
                           4.4617e-02  2.6756e-02  1.7136e-02  1.1050e-02  7.4140e-03
                           4.9116e-03  3.2181e-03

      ••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
N36   dump no.  2 on file   runtpe          nps =   10000          ctm =   8.74

               3 warning messages so far.

N37   run terminated when     10000 particle histories were done.
      mcnp     version 3a    04/16/86        9406cc14n1  y 04/30/86 11:54:20                probid = y 04/30/86 11:23:49
```

Notes:

𝒩1: This first line of the output file indentifies the code and when the program was run. LD=xx identifies the date the code was loaded, which helps identify the modifications that have been made to the code version. The next item is the user's charge and program code, followed by N1 (public MCNP) or N2 (user-modified version). The next letter identifies the computer the problem was run on. The last two entries are the date and time the run was made.

𝒩2: This is the name of the input file.

𝒩3: The numbers in this first column are sequential line numbers for the input file. They will be useful if you make changes to the file with an editor.

𝒩4: Defines a source at the point 0,0,0 on surface 2. The particles will enter cell 2. The entry CEL=2 is not needed, but if you choose to use it and type in the wrong cell number, the code will terminate. The weight of each source particle is 1, the default. VEC and DIR determine the starting direction. In this problem, the source is monodirectional in the y direction. All source particles have a starting energy of 14.19 MeV.

𝒩5: The PWT card controls the number and weight of neutron-induced photons produced at neutron collisions.

𝒩6: Energy bins for all tallies but F6 and F12. 13I means put 13 interpolates between 1 and 15 MeV. These energy bins are printed in PRINT TABLE 30.

-Table 10-

𝒩7: All source variables defined explicitly or by default are printed. The order of sampling of the source variables is also printed, which is important for sources that are dependent on functions.

-Table 30-

𝒩8: This entry identifies which particle type and tally type is used (neutron or photon).

𝒩9: This warning is generated because the upper limit of the E0 card of 20 MeV is higher than the maximum energy specified on the PHYS:N card.

𝒩10: The energy bins are specified by the E0, E6, and E12 cards. Tallies F1, F11, F16, F26, and F34 have energy bins specified by the E0 card. F6 and F12 have energy bins specified by the E6 and E12 cards.

𝒩11: This warning is generated because the lower limit of the E0 card of 1E−04 MeV is lower than the minimum energy specified by the PHYS:P card, 1E−03. The first two bins are below 1E−03 MeV.

−Table 50−

𝒩12: A cell can be composed of physically separate regions or pieces joined with the union operator. Improperly defined cells can be composed unintentionally of more than one piece (for example, a surface is extended unknowingly and forms a cell). If a cell is composed of more than one piece, a warning message is given and you should verify that the number of pieces is correct or incorrect.

−Table 60−

𝒩13: If you know the mass or volume of a geometry or parts of it, you can compare the known volume or mass with what MCNP calculates to verify the correctness of your geometry. Be careful, however, that volumes or masses that MCNP cannot calculate (but supplies a value such as unity) do not affect the totals.

−Table 70−

𝒩14: These entries are the surface coefficients used by the code and are not necessarily the entries on the surface cards.

−Table 72−

𝒩15: These are the temperatures calculated by MCNP for cells 2−17. Because there was not a TMPn:N card in the input file, room temperature (2.53e−08 MeV) is assumed. Cell 1 has zero importance and is therefore not listed.

𝒩16: The minimum and maximum source weights are printed here because they are sometimes dependent upon cell volumes and cannot be printed earlier. When the source is biased in any way, there will be a fluctuation in starting source weights. The minimum source weight is

used in the weight cutoff game when negative weight cutoffs are entered on the CUT cards. By playing the weight cutoff game relative to the minimum source weight, the weight cutoff in each cell is the same regardless of starting source weight. Note that if the source weight can go to zero, the minimum source weight is set to 1.E−10 times the value of the WGT parameter on the SDEF card.

−Table 100−

$\mathcal{N}$17: The cross−section table list shows the nuclear data used in the problem. The C appended to the neutron data indicates continuous energy. A D would indicate discrete reactions. A P indicates photon data. Note that photon data is all elemental (1000.01P) rather than isotopic (1001.50C). Warnings are printed in MODE N P problems if the photon production cross sections are unavailable or are in the less accurate equiprobable bin format.

$\mathcal{N}$18: If a neutron is born at an energy greater than E_{max} as set by the PHYS:N card, that neutron is rejected and the event (such as fission) is resampled until an energy below E_{max} is obtained.

$\mathcal{N}$19: Any neutron cross sections outside the energy range of the problem as specified by the PHYS:N and CUT:N cards are deleted.

$\mathcal{N}$20: The lower energy limit for most neutron cross−section data is 10^{-11} MeV (see Appendix G). The default lower energy limit for neutron problems (CUT:N card) is 0 MeV. Thus the neutron energy cutoff is below the data cutoff, possibly necessitating extrapolation of the data. If a higher cutoff is specified on the CUT:N card, this warning goes away, but in MODE N P problems a new warning is issued because photon production is ignored below the neutron energy cutoff.

$\mathcal{N}$21: At the end of the XACT overlay and before histories are started in the MCRUN overlay, the first dump is made to the RUNTPE file. This dump contains all the fixed information about the problem, namely the problem specification and all nuclear data. Subsequent dumps to RUNTPE will contain information that accumulates as histories are run, such as tally information and particle statistics for summary and ledger tables.

-Table 110-

.N22: This table gives starting information about the first 50 source particles. X, Y, and Z tells the initial position. CELL identifies what cell the particle started in or was directed into. SURF identifies what surface the particle started on, if any. U, V, and W identifies the starting direction cosines. The starting time, weight and energy of the particles are also given.

-Ledger Table-

N23: This is the summary page of the problem. It is a balance sheet with the left side showing how particle tracks, weight, and energy were created and right side showing how they were lost. The ledger table is for accounting only, because most entries, such as "tracks," have no physical meaning and trying to give physical interpretation to these numerical quantities may be dangerous. The weight and energy columns contain the physical results. Because the ledger contains net creation and loss, physical interpretation must be done with care.

N24: The creation from weight cutoff represents the weight and energy gained from winning the weight cutoff Russian roulette game. No tracks are created because the original track continues with an increased weight.

N25: 36474 represents the increased number of tracks obtained and banked from cell splitting, which occurs when the ratio of importances of the cell entered to the cell exited is greater than one. If the ratio is less than one, Russian roulette is played. If the track survives the roulette, its increase in weight and energy are recorded as a gain. If it loses, it is recorded as a loss in all three categories on the loss side of the table.

N26: Any tracks that enter a cell of zero importance are considered to have escaped the geometry and are recorded here. This is the physical leakage from the system. The precision of this result is unknown because no relative error is calculated as is with a tally.

N27: In a scattering event only energy is changed. Energy difference = energy in - energy out. If this difference is positive, it is entered on the loss side; if negative, as in fission or a thermal neutron upscatter, it is entered on the creation side. Thermal neutron

scatter always results in a small energy gain or loss. Higher energy scatter usually is an energy loss mechanism. This energy is only for the track being followed. If the collision is a fission or (n,xn), the tracks in addition to one outgoing track are recorded in the three creation columns of the fission and (n,xn) rows.

$\mathcal{N}$28: Tracks are lost to capture only if the analog capture option is used (PHYS card). In this problem, implicit capture was used to remove a fraction of each particle's weight at each collision. The energy lost is the incident energy of the particle times the weight lost to capture. The weight lost to absorption is a physically meaningful quantity. No relative error is calculated.

$\mathcal{N}$29: Loss to weight cutoff comes from losing the weight cutoff Russian roulette game. With perfect sampling, the weight and energy lost here should equal the weight and energy gained in N24. What is accumulated in the three loss entries is the number, weight, and weight times energy of the tracks lost to weight cutoff. The weight entry in the table is normalized by the number of source particles and the energy entry by the total weight of source particles. Thus the average weight of each track lost to the weight cutoff is: weight entry * NPS/number of tracks lost: 0.08109 * 10000 / 8528 = 0.09509. The small average track weight is caused by the scaling of the weight cutoff criteria by the ratio of the source cell importance to the collision cell importance. The average energy of a particle lost to weight cutoff per source particle is: energy entry/weight entry * average source starting weight: (0.04498/0.08109) * 1.0 = 0.55469. The same normalizing procedure applies to <u>all</u> energy entries in both the creation and loss columns of this table.

$\mathcal{N}$30: Loss to importance sampling results from losing the Russian roulette game played when crossing a surface into a cell of lower importance. The weight and energy losses should agree with gains in $\mathcal{N}$25 with perfect sampling.

$\mathcal{N}$31: Note that the total gain and the total loss of the track quantities is exactly the same in all problems.

$\mathcal{N}$32: The average lifetime of a neutron is extremely problem dependent and does not necessarily represent anything physical in the context here.

The categories are the average time it took a neutron to escape from the geometry or to be captured, the weighted average of capture or escape lifetimes (which is the best estimate of the neutron lifetime in the problem), and the average lifetime to any terminal event such as weight cutoff, importance sampling, or energy cutoff, in addition to capture and escape. These lifetimes are relative to zero time, which is not necessarily the time a neutron is started from the source. These numbers may be of use in setting a time cutoff or in getting a better feel for what is happening in the problem.

$\mathcal{N}$33: The second entry of the net multiplication is the relative error or the multiplication corresponding to one standard deviation. In this problem, the net multiplication, which is the sum of the source weight and the weight from (n,xn) reactions, is 1.0159 ±0.1%. The net multiplication is not the criticality eigenvalue k_{eff} of the system. See page 155 for further discussion of this subject.

$\mathcal{N}$34: The photon annihilation that leads to a creation of weight is pair production with an associated energy loss. For example, a 5–MeV photon undergoing pair production results in one photon of energy 0.511 MeV but with twice the weight of the photon entering the event. There is a resultant loss of 3.978 MeV of energy that is deposited by the electron–positron pair slowing down to annihilation. The energy lost in this problem is 0.52808, as shown under pair production in the energy loss column.

$\mathcal{N}$35: For a MODE N P problem, the photon lifetime is relative to zero time and not to the time when the photon was produced. Thus the photon lifetime may be effectively the mean time to creation because a photon's lifetime is so much shorter than a neutron's lifetime.

–Table 126–

$\mathcal{N}$36: Tracks entering a cell refers to all tracks entering a cell, including source particles. If a track leaves a cell and later reenters that same cell, it is counted again.

$\mathcal{N}$37: Population in a cell is the number of tracks entering a cell plus source particles and does not include reentrant tracks. Comparing N36 to N37 will indicate the amount of back scattering in the problem. An often successful rule of thumb for choosing importances is to select

them so that population is kept roughly constant in all cells between the source and tally regions. Information, once lost, cannot be regained. The 13548 particles in cell 8 can contain no more information than the 7195 particles in cell 6 because all particles in cell 8 are progeny of the particles in cell 6. Oversplitting or undersplitting has occurred between cell 6 and cell 8.

$\mathcal{N}$38: The number of collisions in a cell is important for a detector tally or anything involving collision rate. A lack of collisions may indicate a need to force them. This quantity is not normalized by cell volume. In some problems most of the computer time is spent modeling collisions. Cells with excessive numbers of collisions are possibly oversampled. This often happens when many thermal neutrons rattle around and contribute little of significance to the problem solution. In such cases energy–dependent weight windows are most effective, followed by energy roulette, exponential transform, time cutoff, or energy cutoff. Note that the last two methods may introduce a bias into the problem. Subdividing the cell into smaller cells with different importances also is effective.

$\mathcal{N}$39: The collision times the weight of the particles having the collisions is an indication of how important the collisions were.

$\mathcal{N}$40: The next four items are determined in the HSTORY subroutine after the distance D to the next collision or surface is determined. The time DT to traverse this distance is determined from DT=D/VEL where VEL is the speed of the particle. Furthermore, the flux Φ is equal to the number density n(E) times the speed.

The energy ERG averaged over the number density of particles is determined by

$$\frac{\iint n(E,t)\cdot E\ dE\ dt}{\iint n(E,t)dE\ dt} \quad \text{or} \quad \frac{\sum (\mathrm{WGT{*}DT{*}ERG})}{\sum (\mathrm{WGT{*}DT})}$$

$\mathcal{N}$41: The energy averaged over the flux density is

$$\frac{\iint \Phi(E,t)\cdot E\ dE\ dt}{\iint \Phi(E,t)dE\ dt} \quad \text{or} \quad \frac{\sum (\mathrm{WGT{*}D{*}ERG})}{\sum (\mathrm{WGT{*}D})}$$

It is very difficult, and perhaps meaningless, to determine an average energy because a large spectrum involving several orders of

magnitude is frequently involved leading to the problem of representing this by one number. That is why it has been calculated by the two methods of items 40 and 41. If the number-averaged energy is significantly lower than the flux-averaged energy (as is true in this problem), it indicates a large number of low-energy particles. As the energy cutoff in this problem is raised, these two average energies come into closer agreement.

$\mathcal{N}$42: The relative average track weight is $I_c \sum$ (WGT•D) / ($I_s \sum$ D), where I_c and I_s are the importances of the cell and the source cell. By making the average track weight relative to the cell importance, the weight reduction from importance splitting is removed. For most problems with proper cell importances, the average track weight is constant from cell to cell and deviations indicate a poor importance function. The variation in average track weight for the photons in the following table suggests that the photon importances (all unity) are poor. With weight windows, the average track weight should be within the weight window bounds.

$\mathcal{N}$43: The average track mean free path is

$$\frac{\int \Phi(E)/\Sigma_t(E)\ dE}{\int \Phi(E)dE} = \frac{\sum \mathrm{WGT{\bullet}D/TOTM}}{\sum \mathrm{WGT{\bullet}D}},$$

where TOTM = $\Sigma_t(E)$ is the total macroscopic cross section. The mean free path is strongly dependent upon energy and so this average mean free path may be quite meaningless. A rule of thumb for guessing at importances is that they should double approximately every mean free path. This is usually a very poor rule, but it is sometimes better than nothing. The average track mean free path is thus useful for making poor guesses at cell importances. It is also useful for determining the ficticious radius of point detectors, the outer radius of DXTRAN spheres, exponential transform stretching parameters, the necessity of forced collisions, etc. Occasionally this quantity may even provide physical insight into your problem.

$\mathcal{N}$44: For photons, the number-weighted energy and flux-weighted energy are equal because a photon has a constant velocity regardless of energy. See $\mathcal{N}$36 - $\mathcal{N}$43.

-Table 130-

𝒩45: The next six pages (three for neutrons and three for photons) show all possible ways a particle's weight may be changed in each cell. In addition to telling you what is happening to the particle and where, this information can be useful in debugging a problem.

Note that the neutron weight entering cell 17 is 0.01949, whereas in Table 126 the average relative track weight in cell 17 is 0.66068. This apparent discrepancy is resolved by realizing that the average weight in Table 126 is for a track, while it is for a history in Table 130. Furthermore, in Table 126 the weight is relative, whereas it is absolute here in Table 130. If the average track weight is multiplied by the tracks entering cell 17 (7554) and then divided by both the number of source particles (10,000) and the importance ratio (32), the two weights are in close agreement. Most of the totals over the cells can be compared directly with the weight gain, loss or difference in the Ledger Table. The average value of $\bar{\nu}$ in a problem with fissionable material can be obtained by taking the ratio of fission neutrons to fission loss in the neutron physical events table.

-Table 140-

𝒩46: The activity of each nuclide per cell can tell you how important various nuclides, such as trace elements, are to the problem and may aid in selecting cross-section libraries when memory is limited.

𝒩47: This table is the activity summed over all cells in the problem.

𝒩48: This column shows the total number of photons produced by each isotope in the problem. The earlier entries in this column show photon production per isotope in each cell.

𝒩49: This table is printed only for MODE N P. It gives you an idea of how many photons were produced in each cell and the energy spectrum of the photons averaged over the problem. Because photons are produced only at neutron collisions, there is a correlation between the number of collisions in a cell, the PWT card, and this table. The previous table showing the photon activity for the problem includes isotope-dependent neutron-induced photon production information.

TALLY AND TALLY FLUCTUATION CHARTS

50: All tallies here are caused by the F1, F11, F12, F6, F16, F26, and F34 cards in the input file. The F11 tally gives the photon current summed in both directions integrated over a surface. This tally says that between 13 and 14 MeV, the current is 1.84613E-04 ±17% within one standard deviation.

51: The average tally per history includes energy- and time-dependent multipliers and some constant multipliers, but excludes most constant multipliers. In Tally 11 the total tally is the same as the average tally per history. In Tally 16 the average tally per history has not been divided by the cell mass and is different from the total tally.

In Tally 12 the largest tally, 2.92667, is 165 times larger than the average tally, indicating there is no extremely large weight fluctuation problem. If there were a great difference between the largest and average tally, the large weight particles would represent important phenomena that have been undersampled and/or poor variance reduction technique selection. To understand what causes the large weight particles, the history number of the largest is printed so that this history can be rerun to get its event log. When the undersampled event is identified, the variance reduction should be modified and the problem rerun.

Improved variance reduction usually causes fewer source histories to be run per minute because more time is spent sampling the formerly undersampled important phenomena outside the source. The final result will be an improved (higher) FOM and a lower largest/average tally ratio. As the largest/average tally ratio approaches unity, the problem approaches an ideal zero variance solution.

In practice, performing the steps discussed above is an art usually beyond all but the most experienced users and is often difficult, time-consuming, frustrating, and sometimes unsuccessful. An alternative is to let MCNP determine the better importance function for the next run with the weight window generator, as has been done in this problem. Use of the generated weight windows printed in PRINT TABLE 190 caused a factor of three improvement in problem efficiency when the problem was rerun and reduced the largest/average tally ratio to 35.

$\mathcal{N}52$: If any tally is modified by a divisor (such as area, volume, or mass) or a multiplier (such as from the FM card), the modifier is printed here. You should make sure the modifier is what you expect.

$\mathcal{N}53$: The tally fluctuation charts always should be studied to see how stable or reliable the tally mean, relative error, and FOM are, indicating how the problem is converging as a function of history number, NPS. The FOM is defined as $1/\sigma^2 t$, where σ is the relative error and t is the computer time in minutes. In a well-behaved problem, t is proportional to the number of histories run, N, and σ is proportional to $1/\sqrt{N}$. Thus the FOM should rapidly approach a constant value as it does in this problem. Big changes in the FOM indicate sampling problems that need attention.

The order of printing tallies is: neutron, photon, and combined. Notice that the combined heating tally F6 is exactly the sum of the neutron, F16, and photon, F26, heating tallies.

–Table 190–

$\mathcal{N}54$: This table is a list of the lower weight window bounds generated by the WWG card. These window bounds are themselves estimated quantities and must be well converged or they can cause more harm than good. When well converged, they can improve efficiency dramatically. Use of these printed weight windows results in an increase of three in the FOM for tally 12 when the problem was rerun. Note that the number of histories per minute is often lower in the more efficient problem because more time is spent sampling important regions of the problem phase space.

These weight windows were chosen to optimize tally 12 as specified on the WWG card. In the subsequent run using these weight windows, the FOM of tally 12 improved by three as did the other photon tallies, and tallies 16 and 34 were slightly degraded. The weight window generator optimizes the importance function for one tally at the expense of all others, if necessary. The generator can optimize on a union of tallies to ameliorate this problem.

The lower bound for the photon weight window in cell 2 is zero, meaning that no photon in cell 2 ever contributed to tally 12 in this run. If the zero is unchanged in the run using these windows, the weight cutoff game will be played in cell 2, sometimes with disastrous consequences. Thus a guess should be made for a lower bound rather than leaving the zero value. A good guess is 10, which is several

times higher than the weight window generated for its nearest neighboring cell.

The generated weight windows may be thought of as a forward adjoint solution and thus can provide considerable insight into the physics of a problem. Low weight windows indicate important regions. A low window on a cell bounding the outside world often indicates that the geometry was truncated and more cells need to be added outside the present geometry. Weight windows that differ greatly between adjacent cells indicate poor weight window convergence or, more likely, a need to subdivide the geometry into smaller phase space units that will have different importances.

Energy dependent weight windows are also available.

–Table 200–

$\mathcal{N}$55: The weight window cards from the weight window generator can, with some file editing, be used to replace the IMP:N and IMP:P cards in the next run of this problem. Zero windows should be replaced with a good guess. Windows differing greatly from those in neighboring cells should be replaced also (there are no such cases in this problem). In MODE N P problems the WWN1:P card should be copied to a PWT card for optimum photon generation in the next run.

$\mathcal{N}$56: With this initial run there are two dumps on the RUNTPE. The first dump occurs at the end of the XACT overlay. The second dump is done at the problem end. A continue–run will pick up from this second dump and add a third dump to the RUNTPE when it finishes. CTM is the computer time in minutes used in the transport portion of the problem.

$\mathcal{N}$57: One or more reasons are always given as to why the run was terminated. If there are no errors, most runs terminate after the desired number of particles are run or by a time limit.

III. SHELL PROBLEM AND OUTPUT

This simple problem illustrates how to use and interpret results from detectors. The problem consists of a spherical shell of concrete 390 cm in outer radius and 360 cm in inner radius. A 14.19 MeV point isotropic neutron source is in the center void region. Photon production is ignored (MODE N) and the neutron lower energy cutoff is 12 MeV for convenience. The object is to calculate the flux transmitted through the sphere. Both a

surface flux tally, F2, and a ring detector around the shell, F5Y, are used.

Even though this is a simple problem, it is difficult, and even inappropriate, for a point detector. There are too few collisions near the detector and half the detector tally comes from the .01% of big scores one hundred times greater than the average score. Detectors are inappropriate whenever particles can be transported readily to the region of interest and another type of tally, such as the F2 surface flux tally, can be used. In this problem the F2 tally takes advantage of spherical symmetry, whereas the ring detector only takes advantage of cylindrical symmetry. Also, detectors do not work well close to or in scattering regions.

The following notes apply to the shell problem output file, which is short because there is no PRINT card.

```
mcnp     version 3a     ld=04/16/86        9406cc14n1  3 04/26/86 12:54:57                probid = 3 04/26/86 12:54:57
*********************************************************************************
inp=shell
M1        message:  outp=outcon runtpe=runcon

          concrete shell problem to illustrate poor detector usage
          c
          c
          1      0            -1
          2      1 -2.2505    1 -2
          3      0            2

          1      so 360
          2      so 390

          imp:n  1 1 0
          c      the following is los alamos concrete
          m1       1001      -.004532
                   8016      -.512597
                  11023      -.011553
                  12000      -.003866
                  13027      -.035548
                  14000      -.360364
                  19000      -.014219
                  20000      -.043546
                  26000      -.013775
          sdef   pos=0 0 0 wgt=1 cel=1 erg=14.19
          f2:n   2
M2        f5y:n  0 389.9 -0.50
M3        cut:n  100000 12.0
M4        dd5    -1.e-8 1.e4
          nps    100000
```

cells

print table 60

	cell	mat	atom density	gram density	volume	mass	pieces	neutron importance
1	1	0	0.00000e+00	0.00000e+00	1.95432e+08	0.00000e+00	1	1.00000e+00
2	2	1	7.18969e-02	2.25050e+00	5.30427e+07	1.19372e+08	1	1.00000e+00
3	3	0	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0	0.00000e+00
total					2.48475e+08	1.19372e+08		

minimum source weight = 1.0000e+00   maximum source weight = 1.0000e+00

```
cross-section tables                                                                          print table 100

    table     length

                         tables from file rmccs

  1001.50c      144
  8016.50c     5133
 11023.51c     1433
 12000.51c     5440
 13027.50c     5308
 14000.51c     5010
 19000.51c     1626
 20000.51c     6466
 26000.55c    19026

 total        49786

neutron cross sections outside the range from 1.2000e+01 to 1.0000+123 mev are expunged.


use of dynamically allocated storage

    general                 3907
    tallies                    9
    cross sections         49786

    total                  53702

************************************************************************
dump no.  1 on file   runcon            nps =        0          ctm =    0.00

det       t          wgt         psc        amfp        ddetx       radius        erg       cell          nps  nch           ijk    n an
 1 1.3021e-04 2.0170e-03 3.2027e+00 2.0457e-01 1.7665e+00-5.0000e-01 1.4032e+01     2          8885    1 183298564341817

det       t          wgt         psc        amfp        ddetx       radius        erg       cell          nps  nch           ijk    n an
 1 3.0949e-04 8.3311e-03 1.8949e+00 4.6546e-01 4.0755e+00-5.0000e-01 1.2624e+01     2         29459    3 217407871021169

det       t          wgt         psc        amfp        ddetx       radius        erg       cell          nps  nch           ijk    n an
 1 4.9898e-04 4.4632e-03 5.5375e+00 4.5232e-01 3.9026e+00-5.0000e-01 1.4132e+01     2         35504    1 139323807099901

det       t          wgt         psc        amfp        ddetx       radius        erg       cell          nps  nch           ijk    n an
 1 2.7542e-04 2.0320e-03 6.7157e+00 2.6245e-01 2.2648e+00-5.0000e-01 1.4110e+01     2         49190    2 121148804214133

det       t          wgt         psc        amfp        ddetx       radius        erg       cell          nps  nch           ijk    n an
 1 1.4826e-04 3.8012e-03 1.9337e+00 3.9768e-01 3.4329e+00-5.0000e-01 1.4071e+01     2         50020    1 228022962454349

det       t          wgt         psc        amfp        ddetx       radius        erg       cell          nps  nch           ijk    n an
 1 2.2430e-04 1.5390e-03 7.2164e+00 1.7781e-01 1.5339e+00-5.0000e-01 1.4153e+01     2         83303    1 256445424341377
```

run terminated when 100000 particle histories were done. 3 04/26/86 12:57:35
concrete shell problem to illustrate poor detector usage probid = 3 04/26/86 12:54:57

.18

ledger of net neutron creation and loss (for accounting only)

	tracks	weight (per source particle)	energy (per source particle)
source	100000	1.0000e+00	1.4190e+01
scattering	0	0.	0.
fission	0	0.	0.
(n,xn)	2051	1.4655e-02	3.3604e-02
forced collision	0	0.	0.
weight cutoff	0	2.2643e-04	2.8655e-03
weight window	0	0.	0.
cell importance	0	0.	0.
energy import.	0	0.	0.
dxtran	0	0.	0.
exp. transform	0	0.	0.
total	102051	1.0149e+00	1.4226e+01

	tracks	weight (per source particle)	energy (per source particle)
escape	14381	1.1286e-01	1.5694e+00
scattering	0	0.	5.9121e+00
capture	0	2.4703e-01	3.4379e+00
energy cutoff	87561	6.5474e-01	3.3038e+00
time cutoff	0	0.	0.
weight cutoff	109	2.4928e-04	3.1613e-03
weight window	0	0.	0.
cell importance	0	0.	0.
energy import.	0	0.	0.
dxtran	0	0.	0.
exp. transform	0	0.	0.
dead fission	0	0.	0.
total	102051	1.0149e+00	1.4226e+01

neutron tracks per source particle	1.0205e+00
neutron collisions per source particle	1.9092e+00
total neutron collisions	190916
net multiplication	1.0147e+00 0.0003

average lifetime, shakes	
escape	7.6403e+00
capture	7.2856e+00
capture or escape	7.3968e+00
any termination	7.2402e+00

cutoffs	
tco	1.0000e+05
eco	1.2000e+01
wc1	-5.0000e-01
wc2	-2.5000e-01

computer time so far in this run	1.43 minutes
computer time in mcrun (4c0)	1.42 minutes
source particles per minute	7.0551e+04
field length	239178
random numbers generated	2170879
last starting random number	31746835266493
next starting random number	140183165497513

total neutrons banked	0
per source particle	0.0000e+00
total photons banked	0
per source particle	0.0000e+00
maximum number ever in bank	0
bank overflows to backup file	0

range of sampled source weights = 1.0000e+00 to 1.0000e+00

neutron activity in each cell

print table 126

cell progr name	cell probl name	tracks entering	population	collisions	collisions * weight (per history)	number weighted energy	flux weighted energy	average track weight (relative)	average track mfp (cm)
1	1	101443	100000	0	0.0000e+00	1.4173e+01	1.4174e+01	9.9552e-01	0.0000e+00
2	2	101443	100000	190916	1.6887e+00	1.3913e+01	1.3923e+01	8.8419e-01	8.6331e+00
total		202886	200000	190916	1.6887e+00				

```
17   tally   2         nps =   100000
              neutron tally type 2:   particle flux averaged over a surface.       units   1/cm**2

              areas
                  surface:         2
                           1.91134e+06

     surface  2
                  7.04637e-08 0.0089

     for the tally-fluctuation-chart bin:
     average tally per history = 1.34680e-01       largest tally = 1.77483e+01
     largest/average = 1.31781e+02                 nps of largest =    93754
```

```
     tally   5          nps =    100000
                neutron tally type 5:   particle flux at a ring  detector.           units   1/cm**2

.18  detector symmetric about y-axis located at y = 0.00000e+00 with radius = 3.89900e+02
                        7.66193e-08 0.1031

     detector symmetric about y-axis located at y = 0.00000e+00 with radius = 3.89900e+02
.19  uncollided neutron flux
                        1.63360e-08 0.0000

.110 detector diagnostics                                cumulative         tally          cumulative
                                                          fraction of        per            fraction of
             maximum tally       transmissions            transmissions      history        total tally

a            1.00000e-08            46506                  0.30718        1.71054e-10       0.00223
             2.00000e-08           100978                  0.97415        1.64758e-08       0.21727
             5.00000e-08             1072                  0.98123        3.51708e-10       0.22186
             1.00000e-07              635                  0.98542        4.59905e-10       0.22786
             2.00000e-07              571                  0.98919        8.26684e-10       0.23865
             5.00000e-07              579                  0.99302        1.84784e-09       0.26277
             1.00000e-06              369                  0.99546        2.62989e-09       0.29709
b            1.00000e-05              598                  0.99941        1.63251e-08       0.51016
c            1.00000+123               90                  1.00000        3.75313e-08       1.00000

d    number of tallies =    151398
e    average tally per history = 7.66193e-08          largest tally = 4.98978e-04
f    largest/average = 6.51243e+03                    nps of largest =     35504


     summary of detector contribution by cell

             cell      hits      tally per history     weight per hit

g         1     1    100000       1.63360e-08g         1.63360e-08
h         2     2     51398       6.02833e-08          1.17287e-07
             total   151398       7.66193e-08          5.06078e-08
```

1 tally fluctuation charts

	tally 2			tally 5		
nps	mean	error	fom	mean	error	fom
8000	6.95343e-08	0.0298	9855	8.11040e-08	0.1831	261
16000	6.99310e-08	0.0214	9544	7.67498e-08	0.1592	173
24000	6.96668e-08	0.0175	9556	7.18077e-08	0.1277	179
32000	6.95758e-08	0.0153	9403	7.80233e-08	0.1593	87
40000	7.00281e-06	0.0137	9420	8.52620e-08	0.1908	48
48000	7.01166e-08	0.0124	9541	7.91596e-08	0.1719	50
56000	7.00945e-08	0.0116	9345	8.45621e-08	0.1544	53
64000	7.04371e-08	0.0110	9090	8.15469e-08	0.1409	56
72000	7.02169e-08	0.0104	8998	7.87881e-08	0.1303	58
80000	7.01039e-08	0.0099	9001	7.76028e-08	0.1204	61
88000	7.01381e-08	0.0095	8966	7.82946e-08	0.1137	62
96000	7.05119e-08	0.0091	8900	7.63804e-08	0.1072	64
100000	7.04637e-08	0.0089	8947	7.66193e-08	0.1031	66

••

dump no. 2 on file runcon nps = 100000 ctm = 1.42

run terminated when 100000 particle histories were done.

mcnp version 3a 04/16/86 9406cc14n1 3 04/26/86 12:57:35 probid = 3 04/26/86 12:54:57

Notes:

N1: The MESSAGE block is used to rename the output file OUTCON and the continue-run file RUNCON.

N2: The ring detector is placed just inside the outer concrete shell boundary. Never put a point or ring detector exactly on a surface boundary, especially if the cell on one side has a zero importance as it does here. Roundoff errors may eliminate part of the detector tally because some detector contributions may be killed erroneously by entering the zero importance cell.

The fictitious sphere radius of 0.5 mean free paths (approximately 4.3 cm) assumes a uniform isotropic flux within the sphere. Although this assumption will smooth out the detector response, it is false. The fictitious sphere should never be in more than one material medium as it is here because the material is assumed to be uniform throughout the sphere. This fictitious sphere does not distort answers only because the vacuum half of the sphere has zero importance.

N3: The 12 MeV energy cutoff makes this problem run fast by eliminating all physics below 12 MeV. If physics below 12 MeV and after the the time cutoff of 100000 shakes (1 ms) is important, these tallies will be wrong because the answer is truncated. In some problems, tallies below the energy cutoff may be unimportant but fission or photon production below the cutoff affects answers above the cutoff. You must be very careful with problem cutoffs because they can truncate your answers.

N4: The DD card roulette games are played in all detector problems unless specifically turned off. Results are sometimes disastrous. Here the DD card has been carefully chosen rather than using the defaults. The −1.e−8 entry indicates that all potential detector contributions to the ring detector will be rouletted if the anticipated score will be less than 1.e−8. The value 1.e−8 was selected after a short run indicated that the average score was on the order of 1.e−7. Thus detector contributions less than a tenth of the average score will be rouletted so as not to waste time on them. Rouletting small scores significantly reduces running time.

The second entry on the DD card, 1.e4, causes a diagnostic message to be printed when a tally greater than 1.e4 * 1.e−8 = 1.e−4 is made. If this value is too high, valuable diagnostic messages are lost. If it is too low, there are so many messages that they are hard to read and understand.

.15: This is the diagnostic print regulated by the second entry on the DD card. A similar diagnostic is available for DXTRAN contributions. An explanation of the various columns follows. Recall that the contribution to a detector is usually WGT*PSC*EXP(−AMFP)/(2*PI*R^2). However, all these large scores are made within the fictitious sphere, so the formula is modified as discussed on page 89. The tally is proportional to WGT*PSC either way.

DET: The print is for detector number 1, the only detector in this problem. The detector number corresponds to the order in which detector tallies are printed, not to the tally number.

T: A tally of 1.3021E−04 was made to the detector, a value greater than the 1.E−4 criterion for printing the diagnostic.

WGT: The weight of the particle in the collision that made the detector score was 2.0170E−3, which is lower than the average weight of particles in cell 2 where the collision occurred.

If the value of WGT were abnormally high, the large detector score could be attributed to a bad importance function. Perhaps the importances or weight windows or other variance reduction parameters could have been set poorly in such a problem so as to allow high weight particles.

PSC: This is the value of the probability density function for scattering from the collision point in the direction of the detector. This is not a probability. Its integral over all polar angles is unity, but values of PSC may exceed unity, as it does here. Because the PSC is high, perhaps the reason the detector contribution is large is that the scattering nuclide has sharply forward−peaked scattering and the particle incident to the collision was headed almost straight toward the detector. If such contributions are important and undersampled, causing large tallies to destroy detector convergence, they must be sampled more. One way is to force collisions in the vicinity of the detector.

AMFP: This is the attenuation coefficient of mean free paths for contributing to the detector. If it approaches its minimum value of zero, perhaps the large detector contributions are coming from a

streaming path to the detector. In this example, AMFP is small because the collision is close to the detector. However, AMFP is irrelevant here because the collision is inside the fictitious sphere radius and the modified detector contribution formula is used.

DDETX: This is the distance R separating the collision point and the detector. The concrete shell is 30 cm thick and the collision occurred only 1.7665 cm from the detector. As seen from the usual detector formula given above and in Chapter 2, small values of R (DDETX) can be a major cause of large detector contributions because such contributions are inversely proportional to R^2. In this example, R is not used because the collision is inside the fictitious sphere.

RADIUS: The input fictitious sphere radius in mean free paths.

ERG: This is the energy of the pseudoparticle emerging from the collision and heading for the detector. In this problem all particles have an energy between 12 and 14.2 MeV, so this information is not too helpful. In other problems it may be that particles in a certain energy range are more likely to contribute large scores to the detector and will be shown by this diagnostic print. In such a case the favored energy range should be sampled better by energy-dependent weight windows or energy roulette.

CELL: The cell in which the collision occurred.

NPS: The history number of this large detector contribution.

NCH: The number of collisions the particle has had so far. Because NCH includes progeny of other tracks of this particle, this number can be misleading.

IJK: Random number seed.

N: N or P indicates the particle type.

The large particle diagnostic print is located here rather than with the tally because it is accumulated as the run proceeds rather than at the end of the calculation. For this problem, six detector contributions were greater than the DD card criterion of 1.e−4.

Because all the large scores were inside the fictitious sphere radius – that is, very close to the detector – we can conclude that the large detector scores are caused by a few collisions near the detector. Thus, the way to improve the efficiency of this tally would be to sample more collisions close to the detector at the expense of sampling unimportant contributions far from the detector. Also, because PSC was greater than unity for all these large tallies, we conclude that the largest detector contributions were from particles headed toward the detector when they had collisions near it.

$\mathcal{N}$6: The Ledger Table contains much useful information. Note that most particles were killed by energy cutoff. Also, there were 190916 collisions for 100000 source histories. The net multiplication of 1.0147 is caused by (n,xn) reactions; the system is clearly not supercritical because there is no fissionable material. The weight per escaping source particle is 0.11286, which means that the flux on the outer shell of radius 390 cm is approximately $0.11286/(4*pi*390^2) = 5.90E-8$ neutrons/cm^2. This is somewhat lower than the tally results, as the neutrons are not exiting normal to the surface. Because only 14381 particles escape, penetration through the shell is poor and would be enhanced by splitting by a factor of about 7 to make the problem run more efficiently. The energy cutoff eliminated 87561 tracks, allowing the problem to run very fast.

$\mathcal{N}$7: The surface flux tally has a low relative error, indicating good convergence. The largest tally was 132 times the average tally, indicating that the tally is fairly well behaved and believable. Efficiency could have been improved by using weight windows so that all particles in the region close to the tally surface would have about the same weight, making the largest/average tally ratio smaller. Note that the average tally per history is not the final tally value because the surface area has not been divided yet. The average tally per history is the weight of particles escaping (see Ledger Table) modified by the angle relative to the surface normal (see F2 tally definition in Chapter 2). If a tally had scored without colliding, the score would be 1 because the normal to the surface is unity before being divided by the area. The largest tally is 17.7483, as history 93754 most likely crossed the outer shell at an almost tangential angle.

$\mathcal{N}$8: The detector tally has a relative error of 10.31%, placing its value of 7.66193e−08 within one standard deviation of the mean of the more precise surface flux tally, which was 7.04637e−08.

$\mathcal{N}$9: The direct contribution is exactly 1.63360E−8, or about 23% of the total contribution. The relative error is zero because all direct contributions are identical in this problem.

$\mathcal{N}$10: The letters on the left side of the detector diagnostics table correspond to the notes, (a), (b), etc., following.

There were 151398 detector contributions (d). 100000 were from the source (there were 100000 hits from cell 1)(g) and 51398 from collisions inside cell 2 (h). According to the Ledger Table, there were 190916 collisions. Thus the DD card roulette game eliminated 73% of them. Of the 51398 collisions that did contribute to the tally, 46506 (a) made a tally less than the 1.e−8 cutoff (it was conservatively estimated that their contribution would be higher so they were not rouletted). These 46506 transmissions to the detector contributed only 0.223% of the total tally (a). All but 90 detector contributions scored less than 1.e−5 (c). That is, 99.941% of the contributions provided 51.016% of the total tally (b). The remaining 90 transmissions scored the remainder of the total tally (c). That is, 0.059% of the transmissions contributed 49% of the tally (c). From the large score diagnostic (see note N5), it was concluded that these large scores all came from collisions near the detector. Therefore, if the problem were revised to sample collisions close to the detector, time would not be wasted on collisions that are too far from the detector to contribute much. The average tally per history (e) is the same as the total detector tally. The largest tally (e) is 6512 times higher (f) than the average and occurs on history 35504. The consequences show up in the tally fluctuation chart.

$\mathcal{N}$11: The tally fluctuation charts show how the tallies are converging as a function of history number.

The surface flux tally converges nicely with a high, roughly constant FOM. As expected, the convergence is such that the relative error is inversely proportional to the square root of the number of histories run.

The ring−detector tally does not converge nicely. The large score at history 8885 (see note N5 and the DD diagnostic print) drops the FOM from 261 to 173. An even bigger score at history 29459 lowers the FOM from 179 to 87 and causes the tally to increase from 7.18077e−8 to 7.80233e−08. The biggest score of all at history 35504 lowers the FOM from 87 to 48 and raises the score from 7.80223e−08 to 8.52620e−08. The error increases from 16% to 19% − not exactly decreasing as the inverse square root of the number of histories.

This tally fluctuation chart sharply contrasts the difference between well−behaving and poorly−behaving tallies.

COMMENTS: How should the SHELL problem be run better? First, detectors are inappropriate and should not be used. The shell should be divided into four spherically concentric geometrical regions with outwardly increasing importances of 1, 2, 4 and 8. Then for every source particle, approximately one would cross the outer surface of the shell and score, instead of the present 14381 out of 100000.

How could detectors be made to work better in this problem? In any problem with symmetry, a ring detector rather than a point detector should be used to at least take advantage of cylindrical symmetry and has been done already. The fictitious sphere radius could be made smaller to the point where the $1/r^2$ singularity made about as much difference as the fluctuation in PSC value. Perhaps this would be a fictitious sphere radius of 1 cm. Most importantly, the source direction could be biased to direct particles at the ring, causing a lot more collisions in the vicinity of the detector. To test this suggestion, the SDEF card was replaced with

```
sdef  vec 0 1 0 dir=d5 erg=14.19
si5  -1 -.8 -.5 -.2 -.1 -.05 -.02 -.01 0 .01 .02 .05
      .1 .2 .5 .8 1
sp5  0 2 3 3 1 .5 .3 .1 .1 .1 .1 .3 .5 1 3 3 2
sb5  0 .1 .6 3 5 10 30 50 100 100 50 30 10 5 3 .6 .1
```

This causes 20,000 times as many source particles to be directed toward the detector ($-.01 < \mu < .01$) as away from it along the y-axis ($\mu = \pm 1$.) The summary of this source bias in PRINT TABLE 10 is:

source entry	source value	probability of bin	biased probability	weight multiplier
1	-1.00000e+00	0.000000e+00	0.000000e+00	1.000000e+00
2	-8.00000e-01	1.000000e-01	2.516356e-04	3.974000e+02
3	-5.00000e-01	1.500000e-01	1.509814e-03	9.935000e+01
4	-2.00000e-01	1.500000e-01	7.549069e-03	1.987000e+01
5	-1.00000e-01	5.000000e-02	1.258178e-02	3.974000e+00
6	-5.00000e-02	2.500000e-02	2.516356e-02	9.935000e-01
7	-2.00000e-02	1.500000e-02	7.549069e-02	1.987000e-01
8	-1.00000e-02	5.000000e-03	1.258178e-01	3.974000e-02
9	0.00000e+00	5.000000e-03	2.516356e-01	1.987000e-02
10	1.00000e-02	5.000000e-03	2.516356e-01	1.987000e-02
11	2.00000e-02	5.000000e-03	1.258178e-01	3.974000e-02
12	5.00000e-02	1.500000e-02	7.549069e-02	1.987000e-01

13	1.00000e-01	2.500000e-02	2.516356e-02	9.935000e-01
14	2.00000e-01	5.000000e-02	1.258178e-02	3.974000e+00
15	5.00000e-01	1.500000e-01	7.549069e-03	1.987000e+01
16	8.00000e-01	1.500000e-01	1.509814e-03	9.935000e+01
17	1.00000e+00	1.000000e-01	2.516356e-04	3.974000e+02

The resulting tally fluctuation chart for a run of only 10000 histories is:

	tally 2			tally 5		
nps	mean	error	fom	mean	error	fom
1000	5.33803e-08	0.3350	560	7.46091e-08	0.1944	1663
2000	1.80363e-07	0.5541	104	7.36027e-08	0.1240	2068
3000	1.87444e-07	0.4596	100	8.20762e-08	0.1633	791
4000	1.60865e-07	0.4105	93	7.53539e-08	0.1351	860
5000	1.52925e-07	0.3640	95	7.26085e-08	0.1144	958
6000	1.35362e-07	0.3433	89	6.87670e-08	0.1016	1018
7000	1.21212e-07	0.3291	83	6.65661e-08	0.0908	1088
8000	1.10114e-07	0.3174	78	6.81893e-08	0.0817	1176
9000	1.01099e-07	0.3075	74	6.71028e-08	0.0776	1158
10000	9.76805e-08	0.2888	75	6.65175e-08	0.0716	1228

In 0.16 minutes tally 5 has obtained a relative error of only 7.16%, compared to the 10.31% error in the 1.42 minute run. The FOM is about 20 times higher for tally 5, but 120 times lower for tally 2. Thus the detector tally has been greatly enhanced, although at the expense of the surface flux tally. By spending more time on collisions by the detector, and consequently less time on the source and collisions far from the detector, proper interpretation of the shell problem results has shown us how to more efficiently run this detector problem.

IV. KCODE

The problem selected to illustrate the output from a criticality calculation is the one-dimensional model of the GODIVA critical assembly, composed of about 94% ^{235}U. This is one of several fast neutron critical assemblies discussed in LA-4208 entitled "Reevaluated Critical Specifications of Some Los Alamos Fast-Neutron Systems" by G. E. Hansen and H. C. Paxton (September 1964).

The MCNP input file to perform only the criticality calculation with no separate tallies needs to be only 11 lines long. The KCODE card indicates that this is a criticality calculation for the k_{eff} eigenvalue. To perform such a calculation with neutron-induced photon production, use the MODE N P card. Any tallies that are made in a criticality problem are normalized to the starting weight (default) or number of particles as defined by the user (see Chapter 2, section VIII for details). Tallies should be scaled for the appropriate steady state neutron generation rate.

Following is a partial listing of the output from a KCODE calculation. The pages selected emphasize the criticality aspects of the problem.

```
mcnp     version 3a    ld=04/30/86         9406cc14n2  2 05/14/86 12:13:47
 ***************************************************************************                                   probid = 2 05/14/86 12:13:47
 inp=godiva
.11        bare u(94) sphere   ref. la-4208, g. e. hansen and h. c. paxton, 1969, page 4
          1     10 0.048146 -1
          2     0 1

          1     so 8.7037

          imp:n 1 0
          m10   92235.50c 0.045217  92238.50c 0.0024355  92234.50c 0.0004935
.12       kcode 3000 1. 5 30
.13  warning.  tallies are normed per fission neutron for one generation.
.14       ksrc  0. 0. 0.
          print
          c
          c     tallies
          c
          f1:n 1
          f2:n 1
          f4:n 1
          f14:n 1
          fc14 total neutron fluence, total fissions, total fission neutrons,
                total neutron absorptions, and neutron heating (mev/gram)
          fq14 e m
.15       fm14 (2761.85) (-2761.85 10 (-6) (-6 -7) (-2)) (-0.053183 10 1 -4)
          f24:n 1
          fc24 total neutron fluence, total fissions, total fission neutrons,
                total neutron absorptions, and neutron heating (mev/gram)
          e24   1.67-4 4.54e-4 1.235e-3 3.35-3 9.12-3 0.0248 0.0676 0.184
                0.303 0.50 0.823 1.353 1.738 2.232 2.865 3.68
                6.07 7.79 10.0 12.0 13.5 15.0 20.0
          fq24 e m
          fm24 (2761.85) (-2761.85 10 (-6) (-6 -7) (-2)) (-0.053183 10 1 -4)
          f6:n 1
          f7:n 1
          c
          c     use the sixteen group hansen-roach energy structure as the default
          c
.18       e0    1-7 4-7 1-6 3-6 1-5 3-5 1-4 5.5-4 3-3 1.7-2 0.1 0.4 0.9 1.4 3 20
```

```
17          initial source from ksrc card.                                                                          print table 90

       original number of points                                         1
          points not in any cell                                  0
          points in cells of zero importance                      0
          points in void cells                                    0
          points in ambiguous cells                               0
       total points rejected                                             0
       points remaining                                                  1
       points after expansion or contraction                          3000
       nominal source size                                            3000

       initial guess for k(eff.)                                  1.000000

       cycles to skip before tallying                                    5

       total fission nubar data are being used.

       source distribution written to file srctp           cycle =    0


       material composition                                                                                         print table 40
.18     the sum of the fractions of material   10 was 4.814600e-02

       material
        number      component nuclide, atom fraction

          10        92235.50c,0.93916   92238.50c,0.05059   92234.50c,0.01025


       material
        number      component nuclide, mass fraction

          10        92235.50c,0.93860   92238.50c,0.05120   92234.50c,0.01020

       warning.     1 of the materials had unnormalized fractions.


.19         cell volumes and masses                                                                                 print table 50

       progr probl      atom           gram          input        calculated                                  reason volume
        name  name     density        density        volume         volume         mass        pieces     not calculated

          1     1   4.81460e-02    1.88030e+01    0.00000e+00    2.78185e+03    5.19312e+04       1
          2     2   0.00000e+00    0.00000e+00    0.00000e+00    0.00000e+00    0.00000e+00       0        infinite
```

```
M0  cross-section tables                                                                                    print table 100

     table     length

                           tables from file endf5p2

  warning.  nubar of  92234.50c may be either prompt or total.
   92234.50c    89429

                           tables from file rmccs

   92235.50c    49913                                                              total nu
   92238.50c    75725                                                              total nu

   total      215067

  warning.  neutron energy cutoff is below some cross-section tables.



  use of dynamically allocated storage

     general              27623
     tallies                870
M1   cross sections      215067

     total               243560

  ***********************************************************************
  dump no.  1 on file   runtpe          nps =       0          ctm =   0.00

        4 warning messages so far.
```

bare u(94) sphere ref. la-4208, g. e. hansen and h. c. paxton, 1969, page 4

nps	x	y	z	cell	surf	u	v	w	time	weight	energy
1	0.000e+00	0.000e+00	0.000e+00	1	0	2.724e-01	-9.578e-01	-9.137e-02	0.000e+00	1.000e+00	2.209e+00
2	0.000e+00	0.000e+00	0.000e+00	1	0	3.056e-01	3.129e-01	-8.993e-01	0.000e+00	1.000e+00	4.904e+00
3	0.000e+00	0.000e+00	0.000e+00	1	0	5.717e-01	-7.962e-01	1.982e-01	0.000e+00	1.000e+00	3.809e-01
4	0.000e+00	0.000e+00	0.000e+00	1	0	-2.503e-01	-4.739e-01	-8.442e-01	0.000e+00	1.000e+00	1.331e+00
5	0.000e+00	0.000e+00	0.000e+00	1	0	9.077e-01	-3.611e-01	2.136e-01	0.000e+00	1.000e+00	1.902e+00
6	0.000e+00	0.000e+00	0.000e+00	1	0	6.862e-01	6.811e-01	2.553e-01	0.000e+00	1.000e+00	4.410e-01
7	0.000e+00	0.000e+00	0.000e+00	1	0	4.628e-01	7.977e-01	3.866e-01	0.000e+00	1.000e+00	4.750e-01
8	0.000e+00	0.000e+00	0.000e+00	1	0	-9.478e-01	-1.175e-01	-2.965e-01	0.000e+00	1.000e+00	4.136e+00
9	0.000e+00	0.000e+00	0.000e+00	1	0	-6.401e-01	7.540e-01	-1.479e-01	0.000e+00	1.000e+00	7.453e-02
10	0.000e+00	0.000e+00	0.000e+00	1	0	8.672e-01	-2.841e-01	-4.091e-01	0.000e+00	1.000e+00	3.128e+00
11	0.000e+00	0.000e+00	0.000e+00	1	0	-8.584e-01	2.292e-01	-4.588e-01	0.000e+00	1.000e+00	1.014e+00
12	0.000e+00	0.000e+00	0.000e+00	1	0	1.116e-02	2.035e-01	-9.790e-01	0.000e+00	1.000e+00	1.395e+00
13	0.000e+00	0.000e+00	0.000e+00	1	0	8.237e-01	5.421e-01	-1.661e-01	0.000e+00	1.000e+00	7.748e-01
14	0.000e+00	0.000e+00	0.000e+00	1	0	2.086e-01	-8.823e-01	-4.220e-01	0.000e+00	1.000e+00	1.101e+00
15	0.000e+00	0.000e+00	0.000e+00	1	0	4.248e-01	-9.006e-01	9.153e-02	0.000e+00	1.000e+00	1.951e+00
16	0.000e+00	0.000e+00	0.000e+00	1	0	-7.764e-01	-3.462e-01	-5.266e-01	0.000e+00	1.000e+00	2.186e+00
17	0.000e+00	0.000e+00	0.000e+00	1	0	-3.403e-01	-3.683e-01	8.652e-01	0.000e+00	1.000e+00	1.865e+00
18	0.000e+00	0.000e+00	0.000e+00	1	0	-1.028e-01	3.632e-01	9.260e-01	0.000e+00	1.000e+00	1.229e+00
19	0.000e+00	0.000e+00	0.000e+00	1	0	1.875e-01	-6.654e-01	-7.226e-01	0.000e+00	1.000e+00	1.305e+00
20	0.000e+00	0.000e+00	0.000e+00	1	0	-9.296e-01	3.559e-01	9.599e-02	0.000e+00	1.000e+00	1.000e+00
21	0.000e+00	0.000e+00	0.000e+00	1	0	1.346e-01	-2.447e-01	9.602e-01	0.000e+00	1.000e+00	3.990e+00
22	0.000e+00	0.000e+00	0.000e+00	1	0	-8.666e-01	3.015e-01	-3.977e-01	0.000e+00	1.000e+00	2.665e-01
23	0.000e+00	0.000e+00	0.000e+00	1	0	6.324e-01	8.105e-02	-7.704e-01	0.000e+00	1.000e+00	1.156e+00
24	0.000e+00	0.000e+00	0.000e+00	1	0	7.391e-01	6.198e-01	2.636e-01	0.000e+00	1.000e+00	2.669e+00
25	0.000e+00	0.000e+00	0.000e+00	1	0	-4.460e-01	8.107e-01	-3.794e-01	0.000e+00	1.000e+00	2.185e+00
26	0.000e+00	0.000e+00	0.000e+00	1	0	4.875e-01	8.662e-01	-1.099e-01	0.000e+00	1.000e+00	4.225e+00
27	0.000e+00	0.000e+00	0.000e+00	1	0	-1.177e-01	-8.874e-01	-4.456e-01	0.000e+00	1.000e+00	1.079e+00
28	0.000e+00	0.000e+00	0.000e+00	1	0	9.988e-01	2.035e-04	4.894e-02	0.000e+00	1.000e+00	3.461e+00
29	0.000e+00	0.000e+00	0.000e+00	1	0	-6.569e-01	1.207e-01	-7.443e-01	0.000e+00	1.000e+00	1.836e+00
30	0.000e+00	0.000e+00	0.000e+00	1	0	9.987e-01	-1.801e-02	4.691e-02	0.000e+00	1.000e+00	4.556e-01
31	0.000e+00	0.000e+00	0.000e+00	1	0	-6.446e-02	8.113e-01	5.810e-01	0.000e+00	1.000e+00	6.415e-01
32	0.000e+00	0.000e+00	0.000e+00	1	0	1.705e-01	8.931e-01	-4.163e-01	0.000e+00	1.000e+00	2.764e+00
33	0.000e+00	0.000e+00	0.000e+00	1	0	3.075e-01	9.350e-01	1.767e-01	0.000e+00	1.000e+00	2.785e-01
34	0.000e+00	0.000e+00	0.000e+00	1	0	-7.639e-01	4.272e-01	-4.836e-01	0.000e+00	1.000e+00	9.097e-01
35	0.000e+00	0.000e+00	0.000e+00	1	0	3.773e-01	6.858e-03	9.261e-01	0.000e+00	1.000e+00	3.360e-01
36	0.000e+00	0.000e+00	0.000e+00	1	0	-3.060e-01	-5.581e-01	7.713e-01	0.000e+00	1.000e+00	6.376e-01
37	0.000e+00	0.000e+00	0.000e+00	1	0	-3.715e-01	-2.465e-02	9.281e-01	0.000e+00	1.000e+00	2.186e+00
38	0.000e+00	0.000e+00	0.000e+00	1	0	-6.938e-02	7.584e-01	-6.480e-01	0.000e+00	1.000e+00	7.314e-01
39	0.000e+00	0.000e+00	0.000e+00	1	0	-3.150e-02	1.064e-01	9.938e-01	0.000e+00	1.000e+00	2.997e-01
40	0.000e+00	0.000e+00	0.000e+00	1	0	4.710e-01	-8.627e-01	1.843e-01	0.000e+00	1.000e+00	1.444e+00
41	0.000e+00	0.000e+00	0.000e+00	1	0	-9.073e-01	1.742e-01	3.827e-01	0.000e+00	1.000e+00	1.914e+00
42	0.000e+00	0.000e+00	0.000e+00	1	0	-6.797e-01	-5.789e-01	-4.504e-01	0.000e+00	1.000e+00	1.502e+00
43	0.000e+00	0.000e+00	0.000e+00	1	0	-8.193e-01	-3.227e-01	-4.740e-01	0.000e+00	1.000e+00	5.971e+00
44	0.000e+00	0.000e+00	0.000e+00	1	0	-7.368e-01	-1.367e-01	-6.621e-01	0.000e+00	1.000e+00	1.827e+00
45	0.000e+00	0.000e+00	0.000e+00	1	0	-2.760e-01	-9.374e-01	-2.124e-01	0.000e+00	1.000e+00	1.928e+00
46	0.000e+00	0.000e+00	0.000e+00	1	0	1.343e-01	6.070e-01	-7.833e-01	0.000e+00	1.000e+00	1.351e+00
47	0.000e+00	0.000e+00	0.000e+00	1	0	-4.385e-01	-2.868e-01	-8.518e-01	0.000e+00	1.000e+00	2.288e+00
48	0.000e+00	0.000e+00	0.000e+00	1	0	7.132e-01	-2.260e-01	-6.628e-01	0.000e+00	1.000e+00	1.230e+00
49	0.000e+00	0.000e+00	0.000e+00	1	0	-7.738e-01	2.361e-01	-5.871e-01	0.000e+00	1.000e+00	1.433e+00
50	0.000e+00	0.000e+00	0.000e+00	1	0	-7.912e-01	5.886e-01	1.657e-01	0.000e+00	1.000e+00	6.572e-01

cycle 1 k(collision) 1.360399 removal lifetime(abs) 8.2926e-01 source points generated 4080

cycle 2 k(collision) 1.150124 removal lifetime(abs) 6.7235e-01 source points generated 2575

cycle 3 k(collision) 1.082659 removal lifetime(abs) 6.1977e-01 source points generated 2816

```
      cycle    4  k(collision)  1.024841   removal lifetime(abs)  5.9609e-01   source points generated   2893

      cycle    5  k(collision)  0.986387   removal lifetime(abs)  5.4856e-01   source points generated   2879

.113  cycle    6  k(collision)  1.003523   removal lifetime(abs)  5.5542e-01   source points generated   3013

       estimator       cycle    7      ave of     2 cycles        combination             simple average      combined average        corr
      k(collision)        1.013401          1.008462 0.0049       k(col/abs)            0.000000 0.0000      0.000000 0.0000      0.0000
      k(absorption)       1.010081          1.004574 0.0055       k(abs/tk ln)          0.000000 0.0000      0.000000 0.0000      0.0000
      k(trk length)       0.993774          0.999710 0.0059       k(tk ln/col)          0.000000 0.0000      0.000000 0.0000      0.0000
      rem life(col)   5.6368e-01       5.8025e-01 0.0061
      rem life(abs)   5.6324e-01       5.5933e-01 0.0070        life(col/abs)      0.0000e+00 0.0000    0.0000e+00 0.0000      0.0000
      source points generated   3093

       estimator       cycle    8      ave of     3 cycles        combination             simple average      combined average        corr
      k(collision)        0.988762          1.001229 0.0078       k(col/abs)            0.999855 0.0073      0.986952 0.0093      0.9944
      k(absorption)       0.986295          0.998481 0.0069       k(abs/tk ln)          1.001201 0.0009      1.001523 0.0009     -0.9796
      k(trk length)       1.012340          1.003920 0.0054       k(tk ln/col)          1.002575 0.0015      1.002819 0.0014     -0.9530
      rem life(col)   5.5512e-01       5.5854e-01 0.0047
      rem life(abs)   5.5518e-01       5.5795e-01 0.0047        life(col/abs)      5.5824e-01 0.0047    5.5850e-01 0.0076      0.9870
      source points generated   2853

      source distribution written to file srctp          cycle =    8

      source distribution written to file srctp          cycle =    8

       estimator       cycle    9      ave of     4 cycles        combination             simple average      combined average        corr
      k(collision)        1.009213          1.003225 0.0058       k(col/abs)            1.002314 0.0057      1.001116 0.0075      0.9769
      k(absorption)       1.010170          1.001403 0.0057       k(abs/tk ln)          1.003664 0.0025      1.004064 0.0030     -0.5056
      k(trk length)       1.011939          1.005925 0.0043       k(tk ln/col)          1.004575 0.0023      1.004820 0.0026     -0.6360
      rem life(col)   5.8307e-01       5.6467e-01 0.0113       k(col/abs/tk ln)       1.003518 0.0032      1.007208 0.0042
      rem life(abs)   5.8319e-01       5.6426e-01 0.0117        life(col/abs)      5.6446e-01 0.0115    5.6634e-01 0.0128      0.9989
      source points generated   3105

       estimator       cycle   10      ave of     5 cycles        combination             simple average      combined average        corr
      k(collision)        1.018903          1.006361 0.0055       k(col/abs)            1.005764 0.0056      1.007153 0.0064      0.9803
      k(absorption)       1.020219          1.005167 0.0058       k(abs/tk ln)          1.006285 0.0033      1.006727 0.0034     -0.0890
      k(trk length)       1.013316          1.007403 0.0036       k(tk ln/col)          1.006882 0.0029      1.007046 0.0031     -0.2491
      rem life(col)   5.7970e-01       5.6768e-01 0.0102       k(col/abs/tk ln)       1.006310 0.0037      1.009419 0.0026
      rem life(abs)   5.7802e-01       5.6701e-01 0.0102        life(col/abs)      5.6734e-01 0.0102    5.6711e-01 0.0127      0.9980
      source points generated   3009

       estimator       cycle   11      ave of     6 cycles        combination             simple average      combined average        corr
      k(collision)        0.990512          1.003719 0.0052       k(col/abs)            1.003028 0.0053      1.005343 0.0059      0.9854
      k(absorption)       0.988192          1.002337 0.0055       k(abs/tk ln)          1.003254 0.0040      1.003548 0.0044      0.3252
      k(trk length)       0.988005          1.004170 0.0044       k(tk ln/col)          1.003945 0.0038      1.003993 0.0041      0.2256
      rem life(col)   5.6263e-01       5.6684e-01 0.0085       k(col/abs/tk ln)       1.003409 0.0042      1.007403 0.0046
      rem life(abs)   5.6143e-01       5.6608e-01 0.0085        life(col/abs)      5.6646e-01 0.0085    5.6681e-01 0.0104      0.9979
      source points generated   2866

       estimator       cycle   12      ave of     7 cycles        combination             simple average      combined average        corr
      k(collision)        0.982074          1.000627 0.0054       k(col/abs)            1.000036 0.0054      1.000895 0.0060      0.9883
      k(absorption)       0.982087          0.999445 0.0055       k(abs/tk ln)          1.000770 0.0042      1.001400 0.0045      0.4983
      k(trk length)       0.989648          1.002095 0.0043       k(tk ln/col)          1.001361 0.0041      1.001656 0.0043      0.4409
      rem life(col)   5.6152e-01       5.6608e-01 0.0073       k(col/abs/tk ln)       1.000722 0.0045      1.003873 0.0054
      rem life(abs)   5.6113e-01       5.6537e-01 0.0073        life(col/abs)      5.6572e-01 0.0073    5.6568e-01 0.0086      0.9979
      source points generated   3058

       estimator       cycle   13      ave of     8 cycles        combination             simple average      combined average        corr
      k(collision)        0.989381          0.999221 0.0049       k(col/abs)            0.998638 0.0049      0.999433 0.0054      0.9892
      k(absorption)       0.988334          0.998056 0.0050       k(abs/tk ln)          1.000336 0.0037      1.001413 0.0038      0.4342
          .
          .
          .
          .
```

```
 estimator         cycle   24      ave of    19 cycles        combination             simple average     combined average       corr
k(collision)         1.006033          0.996484 0.0039       k(col/abs)              0.996282 0.0039     0.995577 0.0039     0.9932
k(absorption)        1.008681          0.996079 0.0038       k(abs/tk ln)            0.997703 0.0033     0.998218 0.0034     0.6805
k(trk length)        1.003506          0.999326 0.0034       k(tk ln/col)            0.997905 0.0034     0.998482 0.0034     0.6932
rem life(col)      5.6734e-01        5.6526e-01 0.0051       k(col/abs/tk ln)        0.997296 0.0035     0.997627 0.0036
rem life(abs)      5.8721e-01        5.6499e-01 0.0050       life(col/abs)         5.6514e-01 0.0051   5.6346e-01 0.0049     0.9986
source points generated    3084

 estimator         cycle   25      ave of    20 cycles        combination             simple average     combined average       corr
k(collision)         1.007748          0.997047 0.0036       k(col/abs)              0.996923 0.0037     0.996659 0.0038     0.9923
k(absorption)        1.010468          0.996800 0.0037       k(abs/tk ln)            0.997888 0.0032     0.998237 0.0032     0.6426
k(trk length)        0.992324          0.996976 0.0033       k(tk ln/col)            0.998011 0.0032     0.998387 0.0032     0.6653
rem life(col)      5.6062e-01        5.6505e-01 0.0049       k(col/abs/tk ln)        0.997608 0.0033     0.997970 0.0033
rem life(abs)      5.6145e-01        5.6482e-01 0.0048       life(col/abs)         5.6493e-01 0.0048   5.6364e-01 0.0047     0.9984
source points generated    3010

 estimator         cycle   26      ave of    21 cycles        combination             simple average     combined average       corr
k(collision)         0.995877          0.996991 0.0036       k(col/abs)              0.996838 0.0035     0.996523 0.0036     0.9922
k(absorption)        0.994386          0.996685 0.0035       k(abs/tk ln)            0.997630 0.0030     0.997914 0.0031     0.6412
k(trk length)        0.990574          0.998576 0.0031       k(tk ln/col)            0.997784 0.0031     0.998071 0.0031     0.6618
rem life(col)      5.5266e-01        5.6446e-01 0.0047       k(col/abs/tk ln)        0.997417 0.0031     0.997571 0.0031
rem life(abs)      5.5169e-01        5.6419e-01 0.0047       life(col/abs)         5.6432e-01 0.0047   5.6313e-01 0.0047     0.9983
source points generated    2877

source distribution written to file srctp          cycle =  26

 estimator         cycle   27      ave of    22 cycles        combination             simple average     combined average       corr
k(collision)         1.007764          0.997482 0.0034       k(col/abs)              0.997286 0.0034     0.996820 0.0034     0.9921
k(absorption)        1.005612          0.997091 0.0034       k(abs/tk ln)            0.998348 0.0030     0.998559 0.0030     0.6412
k(trk length)        1.021202          0.999604 0.0032       k(tk ln/col)            0.998543 0.0030     0.998795 0.0031     0.6659
rem life(col)      5.7691e-01        5.6502e-01 0.0046       k(col/abs/tk ln)        0.998059 0.0031     0.998068 0.0031
rem life(abs)      5.7637e-01        5.6475e-01 0.0046       life(col/abs)         5.6488e-01 0.0046   5.6356e-01 0.0045     0.9984
source points generated    3070

 estimator         cycle   28      ave of    23 cycles        combination             simple average     combined average       corr
k(collision)         1.010631          0.998053 0.0033       k(col/abs)              0.997844 0.0033     0.997312 0.0033     0.9923
k(absorption)        1.009619          0.997635 0.0033       k(abs/tk ln)            0.998665 0.0028     0.998874 0.0029     0.6368
k(trk length)        1.001663          0.999694 0.0030       k(tk ln/col)            0.998874 0.0029     0.999099 0.0029     0.6608
rem life(col)      5.5773e-01        5.6471e-01 0.0045       k(col/abs/tk ln)        0.998461 0.0030     0.998284 0.0030
rem life(abs)      5.5612e-01        5.6446e-01 0.0044       life(col/abs)         5.6458e-01 0.0044   5.6335e-01 0.0043     0.9984
source points generated    2999

 estimator         cycle   29      ave of    24 cycles        combination             simple average     combined average       corr
k(collision)         1.032907          0.999506 0.0035       k(col/abs)              0.999299 0.0035     0.998795 0.0035     0.9936
k(absorption)        1.032621          0.999093 0.0034       k(abs/tk ln)            0.999449 0.0028     0.999595 0.0028     0.5928
k(trk length)        1.002348          0.999804 0.0029       k(tk ln/col)            0.999655 0.0029     0.999726 0.0029     0.6167
rem life(col)      5.5393e-01        5.6426e-01 0.0043       k(col/abs/tk ln)        0.999468 0.0030     0.998887 0.0029
rem life(abs)      5.5386e-01        5.6402e-01 0.0043       life(col/abs)         5.6414e-01 0.0043   5.6290e-01 0.0042     0.9984
source points generated    3125

M4
 estimator         cycle   30      ave of    25 cycles        combination             simple average     combined average       corr
k(collision)         1.006439          0.999783 0.0034       k(col/abs)              0.999614 0.0033     0.999268 0.0034     0.9934
k(absorption)        1.007666          0.999445 0.0033       k(abs/tk ln)            0.999874 0.0028     1.000043 0.0028     0.5989
k(trk length)        1.012295          1.000304 0.0028       k(tk ln/col)            1.000044 0.0028     1.000160 0.0028     0.6195
rem life(col)      5.7720e-01        5.6477e-01 0.0043       k(col/abs/tk ln)        0.999844 0.0029     0.999575 0.0028
rem life(abs)      5.7778e-01        5.6457e-01 0.0042       life(col/abs)         5.6467e-01 0.0042   5.6385e-01 0.0042     0.9984
source points generated    3004

source distribution written to file srctp          cycle =  30
```

115 run terminated when 30 kcode cycles were done. 2 05/14/86 12:22:29
bare u(94) sphere ref. la-4208, g. e. hansen and h. c. paxton, 1969, page 4 probid = 2 05/14/86 12:13:47

ledger of net neutron creation and loss (for accounting only)

	tracks	weight (per source particle)	energy (per source particle)		tracks	weight (per source particle)	energy (per source particle)
source	90661	9.9271e-01	2.0343e+00	escape	77758	5.6324e-01	9.4810e-01
scattering	0	0.	0.	scattering	0	0.	4.4389e-01
fission	0	0.	0.	capture	0	4.2770e-02	3.1293e-02
(n,xn)	275	1.8597e-03	1.1704e-03	energy cutoff	0	0.	0.
forced collision	0	0.	0.	time cutoff	0	0.	0.
weight cutoff	0	3.3269e-02	1.7790e-02	weight cutoff	13178	3.3467e-02	1.7736e-02
weight window	0	0.	0.	weight window	0	0.	0.
cell importance	0	0.	0.	cell importance	0	0.	0.
energy import.	0	0.	0.	energy import.	0	0.	0.
dxtran	0	0.	0.	dxtran	0	0.	0.
exp. transform	0	0.	0.	exp. transform	0	0.	0.
				dead fission	0	3.8836e-01	6.1223e-01
total	90936	1.0278e+00	2.0532e+00	total	90936	1.0278e+00	2.0532e+00

neutron tracks per source particle	1.0030e+00
neutron collisions per source particle	3.9628e+00
total neutron collisions	359272
net multiplication	1.0019e+00 0.0003

average lifetime, shakes		cutoffs	
escape	5.8556e-01	tco	1.0000+120
capture	5.7356e+00	eco	0.0000e+00
capture or escape	9.4903e-01	wc1	-5.0000e-01
any termination	1.0055e+00	wc2	-2.5000e-01

computer time so far in this run	6.61 minutes
computer time in mcrun (4c0)	6.56 minutes
source particles per minute	1.3819e+04
field length	433132
random numbers generated	4335729
last starting random number	144868273130233
next starting random number	56706737829237

total neutrons banked	275
per source particle	3.0333e-03
total photons banked	0
per source particle	0.0000e+00
maximum number ever in bank	1
bank overflows to backup file	0

range of sampled source weights = 7.3529e-01 to 1.1650e+00

-398-

116 neutron weight balance in each cell -- physical events print table 130

cell progr	probl	fission loss	fission neutrons	(n,xn)	capture	total
1	1	0.0000e+00	0.0000e+00	1.8597e-03	-4.3113e-01	-4.2927e-01
total		0.0000e+00	0.0000e+00	1.8597e-03	-4.3113e-01	-4.2927e-01

```
V17  tally   1          nps =     90661
               neutron tally type 1:   number of particles crossing a surface.

     surface  1
          energy
       1.0000e-07    0.00000e+00 0.0000
       4.0000e-07    0.00000e+00 0.0000
       1.0000e-06    0.00000e+00 0.0000
       3.0000e-06    0.00000e+00 0.0000
       1.0000e-05    0.00000e+00 0.0000
       3.0000e-05    0.00000e+00 0.0000
       1.0000e-04    0.00000e+00 0.0000
       5.5000e-04    0.00000e+00 0.0000
       3.0000e-03    1.34138e-05 1.0000
       1.7000e-02    2.53367e-04 0.1783
       1.0000e-01    8.95315e-03 0.0291
       4.0000e-01    8.14137e-02 0.0094
       9.0000e-01    1.34197e-01 0.0075
       1.4000e+00    9.64488e-02 0.0096
       3.0000e+00    1.61164e-01 0.0073
       2.0000e+01    9.38912e-02 0.0103
          total      5.76334e-01 0.0020

     for the tally-fluctuation-chart bin:
     average tally per history = 5.76334e-01        largest tally = 1.68652e+00
     largest/average = 2.92629e+00                  nps of largest =    34761
```

```
1tally        6         nps =     90661
                 neutron tally type 6:   track length estimate of heating.            units   mev/gram

                 masses
                         cell:         1
                               5.19312e+04

   cell  1
       energy
     1.0000e-07    0.00000e+00 0.0000
     4.0000e-07    0.00000e+00 0.0000
     1.0000e-06    0.00000e+00 0.0000
     3.0000e-06    0.00000e+00 0.0000
     1.0000e-05    0.00000e+00 0.0000
     3.0000e-05    0.00000e+00 0.0000
     1.0000e-04    0.00000e+00 0.0000
     5.5000e-04    2.93843e-06 1.0000
     3.0000e-03    1.22176e-07 0.4268
     1.7000e-02    2.40289e-06 0.1095
     1.0000e-01    4.07238e-05 0.0235
     4.0000e-01    2.17961e-04 0.0087
     9.0000e-01    2.79079e-04 0.0068
     1.4000e+00    1.94792e-04 0.0082
     3.0000e+00    3.40347e-04 0.0062
     2.0000e+01    1.81714e-04 0.0087
       total       1.25717e-03 0.0021

   for the tally-fluctuation-chart bin:
   average tally per history = 6.52863e+01        largest tally = 4.22391e+02
   largest/average = 6.46982e+00                  nps of largest =     31329
```

```
1l0  tally   7         nps =     90661
                neutron tally type 7:   track length estimate of fission heating.    units   mev/gram

                masses
                        cell:        1
                               5.19312e+04

      cell  1
           energy
         1.0000e-07   0.00000e+00 0.0000
         4.0000e-07   0.00000e+00 0.0000
         1.0000e-06   0.00000e+00 0.0000
         3.0000e-06   0.00000e+00 0.0000
         1.0000e-05   0.00000e+00 0.0000
         3.0000e-05   0.00000e+00 0.0000
         1.0000e-04   0.00000e+00 0.0000
         5.5000e-04   3.13824e-08 1.0000
         3.0000e-03   1.30215e-07 0.4268
         1.7000e-02   2.56054e-06 0.1095
         1.0000e-01   4.33805e-05 0.0235
         4.0000e-01   2.31971e-04 0.0087
         9.0000e-01   2.96497e-04 0.0068
         1.4000e+00   2.07312e-04 0.0082
         3.0000e+00   3.62280e-04 0.0062
         2.0000e+01   1.91675e-04 0.0087
           total      1.33584e-03 0.0021

      for the tally-fluctuation-chart bin:
      average tally per history = 6.93716e+01        largest tally = 4.49822e+02
      largest/average = 6.48424e+00                  nps of largest =     31329
```

```
1tally 14        nps =    90661    total neutron fluence, total fissions, total fission neutrons,
                                     total neutron absorptions, and neutron heating (mev/gram)
           neutron tally type 4:   track length estimate of particle flux.
           multiplier bin  1:    2.76185e+03
           multiplier bin  2:   -2.76185e+03   10         -6
           multiplier bin  3:   -2.76185e+03   10         -6     -7
           multiplier bin  4:   -2.76185e+03   10         -2
           multiplier bin  5:   -5.31830e-02   10          1     -4

           volumes
                  cell:        1
                       2.76185e+03

 cell  1
```

mult bin:	1		2		3		4		5	
energy										
1.0000e-07	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000
4.0000e-07	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.00000
1.0000e-06	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000
3.0000e-06	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000
1.0000e-05	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000
3.0000e-05	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000
1.0000e-04	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000
5.5000e-04	1.25919e-05	1.0000	9.00999e-06	1.0000	2.19546e-05	1.0000	3.81699e-06	1.0000	2.93844e-08	1.0000
3.0000e-03	1.38100e-04	0.4457	3.73851e-05	0.4268	9.10963e-05	0.4268	1.54396e-05	0.4258	1.22177e-07	0.4266
1.7000e-02	5.73427e-03	0.1121	7.35139e-04	0.1095	1.79131e-03	0.1095	3.09238e-04	0.1089	2.40289e-06	0.1095
1.0000e-01	1.57487e-01	0.0234	1.24547e-02	0.0235	3.03951e-02	0.0234	4.06022e-03	0.0238	4.07239e-05	0.0235
4.0000e-01	1.10962e+00	0.0087	6.88000e-02	0.0087	1.63742e-01	0.0087	1.43925e-02	0.0089	2.17961e-04	0.0087
9.0000e-01	1.61562e+00	0.0088	8.51305e-02	0.0068	2.12390e-01	0.0068	1.20409e-02	0.0069	2.79079e-04	0.0068
1.4000e+00	1.06800e+00	0.0082	5.95247e-02	0.0082	1.51449e-01	0.0082	5.37765e-03	0.0083	1.94792e-04	0.0082
3.0000e+00	1.75174e+00	0.0062	1.04017e-01	0.0062	2.75078e-01	0.0062	4.64010e-03	0.0064	3.40348e-04	0.0062
2.0000e+01	9.84369e-01	0.0066	5.50329e-02	0.0087	1.65346e-01	0.0088	1.11483e-03	0.0091	1.81714e-04	0.0087
total	6.69271e+00	0.0021	3.83541e-01	0.0021	1.00030e+00	0.0021	4.19548e-02	0.0040	1.25717e-03	0.0021

```
 for the tally-fluctuation-chart bin:
 average tally per history = 1.84843e+04        largest tally = 9.12404e+04
 largest/average = 4.93611e+00                  nps of largest =     71062
```

```
1 tally 24        nps =    90861    total neutron fluence, total fissions, total fission neutrons,
                                     total neutron absorptions, and neutron heating (mev/gram)
          neutron tally type 4:    track length estimate of particle flux.
          multiplier bin  1:   2.76185e+03
          multiplier bin  2:  -2.76185e+03   10        -6
          multiplier bin  3:  -2.76185e+03   10        -6     -7
          multiplier bin  4:  -2.76185e+03   10        -2
          multiplier bin  5:  -5.31830e-02   10         1     -4

          volumes
                cell:        1
                    2.76185e+03

cell  1
```

mult bin:	1		2		3		4		5	
energy										
1.6700e-04	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000
4.5400e-04	1.25919e-05	1.0000	9.00999e-06	1.0000	2.19546e-05	1.0000	3.81699e-06	1.0000	2.93844e-08	1.0000
1.2350e-03	3.00937e-05	0.7094	1.12100e-05	0.7075	2.73153e-05	0.7075	6.32326e-06	0.7151	3.66226e-08	0.7075
3.3500e-03	1.17717e-04	0.4934	2.83214e-05	0.4913	6.90108e-05	0.4913	9.86009e-06	0.4865	9.25692e-08	0.4913
9.1200e-03	1.29683e-03	0.1757	2.09858e-04	0.1762	5.11361e-04	0.1762	8.75510e-05	0.1738	6.85888e-07	0.1762
2.4800e-02	1.03553e-02	0.0883	1.11687e-03	0.0890	2.72146e-03	0.0890	4.68805e-04	0.0890	3.65100e-08	0.0890
6.7600e-02	7.07383e-02	0.0344	5.91718e-03	0.0344	1.44314e-02	0.0344	2.03849e-03	0.0345	1.93466e-05	0.0344
1.8400e-01	3.48278e-01	0.0157	2.37844e-02	0.0157	5.81934e-02	0.0157	6.35992e-03	0.0159	7.77915e-05	0.0157
3.0300e-01	4.59137e-01	0.0133	2.73436e-02	0.0133	6.72066e-02	0.0133	5.93346e-03	0.0134	8.94839e-05	0.0133
5.0000e-01	7.65307e-01	0.0103	4.19966e-02	0.0103	1.03855e-01	0.0103	7.25708e-03	0.0103	1.37546e-04	0.0103
8.2300e-01	1.02729e+00	0.0088	5.36727e-02	0.0088	1.34019e-01	0.0088	7.39018e-03	0.0088	1.75964e-04	0.0088
1.3530e+00	1.18678e+00	0.0077	6.54314e-02	0.0077	1.66068e-01	0.0078	6.29446e-03	0.0078	2.14259e-04	0.0077
1.7380e+00	6.14241e-01	0.0109	3.58257e-02	0.0109	9.25415e-02	0.0109	2.16587e-03	0.0110	1.17065e-04	0.0109
2.2320e+00	5.99552e-01	0.0111	3.62716e-02	0.0111	9.53038e-02	0.0111	1.59697e-03	0.0111	1.18719e-04	0.0111
2.8650e+00	5.36388e-01	0.0118	3.17828e-02	0.0118	8.58987e-02	0.0118	1.07722e-03	0.0118	1.04062e-04	0.0118
3.6800e+00	4.50240e-01	0.0132	2.52981e-02	0.0132	7.09219e-02	0.0132	6.82780e-04	0.0132	8.29939e-05	0.0132
6.0700e+00	5.05185e-01	0.0124	2.62612e-02	0.0124	7.85905e-02	0.0124	5.18410e-04	0.0126	8.68031e-05	0.0124
7.7900e+00	8.14284e-02	0.0320	5.53919e-03	0.0322	1.86934e-02	0.0323	5.15037e-05	0.0322	1.84761e-05	0.0322
1.0000e+01	3.00001e-02	0.0532	2.51175e-03	0.0532	9.12063e-03	0.0532	1.09229e-05	0.0541	8.39499e-06	0.0532
1.2000e+01	5.50287e-03	0.1230	4.51308e-04	0.1230	1.77853e-03	0.1232	1.02435e-06	0.1234	1.51170e-06	0.1230
1.3500e+01	4.85174e-04	0.4561	4.27358e-05	0.4557	1.80516e-04	0.4555	6.30899e-08	0.4571	1.44490e-07	0.4557
1.5000e+01	3.44493e-04	0.4480	3.37573e-05	0.4488	1.50074e-04	0.4497	3.74925e-08	0.4448	1.15070e-07	0.4489
2.0000e+01	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000	0.00000e+00	0.0000
total	6.69271e+00	0.0021	3.83541e-01	0.0021	1.00030e+00	0.0021	4.19548e-02	0.0040	1.25717e-03	0.0021

```
for the tally-fluctuation-chart bin:
average tally per history = 1.84843e+04        largest tally = 9.12404e+04
largest/average = 4.93611e+00                  nps of largest =     71062
```

122 tally fluctuation charts

	tally 1			tally 2			tally 4		
nps	mean	error	fom	mean	error	fom	mean	error	fom
8000	0.00000e+00	0.0000	0.0e+00	0.00000e+00	0.0000	0.0e+00	0.00000e+00	0.0000	0.0e+00
16000	5.74149e-01	0.0237	33409	9.12573e-04	0.0347	15606	2.48204e-03	0.0227	36515
24000	5.77842e-01	0.0059	40250	9.33421e-04	0.0098	14643	2.43507e-03	0.0062	36873
32000	5.75569e-01	0.0043	39205	9.27643e-04	0.0073	13536	2.44032e-03	0.0045	36162
40000	5.77560e-01	0.0035	39633	9.29710e-04	0.0059	13816	2.42967e-03	0.0037	35972
48000	5.78259e-01	0.0030	41060	9.31699e-04	0.0051	14106	2.42371e-03	0.0032	36733
56000	5.79228e-01	0.0027	41013	9.37521e-04	0.0047	13393	2.41453e-03	0.0029	36413
64000	5.78157e-01	0.0025	40912	9.33904e-04	0.0043	13497	2.41965e-03	0.0026	36673
72000	5.78036e-01	0.0023	40353	9.32162e-04	0.0040	13809	2.42145e-03	0.0024	36307
80000	5.77746e-01	0.0022	40318	9.31084e-04	0.0037	13681	2.42085e-03	0.0023	36284
88000	5.78365e-01	0.0020	39959	9.28829e-04	0.0035	13689	2.42279e-03	0.0021	36431
90661	5.78334e-01	0.0020	40117	9.28875e-04	0.0034	13641	2.42327e-03	0.0021	36554

	tally 6			tally 7			tally 14		
nps	mean	error	fom	mean	error	fom	mean	error	fom
8000	0.00000e+00	0.0000	0.0e+00	0.00000e+00	0.0000	0.0e+00	0.00000e+00	0.0000	0.0e+00
16000	1.28219e-03	0.0228	36216	1.36233e-03	0.0228	36173	6.85502e+00	0.0227	36515
24000	1.28028e-03	0.0062	36079	1.33912e-03	0.0062	38058	6.72529e+00	0.0062	36873
32000	1.26509e-03	0.0045	35167	1.34424e-03	0.0045	35138	6.73981e+00	0.0045	36162
40000	1.25990e-03	0.0037	35100	1.33871e-03	0.0037	35070	6.71040e+00	0.0037	35972
48000	1.25677e-03	0.0032	35903	1.33539e-03	0.0032	35873	6.69393e+00	0.0032	36733
56000	1.25194e-03	0.0029	35574	1.33029e-03	0.0029	35543	6.66857e+00	0.0029	36413
64000	1.25472e-03	0.0026	35810	1.33325e-03	0.0026	35779	6.68271e+00	0.0026	36673
72000	1.25593e-03	0.0025	35397	1.33452e-03	0.0025	35367	6.68769e+00	0.0024	36307
60000	1.25553e-03	0.0023	35342	1.33410e-03	0.0023	35313	6.68602e+00	0.0023	36284
88000	1.25684e-03	0.0022	35491	1.33549e-03	0.0022	35462	6.69138e+00	0.0021	36431
90661	1.25717e-03	0.0021	35601	1.33584e-03	0.0021	35571	6.69271e+00	0.0021	36554

	tally 24		
nps	mean	error	fom
8000	0.00000e+00	0.0000	0.0e+00
16000	6.85502e+00	0.0227	36515
24000	6.72529e+00	0.0062	36873
32000	6.73981e+00	0.0045	36162
40000	6.71040e+00	0.0037	35972
48000	6.69393e+00	0.0032	36733
56000	6.66857e+00	0.0029	36413
64000	6.68271e+00	0.0026	36673
72000	6.68769e+00	0.0024	36307
80000	6.66602e+00	0.0023	36264
88000	6.69138e+00	0.0021	36431
90661	6.69271e+00	0.0021	36554

••

dump no. 2 on file runtpe nps = 90661 ctm = 6.56

4 warning messages so far.

run terminated when 30 kcode cycles were done.

mcnp version 3a 04/30/86 9406cc14n2 2 05/14/86 12:22:30 probid = 2 05/14/86 12:13:47

Notes:

$\mathcal{N}$1: This representation of a one-dimensional Godiva was suggested by the Nuclear Criticality Safety Group HSE-6 and is from LA-4208.

$\mathcal{N}$1: The KCODE card indicates this is a criticality calculation with a nominal source size of 3000 particles, an estimate of k_{eff} of 1.0, skip 5 cycles before averaging k_{eff} or tallying, and run a total of 30 cycles if computer time permits. A tally batch size of 25 is large enough to ensure that the standard normal distribution confidence interval statements at the 1σ and 2σ levels should apply. A total of 3000 particles was selected to run the problem in less than 10 minutes. Tally normalization will be by the starting source weight by default.

$\mathcal{N}$3: This warning is a reminder. The tallies must be scaled by the steady state power level of the critical system in units of fission neutrons per unit time. For example, if Godiva is operating at a power level of 100 watts, the tally scaling factor would be (3.45×10^{12} fission/s) (2.61 neutrons/fission) = 9.0×10^{12} neutrons/s. The tallies will then have the same time units. Tallies for subcritical systems do not include any multiplication effects because fission is treated as an absorption. Tallies can be estimated for subcritical systems by multiplying the results by the system multiplication $1/(1-k_{eff})$. See Chapter 2 Sec. VIII for further discussion of this topic.

$\mathcal{N}$4: One source location at the center of the 94% enriched uranium sphere is used to begin the first cycle. If a suitable SRCTP file exists, the KSRC card should be removed.

The sources for each generation are the fission locations and neutron energies from fission found in the previous generation. Therefore, in a k_{eff} calculation the fission distribution converges to a stable distribution as a function of space. For complicated problem geometries, the fission distribution must converge for the calculated k_{eff} to converge. This effect is minimized by sampling a larger number of particles per generation. Usually the first generation source is not too important because subsequent later sources will have converged. If the user source selects good source points on the KSRC

card, the problem will converge to a stable k_{eff} in fewer generations. It is critical that the source points have converged before k_{eff}s and tallies are calculated to ensure proper mean k_{eff}s and confidence intervals.

$\mathcal{N}$5: This shows the use of the FM card to calculate the quantities described by the FC14 comment card. The volume of the sphere is 2761.85 cc and is used as a multiplier to obtain total tallies. The negative sign in front of the multiplier causes the atom density to be included in the calculation. The number 0.053183 is the reciprocal of the uranium density of 18.8030 g/cc and converts the flux tally to MeV/g.

$\mathcal{N}$6: The Hansen–Roach energy structure is used as the default for all tallies except tally 24. The energy structure of tally 24 is one commonly used at Los Alamos for a wide variety of calculations.

$\mathcal{N}$7: This table gives detailed information about the criticality source from the KSRC card, including points accepted and rejected. Entries from the KCODE card are listed also. The table shows that total (as opposed to prompt) fission $\bar{\nu}$ data are being used by default to account for the effect of delayed neutrons. Delayed neutron generation is combined with prompt neutron generation and the prompt fission neutron energy spectrum is used. Delayed neutrons typically have a softer spectrum than prompt neutrons; therefore, MCNP will predict a slightly harder fission energy spectrum. This is usually a small effect and is not significant. An SRCTP file has been generated for use in future versions of the problem.

$\mathcal{N}$8: The sum of the material fractions from the M10 card is 0.048146, the same as the atom density in cell 1.

$\mathcal{N}$9: The density and volume were used in determining the multipliers for the FM card.

$\mathcal{N}$10: The cross-section tables show that ^{235}U and ^{238}U use the total $\bar{\nu}$ and ^{234}U may use prompt or total. Checking XSLIST or Table G.2 shows this evaluation uses the total value for $\bar{\nu}$. The warning message is explained in N20 in the TEST1 problem output.

$\mathcal{N}$11: If cross–section space required is too large, thinned or discrete reaction cross–section sets can be used for isotopes with small atom fractions (see Print Table 40).

$\mathcal{N}$12: All source points are at the origin as specified on the KSRC card. The directions are isotropic and the energy is sampled from a fission spectrum.

$\mathcal{N}$13: Five cycles are skipped before averaging of k_{eff} and tallying start. Cycle 7 begins the average tally results printout. Cycle 8 contains the first results of the simple and combined averages, which require a minimum of three values for each estimator of k_{eff}.

$\mathcal{N}$14: There are three k_{eff} estimators, and they use the collision, absorption. and track length methods discussed in Chapter 2.VIII.B. All combinations of these estimators are included. The positive correlations of the various k_{eff} estimators result in almost no reduction in the relative errors for the combined estimators. The estimator with the smallest relative error is generally selected. After 30 total cycles and 25 averaging cycles, all of the k_{eff} values agree well at unity and have an estimated relative error at the 1σ level of 0.0028. File SRCTP contains the 3004 source points that were generated during cycle 30.

$\mathcal{N}$15: The ledger table gives the results of the problem and includes the five cycles that were not used for averaging k_{eff} or tallying. The gain side on the left of the table shows that the starting source weight is slightly less than unity, which will increase the tallies slightly because of the normalization by weight. The loss side of the table gives general guidelines about what happened in the problem. The values will not agree exactly with separate tallies in the problem because the point source used in the first cycle required several cycles to approach the correct spatial distribution of fission neutron sources. The dead fission category is for the weight lost to fission, which is treated as a terminal event for the criticality calculation. Parasitic capture is listed separately. No tracks are lost to either the capture or dead fission categories because implicit capture is being used (the default for EMCNF with no PHYS:N card present is 0).

$\mathcal{N}$16: The neutron weight balance in each cell by physical events shows that capture includes both the capture and dead fission contributions from the ledger table.

$\mathcal{N}$17: The total leakage in tally 1 is higher than the escapes in the ledger table for two reasons. For tallying purposes the first five cycles were skipped. These five cycles had a smaller leakage because the point source in cycle one was in the center of the sphere, making leakage less likely. These smaller leakage numbers in cycles one through five are included in the ledger table escapes. Secondly, a small increase in tally 1 (0.7%) is caused by the weight normalization to 0.99271.

$\mathcal{N}$18: The heating in the uranium sphere does not include any estimate from photons. To account for photons, a coupled neutron/photon criticality problem must be run using a MODE N P card. An F7 fission heating tally may give a good approximation, see note N19.

$\mathcal{N}$19: The fission heating estimate assumes that all photons are deposited locally. The difference between the F6 and F7 tally is discussed on page 85. Because Godiva is an optically thick system to photons, the F7 tally should be a good approximation to the total heating. A MODE N P calculation of this problem produced a neutron heating (F6) of 1.252×10^{-3} (0.0021) MeV/g and a photon heating of 6.502×10^{-5} (0.0042), which adds to about the estimate of the F7 tally, 1.336×10^{-3} (the estimated relative errors are listed in parentheses). If the 100 watt power level normalization in note 3 is used to scale tally 7, (1.336×10^{-3} MeV/g) (51931 g) (9.0×10^{12} neutrons/s) (1.602×10^{-13} W/MeV/s) = 100.03 watts. Thus, the source normalization and tally are consistent with the 100 watt assumed power level.

$\mathcal{N}$20: Tally 14 includes several quantities based on the track length estimator. The first five cycles were not included in these estimates. Column 1 is the total volume integrated fluence, which is small below 0.1 MeV because there is no moderator. Column 2 is the total fissions. This is lower than the dead fission entry in the ledger because of the reduced leaking and more interactions of the point source in cycle 1. Note that the column 2 result is obtained by a track length estimator and the ledger table dead fissions result is

from a collision estimator. Column 3 is the total fission neutrons produced, which agrees exactly with the track length k_{eff} estimator. The estimated relative error is different for column 3 because the error estimation procedures are handled differently. Dividing the total of column 3 by the total of column 2 gives an average value of $\bar{\nu}$ of 2.608 neutrons per fission. Column 4 gives the track length estimation of absorptions, which is slightly less than the collision estimator for captures in the ledger table as expected. The neutron heating in MeV/g is the same as the F6 tally.

$\mathcal{N}$21: Tally 24 is the same as tally 14 except for the energy group structure.

$\mathcal{N}$22: The tally fulctuation charts confirm stable, efficient tallies in the bins monitored. The charts confirm that the first five cycles (15243 histories) were skipped because of the zeros after 8000 particles were run and the large reduction in the estimated relative error between 16000 and 24000 histories.

A few final points should be made about KCODE calculations. To perform a KCODE calculation with the source points file SRCTP produced by the above initial run, the KSRC card should be removed from the input file. To perform a continue-run, the standard MCNP rules apply. Having an input file beginning with CONTINUE is optional. In addition, the SRCTP file is not required for a KCODE continue-run because the source points information is contained on the RUNTPE file.

V. EVENT LOG AND GEOMETRY ERRORS

MCNP cannot detect a geometry error while processing data from the INP file. Particles must actually be run and when a particle gets to a place in the geometry that is not correctly specified, it gets lost – it simply does not know where to go next. If this happens, you will get on the output file a debug print and event-log print for each of ten lost particles before MCNP terminates. The default of ten lost particles for printing and termination can be changed with the LOST card, but this is generally an unwise thing to do.

See page 173 for a more complete discussion of how to use the plotter and set up a problem to flood the geometry with particles to check for geometry errors.

A. Event Log

An event-log print is produced by a lost particle and also by the third and fourth entries on the DBCN card. When a particle gets lost, the history is rerun and event-log printing is turned on during the rerun, making some of the summary information slightly incorrect. The following example is from the file SHELL2 stored under /X6CODE/CTSS/MCNPEX. It is the same as the SHELL problem but all tallies were taken out. SHELL2 runs only two histories and an event log is forced by the DBCN card. The shell is given an importance of two to cause particles to split when they leave the source cell and enter the shell. The event log is reproduced on the next page.

In column 1 of the event log, SRC is source, S is surface, C is collison, T is termination, BNK is return a track from the bank, and r refers to the reaction type used. See Table F.8 in Appendix F for a full description of the TYR Block, which explains the value for r.

event log for particle history no. 1 ijk = 249770827350057

	x	y	z	u	v	w	erg	wgt	cell		nch	nrn
src	0.000+00	0.000+00	0.000+00	2.724-01	-9.578-01	-9.137-02	1.419+01	1.000+00	1			2
s	9.808+01	-3.448+02	-3.289+01	2.724-01	-9.578-01	-9.137-02	1.419+01	5.000-01	2	surf= 1 npa= 1		2
c	1.024+02	-3.601+02	-3.435+01	-1.056-01	-6.225-01	7.755-01	1.369+01	3.853-01	2	14000.51c r=-99	1	11
c	1.022+02	-3.612+02	-3.304+01	9.780-02	-9.637-01	-2.485-01	6.747+00	3.398-01	2	8016.50c r= -1	2	19
t	1.022+02	-3.612+02	-3.304+01	9.760-02	-9.637-01	-2.485-01	6.747+00	3.396-01	2	energy cutoff		19
bnk	9.806+01	-3.448+02	-3.289+01	2.724-01	-9.578-01	-9.137-02	1.419+01	5.000-01	2	s split at	2	19
c	1.007+02	-3.542+02	-3.379+01	8.449-02	2.100-01	-9.740-01	5.715+00	4.437-01	2	8016.50c r= -1	3	27
t	1.007+02	-3.542+02	-3.379+01	6.449-02	2.100-01	-9.740-01	5.715+00	4.437-01	2	energy cutoff		27

1 event log for particle history no. 2 ijk = 276080085645285

	x	y	z	u	v	w	erg	wgt	cell		nch	nrn
src	0.000+00	0.000+00	0.000+00	3.056-01	3.129-01	-8.993-01	1.419+01	1.000+00	1			29
s	1.100+02	1.127+02	-3.237+02	3.056-01	3.129-01	-6.993-01	1.419+01	5.000-01	2	surf= 1 npa= 1		29
c	1.140+02	1.167+02	-3.354+02	2.605-01	1.039-01	-9.599-01	1.416+01	3.853-01	2	14000.51c r=-99	1	36
c	1.149+02	1.171+02	-3.389+02	-6.262-01	5.148-01	-5.655-01	4.571+00	3.417-01	2	8016.50c r= -1	2	44
t	1.149+02	1.171+02	-3.389+02	-6.262-01	5.148-01	-5.855-01	4.571+00	3.417-01	2	energy cutoff		44
bnk	1.100+02	1.127+02	-3.237+02	3.056-01	3.129-01	-8.993-01	1.419+01	5.000-01	2	s split at	29	44
c	1.163+02	1.191+02	-3.422+02	-6.690-01	-3.064-01	-6.772-01	7.107+00	4.437-01	2	8016.50c r= -1	3	54
t	1.163+02	1.191+02	-3.422+02	-6.690-01	-3.064-01	-6.772-01	7.107+00	4.437-01	2	energy cutoff		54

The first neutron starts with the correct parameters and immediately crosses surface 1 into cell 2 as we would expect because cell 1 is a void. The cell importance increases to 2 in cell 2 and the original particle is split into two tracks, one of which is put in the bank (NPA=1) and the other followed. If there had been a four-for-one split instead of two-for-one as we have here, NPA would be 3 indicating one entry into the bank representing three tracks.

The next event is a collision for the track that is being followed. It has an elastic collision in the center of mass system (r is −99) with silicon (14000.51c) in cell 2. Its energy after the collision is 13.69 MeV. A second collision follows that is with oxygen (8016.51c) in the center of mass system, but this time it is inelastic (r=−1) with one neutron out. The energy after collision is 6.747 MeV, which results in a termination because the energy cutoff in the problem is 12 MeV.

At this point the bank is checked for any tracks and one is found that got there as a result of importance sampling. "s split at 2" means the particle was put in the bank at random number nrn=2 from a split occurring at a surface. That track is started at the point where it was created and it has an inelastic collision with oxygen, resulting in its termination due to energy cutoff.

The second source particle is started. It is split, has one collision with silicon and one with oxygen, and dies because of energy cutoff. The second track of this second source neutron is returned from the bank. It has a collision resulting in such an energy loss that it falls below the energy cutoff and is terminated.

By default only 600 lines of the event log are printed for each history. This can be changed by the fifth entry on the DBCN card.

B. Debug Print

In addition to getting the event-log print for a lost particle, you will also get a debug print which will give you additional information. In particular it will tell you what the geometry description is in terms of cell/surface relations at the point the particle get lost. Frequently the problem is an incorrectly specified sense.

As an example of a debug print, if the geometry of Figure 4.1l in Chapter 4, page 275, is specified incorrectly such that the undefined tunnel going off to the right of surface 5 remains, you will get the following print:

```
   lost particle no.  2     no cell found in subroutine newcel

the neutron currently being tracked has reached surface     5.  there
appears to be no cell on the other side of the surface from cell    2
at that point.

the neutron is in cell     2.

x,y,z coordinates:            5.00000e+00    8.15027e-04    1.96005e-01
u,v,w direction cosines:      9.98802e-01    2.03513e-04    4.89425e-02
energy = 1.40000e+01    weight = 1.00000e+00    time = 7.82455e-02
sqrt(y**2+z**2) = 1.96007e-01
the distance to surface    5 from the last event is 2.00240e+00
the distance to collision from the last event is 1.00000+123
the number of collisions so far in this history is     0.

the cells so far found on the other side of surface    5 of cell    2
(and the surface with respect to which the point x,y,z had
the wrong sense) are:  (see chapter 5 of the mcnp manual.)

    3
```

The x,y,z coordinates give the location of the lost particle. If the geometry is plotted with x,y,z as the origin, the geometry in the vicinity of the lost particle can be examined. Dashed lines in the plot indicate the improperly specified portion of the geometry (see page 173).

The last paragraph of the debug print pinpoints the geometry error. The particle has just exited cell 2 by crossing surface 5. The only known cell on the other side of surface 5 from cell 2 is cell 3. However, cell 3 has been defined as (2:−1) (4:5:−3). The particle is in the undefined tunnel region (−2 5), not in cell 3. If cell 3 were only the area to the right of surface 5 and defined without the union operator, the debug print would be even more specific, listing 3 (2) to indicate that the particle has the wrong sense with respect to surface 2 of cell 3.

Appendix A

HOW TO USE MCNP ON CTSS

The executable binary file for Version 3A is called MCNP which is a public file that the user does not have to get. Many of the cross-section libraries are also public files; these include RMCCS, DRMCCS, TMCCS, and MCPLIB. A complete list of all cross-section libraries and their contents is in Appendix G.

Some example XEQ and COSMOS files for automating the following procedures and for running unattended production jobs are illustrated in this Appendix, but in no way are these examples you should necessarily use. They are merely possible ways of doing things, and you should rewrite them to do your particular job in your particular environment. However, you will probably find much useful information in the example files. Any items you will most likely need to change are written in italics.

All files are under the /X6CODE/CTSS node on the Common File System (CFS) at Los Alamos. All the example files mentioned in this Appendix are available from the /X6CODE/CTSS/MCNPEX node.

The latest MCNP Version 3A master backup files (normally not needed by the user) are under the latest directory node on CFS under /X6CODE/CTSS/MCNP3AD. This node contains backups for the public file MCNP and other related files. All the above files have universal read-only access.

If a public file ever suffers from a disk problem (which does happen from time to time with the large cross-section files) and you must run on that particular machine for some reason before X-6 can get the file replaced, the following procedure is simplest. Sign onto a different Cray in the same partition, copy the public cross-section file to your local file space and move the file onto the required machine.

A. MCNP Quick and Easy

This section is for someone who has run MCNP very little. A basic knowledge of CTSS is assumed. If this assumption is wrong, contact the Consultants in C-Division. This section will show you how to run a simple job and refer you to specific pages in this manual where you can find a description of the basics you will need to know to set up your own simple jobs. Quickly learning how to set up and run simple jobs may give you a false sense of confidence, however. Before attempting more difficult jobs (especially those requiring variance reduction techniques) you should study appropriate sections of Chapter 2 and ask X-6 for guidance. This cannot be

emphasized too strongly when using detectors; unfortunately, they are very easy to use but can get you into a lot of trouble.

Press the RETURN key after each input line described in this section.

To run a job with 10 MeV neutrons starting isotropically from the center of a bucket of water where you want to know the current of neutrons getting out the top and also the side of the bucket, get the file BUCKET from /X6CODE/CTSS/MCNPEX. Then simply type in

MCNP INP=BUCKET / 1 *p*

This will produce an output file OUTP that you can examine at your terminal and/or send to a printer by

HSP OUTP -UC -CC T="*your title*

Looking at the output, you will see that 2.016E-01 ± 1.9% neutrons per starting neutron get out the top and 5.647E-01 ± 0.8% get out the side. Adding E1, C1, and T1 input cards with appropriate entries on them will make the results energy-, direction-, and time-dependent, respectively. Note that this bucket has no real walls; the geometry is a chunk of water in the shape of a bucket.

To take a look at the geometry with the PLOT overlay, simply type in

MCNP INP=BUCKET IP / 1 *p*

After you are prompted with a question mark, if you are *not* at an interactive graphics terminal, type in

TERM=0 DISP=GIVE PX=0 EX=20

and at the next prompt type in END; your plot will come out on 35-mm film. If you are at an interactive graphics terminal, at the first prompt simply type in

PX=0 EX=20

This plot will show an intersection of the surfaces of the problem by the plane X=0 with an extent in the x-direction of 20 cm on either side of the origin. If you want to do more with PLOT (such as send the plot to an electrostatic plotter), *see* the instructions in Appendix B.

To understand the input file BUCKET and gain some insight into setting up your own problems, *see* Chapters 1 and 2 about cells and surfaces, thumb through Chapter 3 to get confused about (but exposed to) all the available input cards, and refer to Chapter 4 for examples of specifying tallies and

more exotic geometries. After you have gained a little expertise in the above areas (try doing other things with BUCKET like adding different tallies), spend some time in the remainder of this Appendix to gain more versatility in the mechanics of running MCNP.

A more realistic bucket – also from MCNPEX – than the preceding one is set up in BUCKET2. This bucket has real walls. It would be instructive to plot it and to decipher the input file. This geometry can be plotted exactly like the BUCKET geometry, but you will get a better plot if you use

B=0 1 0 0 0 1 EX=13 *O*=0 0 10

One final example from MCNPEX is the input file LOGO, a legitimate geometry consisting of two cells and fifteen surfaces (including planes of two sheets). Plot it with the command

B 1 0 0 0 1 0 *O* 2800 0 0 EX 3700 L 0

Note that equal marks are omitted from this command and the character just before the 2800 is the letter *O*. After you have deciphered and understood this input file, contact X-6 for your diploma (with honors) from the MCNP geometry school.

B. Running MCNP

This section consists of specifying the MCNP execution line, a simple controller, and a production controller. The MCNP interrupts are explained on page 32.

Upon execution, the first thing MCNP will do is create a drop file named +MCNP (or +MCNPA, etc., if a file of the first name already exists). If MCNP ungracefully terminates because of insufficient time (very unlikely when running particles but more possible when plotting the geometry), just restart the drop file:

+MCNP / *t p*

<u>1. Execution Line</u>:

The MCNP execution line has the following form:

MCNP *Files Options* / *t p*

where *Files* and *Options* are described below (the order on the execution line is irrelevant).

a. Files: MCNP uses several files for input and output. The files most frequently encountered by the user are listed in Table A.1. In addition, cross-section libraries will be accessed as needed. A complete list of the MCNP files is found at the beginning of the main overlay MCNP in the source listing of the code.

Table A.1
MCNP Files

Default Name	Description
INP	Problem input specification
OUTP	Output for printing
SRCTP	KCODE source distribution
RUNTPE	Binary start-restart data
XSDIR	Cross-section directory

The default name of any of the files in the above table can be changed on the MCNP execution line by entering

Default Name_1=User File Name_1 Default Name_2=User File Name_2 etc.

For example, if you have an input file called MCIN and wish the BCD output file to be MCOUT, the execution line would be

MCNP INP=MCIN OUTP=MCOUT / *t p*

If there are no changes in default names, nothing is entered for *Files*.

If a file in your local file space has the same name of a file MCNP needs to create, the file is created with a different unique name by changing the last letter of the name of the new file to the next letter in the alphabet. For example, if you already have an OUTP, MCNP will create OUTQ.

b. Options: There are two kinds of options, overlay execution and other. Execution options are discussed in Chapter 1 on page 31. The other options are shown in Table A.2 and add more flexibility in running MCNP.

As an example of an execution line,

MCNP DBUG 1000 INP=MCIN PRINT / *t p*

results in file MCIN being used for input, all overlays but PLOT being

executed, a full output file OUTP, and a debug statement being written to OUTP every 1000 particles.

After a job has been run with MCNP, the print file OUTP in your local file space may be examined with an editor and/or sent to a printer or microfiche or both.

Table A.2
Other Options

Mnemonic	Operation
C m	Continue a run starting with m^{th} dump. If m is omitted, last dump is used. See page 168.
DBUG n	Write debug information (see DBCN card, page 262) every n particles
NOTEK	Indicates that your terminal has no graphics capability. PLOT output is in local film files. Equivalent to TERM=0 (see page 422)
FATAL	Transport particles and calculate volumes even if fatal errors are found by IMCN
D	Destroy drop file unless there is an abnormal termination of MCNP. Default is do not destroy drop file.
PRINT	Create the full output file; equivalent to PRINT card (see page 265)

2. A Simple Controller

MCNP does not have a standard controller. Except for unattended production runs, MCNP is most commonly run by manually getting the necessary input files in your local files space, executing it as illustrated in the Quick and Easy section and disposing of the output file as desired.

If a controller is desired to automate the process somewhat, something similar to the COSMOS file COS1 in Figure A.1 could be used. It gets an input file TEST1, executes MCNP with the appropriate options, and sends the output file MYOUT to the printer using HSP and to microfiche using FICHE. A COSMOS file is executed by typing in (assuming you have COS1 in your local file space)

COSMOS l=COS1

COS1

```
*MASS GET /X6CODE/CTSS/MCNPEX/TEST1
*MCNP INP=TEST1 OUTP=MYOUT PRINT
*HSP MYOUT -CC -UC T="your title"
*FICHE MYOUT
```

Figure A.1

3. A Production Controller

COSMOS files may be submitted for noninteractive production execution. Three separate routines, SUBMIT, STATUS, and DELETE, are used. You should obtain documentation from C Division for these routines to see how to use them. Figure A.2 is a sample COSMOS file for production use.

COS for COSMOS

```
*/ MAKE SURE /yourplace HAS PROPER ACCESS FOR
*/ STORING FILES; YOU MAY NEED TO ADD PASSWORDS TO COS
*/
*MASS GET WAIT=ON /X6CODE/CTSS/MCNPEX/TEST1
*LET Y=TIMELEFT - 1
*SELECT TASKTIMELIMIT = \Y\
*MCNP INP=TEST1 PRINT
*MASS STORE WAIT=ON DIR=/yourplace RUNTPE +MCNP OUTP
*HSP OUTP -CC -UC T="your title"
*FICHE OUTP
*SELECT PRINTLOG=COSLOG
*MASS STORE WAIT=ON /yourplace/COSLOG
```

Figure A.2

The second MASS execution is simply good, cheap insurance because the necessary files are saved in case the output is lost, a subsequent continue-run is deemed necessary, or a problem needs to be tracked down with DDT. The files can be deleted when everything is satisfactory. The TASKTIMELIMIT is set to run MCNP for a minute less than the global time for the entire job defined when submitting the job to BATCH with the SUBMIT utility. The file COSLOG is a log of everything that has happened to the job and may be useful in reconstructing a problem that developed during production. Everything following the */ is a comment.

C. Files and Updating

There is only one source file (MCNPID) for all systems. We accomplished this by the *IF *ENDIF feature of HISTORIAN, which you do not need to be concerned about unless you are modifying some of the sections of the code where this technique is used. The appearance of *DEFINE CRAY,TSS,CTSS,PLOT,CGS,ACE in a patch will produce a COMPILE file for the CRAY at Los Alamos.

Most of the relevant files for MCNP are on the /X6CODE/CTSS node.

MCNPID	Source listing with Update identifiers
MCNP3A	Update file that created Version 3A from 3
MCNPIX	Master index of variables
MCNPPL	Update program library
MCNPBL	Relocatable binary library
MCNPMP	LOD load map
MCNP3AX	Executable binary equivalent to public MCNP

To print the code listing and index files:

HSP MCNPID –SEQ –UC –V T="*your title*"
HSP MCNPIX –CC –UC T="*your title*

Figure A.3 shows an example of a simple patch that puts two WRITE statements near the beginning of the HSTORY subroutine.

```
*DEFINE CRAY,TSS,CTSS,PLOT,CGS,ACE
*IDENT WLT
*I,HS.8
      WRITE(IUO,1000)
      WRITE(JTTY,1000)
 1000 FORMAT(7HTESTING)
```

Figure A.3

Note that IUO refers to the output file OUTP and JTTY is for output terminal messages. IUI is reserved for the input file INP. IUO is not created until MC.116 and IUI is not opened until EM.41. Make sure a patch you write does not require these I/O units before they are assigned.

If you add or replace an entire subroutine (such as SOURCE), you must have *CALL CM and possibly *CALL SC or *CALL PC or *CALL RC (beginning in column 1) right after the beginning of your new subroutine to get the COMMON blocks in. If you change COMMON simply make the change to the CM

deck (the first few lines of MCNPID) with your patch; you do not have to do anything else.

To add a SOURCE subroutine, you may replace everything from SO.2 through SO.15 with a complete new subroutine (including *CALL CM and *CALL RC) or you may simply replace SO.13 with the necessary FORTRAN lines to define at least XXX, YYY, ZZZ, ICL, JSU, ERG, WGT, IPT, and TME (see page 218). Note that SO.13 must be removed.

To add an SRCDX subroutine, you may also replace everything that is now in MCNP (SX.2 through SX.8) or just replace SX.6 with the necessary FORTRAN lines to define PSC (see page 305 in Chapter 4) for each detector and/or DXTRAN sphere. In any event, SX.6 must be removed.

<u>Caution</u>: The following procedures for modifying MCNP are for relatively simple modifications. If more extensive changes are required (such as adding a COMMON block), see Appendix D. Further, MCNP follows a convention for naming variables to help you avoid conflicts with variables in COMMON blocks which are in almost all subroutines; this is also explained in Appendix D.

When updating MCNP, you must have the file MCNPPL from /X6CODE/CTSS and your patch in your local file space. MCNPPL is the HISTORIAN program library (OLDPL). The patch is prepared based on identifiers from the file MCNPID. Note that each subroutine in MCNP is a separate HISTORIAN deck.

Once you have the necessary files, to get a new executable binary called MCNP,

HISTORN (I=*patchname*,P=MCNPPL,F) / 1 *p*
CFT I=COMPILE,L=0
LDR LIB=(CFTLIB,CGSCFT),X=MCNP,DYN=DAC,CS=70140

Note that the LDR line has X=MCNP rather than X=*yourcode*; however, the latter is perfectly acceptable.

The above CFT line creates a compiler listing called $OUT which contains any FORTRAN error messages. A pattern search through $OUT for ERROR (in upper case) will find the Fortran errors you had in your patch.

The steps outlined above are automated in an XEQ file called XEQ3 that is stored under /X6CODE/CTSS. It gets MCNPPL from CFS and executes the above three lines. The file assumes the name of your patch is PATCH.

Modifying MCNP and then immediately making a long production run is strongly discouraged. Modifications should be made and followed with a short test run first. Then if everything is satisfactory, a long run can be made with the saved executable binary.

To see how we create all the files we have for MCNP (such as MCNPIX), contact X-6.

Appendix B

THE MCNP INTERACTIVE PLOTTER

The PLOT section of MCNP is used to plot two-dimensional slices of a problem geometry specified in the INP file. This feature of MCNP is invaluable for debugging geometries. You should first verify your geometry model with the MCNP plotter before running the transport part of MCNP, especially with a complicated geometry in which it is easy to make mistakes. The time required to plot the geometry model is small compared with the potential time lost working with an erroneous geometry.

The implementation of the plot feature in MCNP may differ slightly from installation to installation. MCNP supports three graphics systems: CGS (Los Alamos Common Graphics System), Tektronix PLOT10, and the ISSCO DISSPLA System. These graphics libraries are device-independent in general and give considerable flexibility in processing graphical output.

A. *Input for the MCNP Plotter*

To run the plot section of MCNP, enter the following command:

MCNP INP=filename IP

The "IP" option on the execution line initiates the plotting section. The most common method of plotting is with an interactive graphics terminal. MCNP will read the input file, perform the normal checks for consistency, and then will prompt for plot input with a question mark.

MCNP can be run in a batch environment without much difficulty, but the user interaction with the plotter is significantly reduced. If you are not using an interactive graphics terminal, use the NOTEK option on the MCNP execution line or set TERM=0 along with other PLOT commands when first prompted by PLOT. Every view you plot will be put in a local film file called FILMF and, by using the ESP command, put in a local electrostatic plotter file called ESPF. The filenames FILMF and ESPF cannot be changed by the user as can other filenames such as INP or OUTP. By default, file FILMF for 35-mm film is always produced and kept in your local file space. FILM=0 turns off the film file creation.

At Los Alamos only, you can automatically give FILMF to the system for processing by setting DISP=GIVE while you are in PLOT. If it is still a local file after MCNP has ended, you can give FILMF to the system manually by entering:

PFILM FILMF FORMAT=35

Both FILMF and ESPF are disposed of automatically through the Los Alamos PAGES system by MCNP if DISP=GIVE is set (default is DISP=KEEP). As a result, they will not exist as local files when MCNP is terminated. FILMF is processed by an FR80 film recorder, and ESPF is processed by a Versatec electrostatic plotter. If you need the full range of PFILM and PESP options, do not set DISP=GIVE. Keep FILMF and ESPF as local files and dispose of them manually, selecting required options of PFILM or PESP at the time of disposition. For the same plots, FILMF and ESPF are identical PAGES metafiles except for a difference in size of the surface label characteristics required by the different plotting devices.

On all systems, the FILM keyword remains in effect and cannot be changed, once set to 35 or 105. An ESP option, once selected, cannot be changed either. Keyword ESP is required for each electrostatic plot desired and must be retyped for each one. The desired option is set at the first use of ESP and need not be retyped.

Input lines are terminated by pressing the RETURN key on the terminal. Lines may be continued with the & symbol. Between a keyword and its data, a blank or an equal "=" symbol can be used as a delimiter. The input to the plotter is completely free form and does not require a decimal point for floating point data. To obtain a list of the input commands, type HELP or H. All input commands are explained starting on page 424. If an error occurs in processing the input, MCNP will request that you RETYPE the entire line of input. Options and parameter settings remain in effect until you change them.

Before describing the individual plotting commands, it will help to explain the mechanics of two-dimensional plotting. To obtain a two-dimensional slice of a geometry, you must decide where the slice should be taken and how much of the slice should be viewed on the terminal screen. The slice is actually a two-dimensional plane that may be arbitrarily oriented in space; therefore, the first problem is to decide the plane position and orientation. In an orthogonal three-dimensional coordinate system the three axes are perpendicular to each other. An orthogonal axis system is defined with a set of BASIS vectors on the two-dimensional plane used to slice the geometry to determine the plot orientation. The first BASIS vector is the horizontal direction on the screen. The second BASIS vector is the vertical direction on the screen. The surface normal for the plane being viewed is perpendicular to the two BASIS vectors.

How much of the slice to view is determined next. The center of the view plane is set with ORIGIN, which serves two purposes: first, for planes not corresponding to simple coordinate planes, it determines the position of the plane being viewed, and second, the origin becomes the center of the cross-sectional slice being viewed. For example, for a Y-Z

plot, the X-coordinate given with the PX command determines the location of the PX plane. The ORIGIN is given as an X, Y, and Z coordinate and is the center of the plot displayed. Because planes are infinite and only a finite area can be displayed at any given time, you must limit the extent of the cross-sectional plane being displayed with the EXTENT command. For instance, a plane defined with $PX=X_1$ at an ORIGIN of X_1, Y_1, and Z_1 would produce a Y-Z plane at $X=X_1$, centered at Y_1 and Z_1 using the default BASIS vectors for a PX plane of 0 1 0 and 0 0 1. If the EXTENT entered is Y_2 and Z_2, the plot displayed would have a horizontal extent from Y_1-Y_2 to Y_1+Y_2 and a vertical extent of Z_1-Z_2 to Z_1+Z_2.

The BASIS vectors are arbitrary vectors in space. This may seem confusing to the new user, but the majority of plots are PX, PY, or PZ planes where the BASIS vectors are defaulted. For the majority of geometry plots, these simple planes are sufficient and you do not have to enter BASIS vectors.

All the plot parameters for the MCNP plotter have defaults. You can respond to the first MCNP prompt with a carriage return and obtain a plot. The plot probably will not be what you wish, but it will nevertheless be a plot. The default plot is a PX plane centered at 0,0,0 with an extent of −100 to +100 on Y and −100 to +100 on Z. The Y axis will be the horizontal axis of the plot, and the Z axis will be the vertical axis. This would be the equivalent to entering the command line:

```
ORIGIN  0 0 0   EXTENT  100 100   BASIS  0 1 0  0 0 1
```

By resetting selected plot parameters, you can obtain any desired plot. Most parameters remain set until you change them. Note that the PX, PY, PZ feature is a shorthand way of obtaining a plot and is accompanied by certain restrictions. If the restrictions are unsatisfactory, the solution is to type out the longer version of the input line with exactly the values you want.

After typing a command to MCNP, you can use either an equal or a space as a delimiter. After the first parameter for a command has been entered, a space delimiter is required for subsequent entries.

Following is a detailed description of each of the MCNP commands. Inside the parentheses adjacent to some of the commands are abbreviations. Those commands listed without abbreviations must be spelled out completely to be accepted by MCNP.

BASIS (B) X_1 Y_1 Z_1 X_2 Y_2 Z_2

The two vectors describing the horizontal and vertical axes of the plot. The default display is a Y-Z plot with basis vectors of 0 1 0 and 0 0 1.

CANCEL

In the event of an error message, this command returns the control to the previous situation.

CENTER (C) DH DV

The CENTER command is normally used to move the center of a plot when using the ZOOM feature, a manual enlargement technique that is an alternative to the CURSOR command. The CENTER command is given with horizontal and vertical coordinates relative to the current plot ORIGIN. DH is the relative horizontal displacement and DV is the relative vertical displacement.

CURSOR

This command has no parameters and causes the graphic cross-hair cursor to appear. It can be used only on interactive graphic terminals. The cursor is used to obtain an enlargement of part of the plot displayed by positioning the cross-hairs in one of the following two ways:
1. Position the cursor at one corner of the desired window and trigger it with any key except <cr>. Then move the cursor to the desired position at the diagonal corner and trigger it again. 2. Position the cursor at any corner of the desired window and trigger it with the 1 key. The new plot will have the same center as the previous one.
When in CURSOR mode, FACTOR is reset to 1, and THETA and CENTER are set to 0. The CURSOR command line may not contain any additional input to reset the EXTENT or ORIGIN.

DISP FD

This command is useful only at Los Alamos for film file disposition. The parameter FD can be either KEEP or GIVE. The KEEP option keeps files FILMF and ESPFin your local file space; the GIVE option gives them to the PAGES System for processing.

END

This command terminates the interactive input session.

ESP VZ

This command produces the file ESPF that can be sent to the electrostatic plotter. The parameter VZ may be:

11 - for 11" electrostatic plotter paper
36 - for 36" electrostatic plotter paper
36V - for 36" vellum paper

EXTENT (EX) EH EV

Half the horizontal dimension and half the vertical dimension of the plot. If only one parameter is entered, it is used as the value for both dimensions. EH is the horizontal extent, and EV is the vertical extent. Unequal EXTENT values will cause the plot to be distorted.

FACTOR (F)

This command is used to vary the magnification of a currently displayed plot. A FACTOR greater than 1.0 results in an enlargement, less than 1.0, a reduction. The center of the current plot is not changed unless the CENTER command is entered. FACTOR is part of the manual ZOOM feature of MCNP and is an alternative to the CURSOR command.

FILM VZ

This command produces a film file FILMF for processing. The parameter VZ may be:

0 - no file produced
35 - 35-mm film file
105 - microfiche

HELP

This command displays the MCNP PLOT commands and abbreviations.

LABEL (L) S C DES

The three input parameters for LABEL are surface label option S, cell label option C, and descriptive data option DES. The default value is to display the problem surface numbers and nothing else. For the parameters S and C, a zero means do not display the label, and a one means display the label. For example,

LABEL 1 0 displays the surface numbers;
LABEL 1 1 displays both the surface and the cell numbers.

The possible values for DES are:

CEL - problem names
IN - neutron importances
IP - photon importances

RHO - atom densities
DEN - mass densities
VOL - volumes
FCN - neutron forced collisions
FCP - photon forced collisions
MAS - masses
PWT - photon production weights
MAT - material numbers
TMPn - Temperatures n
WFNn - Neutron weight window n
WFPn - Photon weight window n

The labels are displayed in I, F, or E format such that the number of characters is minimized and at least 5 digits of precision are kept. The E in the exponential format is omitted. The meaning of the cell labels is displayed in the title block on interactive devices.

LS BR

This command sets the terminal line speed. Acceptable values of BR are 300, 1200, or 9600 baud. The default terminal line speed is 9600 baud.

ORIGIN VX VY VZ

The ORIGIN determines the center of the geometry plot being displayed. The coordinates entered are in the three-dimensional coordinate system of the geometry model.

PX VX

This command requests a Y-Z plot by specifying a PX plane at X=VX. The default BASIS vectors are 0 1 0 and 0 0 1.

PY VY

This command requests an X-Z plot by specifying a PY plane at Y=VY. The default BASIS vectors are 1 0 0 and 0 0 1.

PZ VZ

This command requests an X-Y plot by specifying a PZ plane at Z=VZ. The default BASIS vectors are 1 0 0 and 0 1 0.

RESTORE (R)

This command returns you to the window in use before the CURSOR command was invoked. RESTORE should not be used with any additional commands on the same line of input.

TERM VZ

This command sets the terminal characteristics to your particular graphics terminal. The options for VZ are:

1 for Tektronix 4012,
2 for Tektronix 4014 without enhanced graphics,
3 for Tektronix 4014 with enhanced graphics, and
0 for the same effect as NOTEK on the MCNP execution line.

THETA (T) TH

This command causes the displayed plot to be rotated through the specified number of degrees, either positive or negative. A positive value causes a counterclockwise rotation.

VIEW (V)

This command, when used alone as the first command of a plotting session, draws a cross-sectional view of the geometry with the following default parameters:

O= 0. 0. 0. B= 0 1 0 0 0 1 EX= 100. 100.

It has the same effect as pressing the RETURN key after the first prompt.

WHAT (W)

This command displays current settings of all interactive input commands.

ZOOM (Z)

The ZOOM option varys the magnification and orientation of a displayed plot. It is a manual technique controlled by setting FACTOR, CENTER, and THETA. Plot enlargements can also be obtained by using the CURSOR command.

B. Geometry Debugging and Plot Orientation

Surfaces appearing on a plot as dashed lines usually indicate that adjoining space is improperly defined, although they may appear because of a cookie cutter cell or because the plot plane corresponds to a bounding planar surface. Dashed lines caused by a geometry error can indicate space that has been defined in more than one cell or space that has never been defined. The reason for the presence of dashed lines on an MCNP plot should be understood before running a problem.

When checking a geometry model, errors may not appear on the two-dimensional slice chosen, but one or more particles will get lost in tracking. To find the modeling error, use the coordinates and trajectory of the particle when it got lost. Entering the particle coordinates as the ORIGIN and the particle trajectory as the first basis vector will result in a plot displaying the problem space.

The ORIGIN, EXTENT, and BASIS vectors all define a space called the plot *window* (in particular, the window that appears on the terminal screen). The window is a rectangular plane twice the length and width of EXTENT, centered about the point defined by ORIGIN. The first BASIS vector B1 is along the horizontal axis of the plot window and points toward the right side of the window. The second BASIS vector B2 is along the vertical axis of the plot window and points toward the top of the window.

The signs are determined by the direction of the vectors; in particular, do the vector components point in the ±x, ±y, or ±z direction? After signs have been fixed, determine the magnitudes of the vector components. Assume the vector is parallel to the x-axis. It has no y-component and no z-component so the vector would be 1 0 0. If there is no x-component but both y and z, and y and z have equal magnitudes, the vector would be 0 1 1. The vector does not have to be normalized. If the angle between the vector and the axes is known, the user can use the sine and cosine of the angle to determine the magnitude of the components. A rough approximation will probably be sufficient.

C. Sample Plotting Exercise

The purpose of this section is to illustrate the operation of the MCNP geometry plotter using a rectangular duct geometry inside a box. A three-dimensional isometric drawing of the geometry is shown in Figure B.1, as well as the input file.

APPENDIX B
PLOT

```
title card
    1 0 1 -2 3 -4 5 -6:7 -2 3 -4 8 -5 : 2 -9 3 -4 8 -10
    2 0 1 -9 11 -12 13 -14 (-1:2:-3:4:-5:6)
        (-7:2:-3:4:-8:5) (-2:9:-3:4:-8:10)
    3 0 -1:9:-11:12:-13:14

    1 px 0
    2 px 50
    3 py -5
    4 py 5
    5 pz 25
    6 pz 35
    7 px 40
    8 pz 0
    9 px 100
   10 pz 10
   11 py -50
   12 py 50
   13 pz -50
   14 pz 50
```

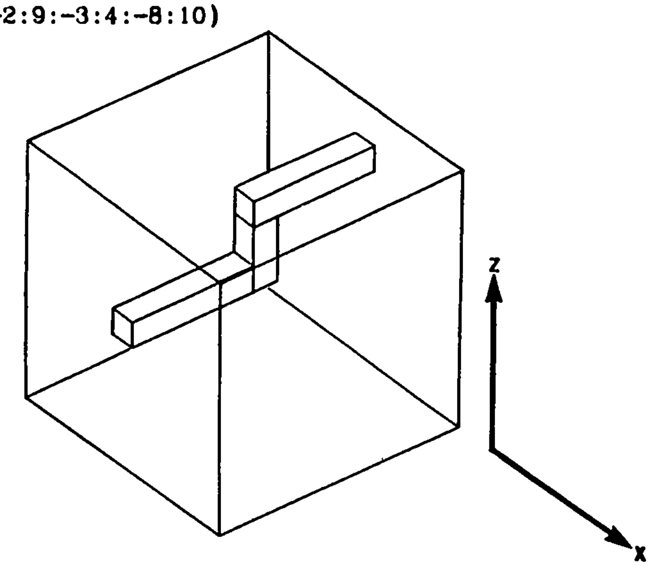


Figure B.1

The first slice of the geometry is shown in Figure B.2. This figure was obtained by entering the following command:

PY 0 0 50 0 0 EX 60 L 1 1

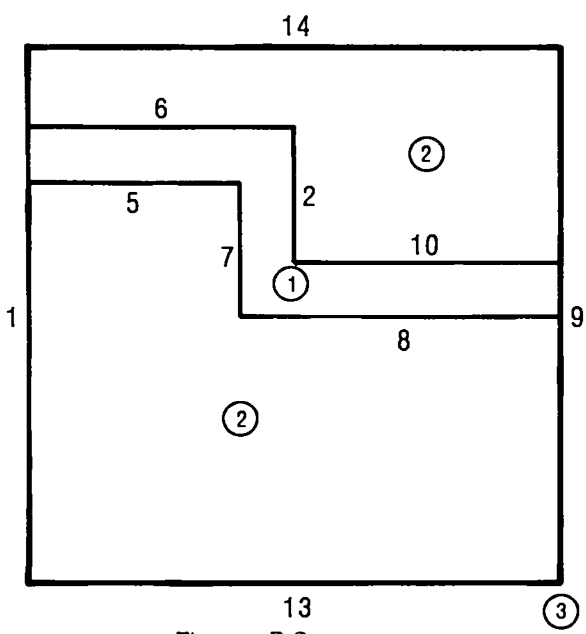


Figure B.2

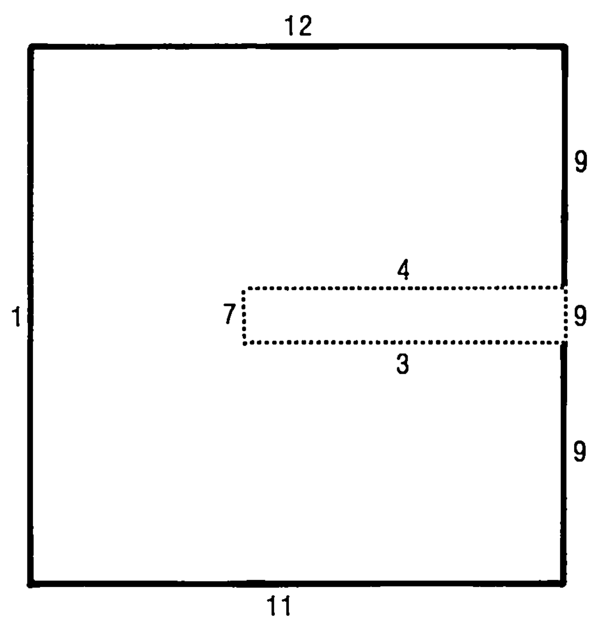


Figure B.3

and results in an X-Z cross section centered at x=50, y=0, z=0. The horizontal dimensions extend from x=-10 to x=110 and the vertical dimensions extend from z=-60 to z=+60. The label command causes both surface and cell problem numbers to be printed. This cross section gives the best indication of the true geometry of the bent rectangular duct.

Figure B.3 is an X-Y plane produced by the command:

```
B=1 0 0 0 1 0    O=50 0 0    EX=60
```

The dashed lines are caused by the coincidence of the plot plane at z=0 with a geometry plane (surface 8), not a real geometry error. When a plotting plane is a bounding planar surface, it is not clear whether the cell boundaries plotted should be on the positive or negative sense of the plane. Moving the plotting plane from z=0 to z=1 with the command O=50 0 1 makes the dashed lines appear as solid.

By methodically marching through the bent duct geometry for the z axis at various elevations and marching through the geometry for the x axis, MCNP is able to verify the completeness of the model. The cross section at y=0 in Figure B.2 gives the best illustration of the model, but additional cuts were necessary to verify other parts of the space being modeled. A cross-sectional slice is only a two-dimensional verification of the model, and for completeness users should take many different slices of the geometry to verify all portions of a model. Suggested input lines to show the various slices you might take to verify this model, continuing from Figure B.3, are:

```
O=50 0 20       shows center vertical section of duct;
O=50 0 30       shows upper portion of duct;
B=0 1 0 0 0 1  O=1 0 0    upper elevated portion of bent duct
O=45 0 0        center vertical portion
O=90 0 0        lower portion of bent duct.
```

D. Electrostatic Plots at Los Alamos

To obtain either 11-inch or 36-inch electrostatic plot output, you should follow the procedure given below:

1. Create a plot with MCNP.
2. Type ESP B, where B is 11, 36, or 36V at the first use of ESP after each plot you want sent to the plotter.
3. Use the utility routine PESP to dispose of the ESPF file if DISP=KEEP.

For an electrostatic plot, the LABEL keyword controls the size of the printed surface or cell labels. Normally, a zero parameter designates no

label and a one parameter designates a label. Any other value is a multiplication factor for the label size relative to the default size of 0.095 inches in height for 36" plots and 0.073 for 11" plots. L may be between 0.3 and 200. For nonelectrostatic plots, the appropriate character size is used for the plot device being used.

E. Running MCNP with Different Graphics Terminals

The MCNP plotter is designed to run on Tektronix 4012, 4014, and 4014E terminals; therefore, you must use terminals that are Tektronix compatible. Many software vendors sell Tektronix simulation systems that allow various terminals to respond to Tektronix graphics instructions. On Tektronix terminals that have separate dialogue and graphics areas, you should disable the dialogue area to allow surface and cell labels to be written onto the graphics area. MCNP has been run extensively at Los Alamos with Tektronix terminals and with terminals such as the DEC VT 100 with a Retro Graphics board simulating Tektronix. Other terminals such as the IBM PC, APPLE MACINTOSH, SUN, and APOLLO have been used with Tektronix simulation packages at Los Alamos. Most terminals that provide a graphics capability are compatible with Tektronix graphics drivers.

The compatibility of graphics terminals with MCNP is independent of the particular graphics libraries MCNP is using. Fortunately, most graphics terminals are becoming standardized for line plots, so incompatibility is not a common problem if good terminal drivers are available.

Appendix C

INSTALLING MCNP

This appendix explains how to install MCNP on various computing systems, starting with the MCNP code package distributed by the Radiation Shielding Information Center (RSIC) of the Oak Ridge National Laboratory.

The files in the MCNP code package are:

MCNP	FORTRAN source for MCNP
PRPR	FORTRAN source for PRPR
MAKXSF	FORTRAN source for MAKXSF
MCNPIX	Index of symbolic names in MCNP
SPECS	Sample input file for MAKXSF
XSDIR	Cross-section directory file
MCPROB	Sample problem input decks
	Files of cross-section tables

All of these files are sequential formatted files with record lengths no greater than 80 characters. They are suitable for examination and modification with any text editor utility program. All letters in all files are in lower case. The requirements of your computer system may make it necessary for you to convert the letters to upper case before you proceed. The file MCPROB contains some sample problem input decks that may be separated with any text editor.

We strongly encourage you to have a printed listing of MCNPIX in front of you if you are going to modify MCNP in any way.

I. PREPARING THE MCNP SOURCE FILE FOR COMPILING

You must preprocess the MCNP source file before you can compile it. The preprocessor inserts comdecks and deletes the sections of system dependent code that are not appropriate for your particular computer system. For the preprocessor you may use either CDC's UPDATE,[49] OpCode's HISTORIAN,[50] or MCNP's own preprocessor PRPR. If you are going to use PRPR, the first step is to compile and load PRPR, which is short, is written in pure FORTRAN 77, and contains no system dependent features. You should have no trouble compiling it.

Before running the preprocessor, you will have to make some changes in the MCNP source file. At the very least you will need to add a *DEFINE directive so that the preprocessor can tell which system dependent sections of code are to be retained and which are to be deleted. The preproccessor retains or deletes sections of code according to *DEFINE, *IF DEF, and

*ENDIF directives. The *IF DEF and *ENDIF directives are already in the MCNP source file. You need to add the *DEFINE directive yourself. The form of the directive is *DEFINE name1,name2, ... The names are chosen from the list below. The directive must start in column 1 and must be a new line of text inserted before the first line of the MCNP source file.

Names for operating systems

CTSS	The time-sharing system for CRAY computers
TSS	Also required when installing on CTSS
COS	Another operating system for CRAY computers
VMS	The operating system for Digital Equipment VAX computers
NOS	The operating system for Control Data computers
IBM	Operating systems for IBM computers

Names for hardware

CRAY	Computers from Cray Research
LCM	CDC machines with LCM
ECS	CDC machines with ECS or ESM
CHEAP	Computers having 32-bit numeric storage units

Names for memory

FIXED	Non-adjustable or virtual memory

Names for plotting

PLOT	To include the plotting capability
CGS	Los Alamos Common Graphics System
PLOT10	Tektronix PLOT10 graphics system
DISPLA	ISSCO DISSPLA graphics system
DISS1	For initialization of DISSPLA via CGS
DISS2	For initialization of DISSPLA by other means

Other names

MFE	For CTSS at NMFECC
ACE	For ACE-format data files (used only at Los Alamos)

If your computer system has either virtual memory or fixed memory with no way for a program to change the amount of memory available to it once it is loaded, you need to include the name FIXED in the *DEFINE directive. You may also need to reset the value of the named constant MDAS, which is declared in a PARAMETER statement in comdeck ZC near the beginning of the MCNP source file. The default value of MDAS is 1000000. MDAS must be big enough for all the data required for your biggest problem but not so big as to violate any real or administrative limits on the amount of memory that may be used by jobs that run in your computer system. It may not be easy to accomodate those requirements. One possible compromise might be to use a relatively small value of MDAS for most of your work and to compile special versions of MCNP for problems that require exceptionally large amounts of data storage. MDAS determines the upper dimension bound of the

array DAS, which is used to hold the geometry description, the tally blocks, the cross-section data, and other miscellaneous dynamically allocated data. In most problems the cross-section data are by far the biggest part of the data in DAS.

Some users install MCNP on two systems, a large system such as a CRAY for production work and a small interactive system such as a VAX for plotting. Since the plotting capability does not use the cross-section tables, the VAX version could have a small value of MDAS, perhaps 100000.

If your computer system has a way for programs to adjust the amount of memory they use, you may want to add an appropriate section of code for your system to the subroutine CHGMEM, which is near the end of the MCNP source file. CHGMEM already has sections of code for CTSS and NOS. Alternatively you may choose to run as if your memory adjustment capability did not exist, that is, to include FIXED on the *DEFINE directive and to set MDAS to a value adequate for your problems.

If you will be using the DISSPLA graphics library and do not have CGS available for DISSPLA initialization, you will need to add an appropriate section of code to the MM=1 part of subroutine DISSF. You may also need to change the MM=2 part which contains calls of the PLOT10 routines ANMODE and HOME for flushing the graphics output buffer. Consult your local DISSPLA experts to find out how graphics initialization and buffer flushing are accomplished on your computer system.

One more change may be required before preprocessing. If you are using PRPR, change the name of the MCNP source file to CODEF because PRPR does not have any means for the user to specify the name of the file it is to work on.

The preprocessor will produce an output file called COMPILE. COMPILE is the MCNP source file tailored to match your computer system. It is ready to be compiled and loaded for execution.

II. CONVERTING CROSS-SECTION FILES WITH MAKXSF

MCNP can read cross-section data from either of two types of file. One type of cross-section file, called Type 1, is formatted and has sequential access. The other type, called Type 2, is unformatted and has direct access. The cross-section files distributed by RSIC are all Type 1 files because Type 1 files can be made to be portable. But reading large formatted files is slow and formatted files are more bulky than unformatted files. The portable auxiliary program MAKXSF has been provided for translating big, slow, portable, Type 1 files into compact, fast, unportable (but still in compliance with FORTRAN 77), Type 2 files. You may also use MAKXSF to delete cross-section tables that you do not need and

to reorganize the cross-section tables into files in a way that is most convenient for the type of work you are doing.

MAKXSF must be preprocessed and compiled in a manner similar to that described above for MCNP. It should not be necessary to make any changes in the MAKXSF file other than to add a *DEFINE directive. The following names are available for the *DEFINE directive:

CTSS	The time-sharing system for CRAY computers
TSS	Also required when installing on CTSS
IBM	Operating systems for IBM computers
CHEAP	Computers having 32-bit numeric storage units

The input files to MAKXSF are one or more existing cross-section files, a directory file that describes the input cross-section files, and a file called SPECS that tells MAKXSF what it is supposed to do. The output files are one or more new cross-section files, a new directory file that describes the new cross-section files, and a file called TPRINT that contains any error messages generated during the run. The input and output cross-section files may be any combination of Type 1 and Type 2 files. The various types of cross-section files and the form and contents of a cross-section directory file are described in detail in Appendix F. The directory file XSDIR in the MCNP code package contains complete descriptions of all of the cross-section files in that package. You might print XSDIR and keep the listing as a reference that will tell you what cross-section tables you actually have on hand. The sample SPECS file in the MCNP code package is provided not only as an example of the correct form for a SPECS file but also as one that will be immediately useful to many users. With SPECS and MAKXSF you can create a set of Type 2 files that contain the same cross-section tables that are in the Type 1 files in the MCNP code package.

The SPECS file is a formatted sequential file with records not exceeding 80 characters in length. The data items in each record may start in any column and are delimited by blanks. The contents of the file are as follows:

Record	Contents	
1	Name of old directory file	Name of new directory file
2	New cross-section file Type	[Record Length Entries per Record]
3	Access Route (may be blank)	
4 +	List of tables for the new cross-section file	
Blank record		

Records 2 through the blank record may be repeated any number of times with data for additional new cross-section files. If the Type of the new

cross-section file is specified to be 1 in record 2, only the name of the new cross-section file and the 1 for the Type are required in that record. If the Type in record 2 is 2, the Record Length and the number of Entries per Record are also required. The Record Length is in 8-bit bytes on CRAY, IBM, and PRIME systems and in words on NOS and VMS systems. If the Record Length is in words, it must be set equal to the number of Entries per Record. If the Record Length is in bytes, it must be set equal to either four times the number of Entries per Record (for systems with 32-bit numeric storage units) or eight times the number of Entries per Record (for systems with 64-bit numeric storage units). The best value to use for the number of Entries per Record depends on the characteristics of the secondary storage, usually disks, on your computer system. If the number of entries is too large, there will be a lot of wasted space in the file because of the partial record at the end of each cross-section table. If the number of entries is too small, reading may be slow because of the large number of accesses. For many systems 512 is a good value. If you intend to use the SPECS file from the MCNP code package, be sure that the values of the Record Length and number of Entries per Record are suitable for your system. The Access Route on record 3 is presently useful only at Los Alamos. It is used by subroutine FFETCH in MCNP to find cross-section files in the Los Alamos Common File System. If you write a section of code for FFETCH that is suitable for your computer system, you might also use the Access Route entry advantageously.

It is not necessary to generate all the cross-section files that you will ever need in one MAKXSF run. You can combine and edit directory files at any time with a text editor. The only requirement is that you must give MCNP a directory file that points to all the cross-section tables that are needed by the current problem. If you plan to run a long series of MCNP problems that all use the same small set of cross-section tables, it might be convenient to generate, with MAKXSF, a small special-purpose cross-section file and directory file just for your project. It might be easier than fetching several large cross-section files from your external mass storage system every few days just to run one problem. Remember that it is not necessary to segregate different classes of cross-section tables in different files. If you want to put a collection of continuous-energy neutron tables, thermal tables, and photon tables all on the same file, you may do so.

There is another good use for MAKXSF that has nothing to do with cross-section tables, which is to use it as a test code to see whether your computer system fully supports FORTRAN 77. You might compile MAKXSF and convert the Type 1 cross-section files to Type 2 before tackling MCNP. Although MAKXSF is a small program of less than 400 lines, it uses many of the new FORTRAN 77 statements such as INQUIRE, OPEN, and CLOSE. It uses character variables extensively and performs many internal reads and

writes. The small size of MAKXSF makes it more convenient than MCNP for this testing purpose.

III. IMPLEMENTATION OF MCNP ON SPECIFIC SYSTEMS

A. CTSS

The *DEFINE directive for CTSS is:

*DEFINE TSS,CTSS,CRAY,PLOT,CGS

The following procedure will compile, load, and execute PRPR and MCNP. It assumes that the CGS graphics library CGSCFT is available, that PRPR, MCNP, XSDIR, cross-section files, and TEST1 are among your local files, and that the *DEFINE directive has been added to the MCNP source file.

```
CFT  I=PRPR,L=0
LDR  LIB=CFTLIB,X=PRPRX
SWITCH   MCNP  CODEF
PRPRX
CFT  I=COMPILE,L=0
LDR  LIB=(CFTLIB,CGSCFT),X=MCNP,CS=70140,DYN=DAC
MCNP  INP=TEST1
```

B. COS

The Cray Operating System is a pure batch environment, making it impossible to use the interactive geometry plotting capability in MCNP. The *DEFINE directive is just

*DEFINE COS,CRAY

The primary consideration in running MCNP on COS is the need to handle input and output files in a mode convenient in a batch environment. The MESSAGE block option in the INP file, page 165, provides a way to substitute file names. OUTP=$OUT would cause output to go to the system output file $OUT. Alternatively the OUTP file could be copied to $OUT after the MCNP task was completed, but output might be lost in an abnormal termination. Substituting INP=$IN is not possible and might not be desirable anyway because INP is read twice by MCNP with a rewind between the two reads.

C. VMS

The *DEFINE directive for a VAX with VMS is

*DEFINE VMS,CHEAP,FIXED,PLOT,pp

where pp is either CGS, PLOT10, or DISPLA, depending on which graphics library is available. If DISPLA is used, it must be followed by DISS1 or DISS2, depending on whether CGS is being used for initialization of DISSPLA.

A typical absolute generation on the VAX under the VMS operating system is accomplished with:

```
LINK/exe=MCNP MCNP,disspla9/lib,cdiv←old:cgsvax/lib
```

In this command, the object code for MCNP exists as MCNP.OBJ. The ISSCO DISSPLA library and the Los Alamos CGS library are referenced. The executable file generated is named MCNP.EXE.

For running MCNP on the VAX, it is convenient to define a global symbol MCNP in the user's LOGIN.COM file. For example:

```
MCNP :== "$DRB2:[WEST.MCNP]MCNP.EXE"
```

Here the symbol MCNP references a disk named DRB2 containing a directory named WEST.MCNP with a file called MCNP.EXE. With MCNP defined, the execution line is simply MCNP followed by files and options as discussed in Appendix A. On the VAX, the interrupt feature of MCNP is as described on page 32 with the exception that the (ctrl e) used on CTSS is replaced by (ctrl c).

When MCNP runs on a VAX, the time interrupt subroutine uses actual wall clock time and not the CPU time. This may present some problems for VAX users on heavily used systems where the discrepancy between wall time and processor time is great.

D. NOS

The *DEFINE directive for NOS on a CDC machine that has LCM is:

```
*DEFINE NOS,LCM,PLOT,pp
```

where pp is either CGS, PLOT10, or DISPLA, depending on which graphics library is available. If DISPLA is used, it must be followed by DISS1 or DISS2, depending on whether CGS is being used for initialization of DISSPLA.

A NOS procedure for compiling and executing PRPR and MCNP follows. The file MCNPS is the MCNP source file with the *DEFINE directive inserted.

```
ATTACH,CODEF=MCNPS.
GET,PRPR.
FTN5,I=PRPR,L=0,GO.
REWIND,COMPILE.
```

```
FTN5,I=COMPILE,LCM=G,L=0.
ATTACH,CGSLIB/UN=LIBRARY.
LIBRARY,CGSLIB.
GET,TREE.
DEFINE,MCNP.
ASSIGN,MS,OUTPUT.
RFL,EC=1400.
LDSET,PRESET=NGINDEF.
LDR>?  SEGLOAD,I=TREE,B=MCNP.
LDR>?  LOAD,LGO.
LDR>?  NOGO.
REWIND,*.
ASSIGN,TT,OUTPUT.
ATTACH,XSDIR,[cross-section files].
GET,TEST1.
ASSIGN,TT,TEKT.
MCNP.  INP=TEST1
ROUTE,OUTP,DC=PR.
```

The file TREE contains segmenting directives:

```
MCNP    COMMON    FIXCOM,VARCOM,NBVCOM,EPHCOM,PBLCOM,TABLES
MCNP    COMMON    CHARCM,PB9COM
IMCN    GLOBAL    IMCCOM,JMCCOM
PLOT    GLOBAL    PLTCOM
MCRUN   GLOBAL    RUNCOM
LAST    GLOBAL    DAC
MCNP    INCLUDE   BLKDAT
IMCN    INCLUDE   IBLDAT
PLOT    INCLUDE   PBLDAT
MCRUN   INCLUDE   RBLDAT
AA      TREE      IMCN-(PASS1,RDPROB,IGEOM,ISOURC,ITALLY,VOLUME)
BB      TREE      PLOT-(SETUP,WRAPUP)
CC      TREE      XACT- GETXST
DD      TREE      MCRUN-(TRNSPT,OUTPUT)
PEAR    TREE      MCNP-(AA,BB,CC,DD)
        LEVEL
APPLE   TREE      LAST
        END       MCNP
```

These directives set up a segment structure to allow MCNP to run in a machine with a small memory. This is particularly important when using the DISSPLA graphics library. Modifications may be required for the COMMON directives, depending on whether the computer has LCM.

The period following the word MCNP on the execution line is required. Any characters after the first occurrance of a period in a NOS command are treated as comments by the operating system, but the entire line is made available to the executing program, so MCNP is able to get any files or options specified there.

If the geometry plotting feature of MCNP is used, the command ASSIGN,TT,TEKT. causes the graphics information that CGS writes on file TEKT to be sent directly to the user's terminal.

E. IBM

Implementation of MCNP on IBM systems will be described in the forthcoming Oak Ridge National Laboratory report, "Systems Guide to MCNP Version 3A", by Bernadette L. Kirk.

F. Other systems

MCNP has been run on several modern small systems: Ridge, Prime, Apollo, and Sun. It would probably even work on a Macintosh Plus with a hard disk. We do not have specific details for installation on these systems. The general installation procedure described in the first half of this Appendix has been sufficient in each case so far.

Appendix D

MODIFYING MCNP

Users sometimes have to modify MCNP for particular applications. In the past, most user modifications were for special sources or special tallies. The need for tally modifications has been greatly reduced by the generalization of the standard tallies in MCNP Versions 2 and 2B. The generalization of the standard sources in Version 3A should do the same for source modifications. However, users continue to find new applications for MCNP and will find new reasons to modify it.

This appendix contains information that users will need when they write modifications to MCNP. Other sections of this manual are also applicable, especially Chapter 2 for theory, Appendix E for variables and arrays in common, and Appendix F for the details of the cross-section tables.

This appendix is written with the assumption that the reader has a listing of MCNP, such as the file MCNPID, open in front of him so he can look at the sections of code referred to in the text. A listing of MCNPIX, the index of symbolic names, is useful also and for non-Los Alamos users it is included in the RSIC package of files.

CONTENTS OF APPENDIX D

I. PREPROCESSORS

Before MCNP is compiled, it must be preprocessed to distribute the comdecks and to delete inappropriate system-dependent sections of code. The preprocessor may be either CDC's UPDATE[49], OpCode's HISTORIAN[50], or MCNP's own preprocessor PRPR. HISTORIAN and UPDATE may also be used to modify MCNP by means of a program library file and a patch file. MCNP

contains the *DECK directives necessary for the creation of an UPDATE or HISTORIAN program library. See Appendix C for information on the *DEFINE directive required for selecting the appropriate system-dependent sections and on how to load MCNP on the various systems.

At Los Alamos and NMFECC, a current MCNP program library file MCNPPL and a program listing file MCNPID are maintained for the convenience of users who need to modify MCNP. MCNPPL is for HISTORIAN, not UPDATE. MCNPID is a listing of the current version of MCNP with HISTORIAN line identifiers.

II. PROGRAMMING LANGUAGE

MCNP is written mostly in standard FORTRAN.[51] Deviations from the standard are avoided because they make it more difficult to maintain portability. MCNP programming currently deviates from the standard in the following areas: dynamically allocated storage, zeroing and moving common blocks, system-dependent features, and system peculiarities.

Every dynamically allocated storage array in MCNP has an offset that is added to the first subscript expression in every reference to the array. This causes the value of the subscript expression to exceed the corresponding upper dimension bound for the array, which violates a FORTRAN rule. So far this has not caused trouble because the systems that MCNP currently runs on do not enforce the rule dynamically. The rule can not be enforced at compile time because the offset is a variable. If the situation changes and it becomes necessary to abandon the offset scheme, we will have to use the well-known statement-function scheme, that we have avoided until now because it seems to be more clumsy.

Whole common blocks are zeroed in MCNP by equivalencing an integer array of the correct length to the common block and defining the values of the elements of the array to be zero by a DATA statement or by a DO-loop. The array is also used in I/O lists and in some DO-loops that copy it to similar arrays. This is illegal in FORTRAN but works because the equivalenced arrays and variables are actually stored in the same places in memory and because the hardware representation of zero is the same for real, integer, and both parts of double-precision entities on all systems on which MCNP presently runs.

Some special features of MCNP cannot be provided within the FORTRAN language. See Sec. IV of this appendix for a list of them. The special features are implemented by calling subroutines in the local libraries in the various computing facilities where MCNP is used. Some of the subroutine calls or preparations for the calls require nonstandard language. For example, the statement that sets the character size on a graphics terminal in the CGS graphics system is CALL GESCPE(5HCSIZE,1.,1).

The argument 5HCSIZE is not a standard form for any kind of FORTRAN constant. Another example is the statement that fetches the execute-line message in the DEC VMS system: CALL LIB$GET←FOREIGN(HM,,). There are too many characters in the name of this subroutine, it contains some illegal characters, and two of the arguments are void.

The one system peculiarity that requires nonstandard language is the two-level memory on some CDC machines. LEVEL statements are required to put common blocks into LCM or ECS.

III. SYMBOLIC NAMES

The file MCNPIX lists all symbolic names used in MCNP and tells how many times each name appears in each subroutine that uses it. Refer to MCNPIX to keep from introducing names that would conflict with existing names.

In MCNP, the name of every entity in common and the name of every function subprogram is at least three characters long. The name of every local entity, including statement functions, is less than three characters long. Thus, the local or global status of a symbolic name can be determined at a glance.

The default implicit typing of FORTRAN is used for all integer and real entities in MCNP. When MCNP is compiled on any 32-bit computer, the statement IMPLICIT DOUBLE PRECISION (A-H,O-Z) is included in all program units. There are no complex entities in MCNP nor are there any double-precision entities other than when double precision is used instead of real on 32-bit machines. Logical entities are rare and are always local. The names of most, not all, character entities begin with the letter H.

IV. SYSTEM DEPENDENCE

The use of standard FORTRAN goes a long way by itself toward making MCNP run on many different computer systems. However, differences between the systems still have to be allowed for to some extent.

The most important difference between hardware systems is that some have 60-bit or 64-bit words, whereas others, such as IBM and DEC machines, have 32-bit words. MCNP assumes that no more than 32 bits are available for integer quantities. This is awkward but so far has not been a serious limitation. MCNP assumes that at least 48 bits of precision are available for floating-point quantities. This requires double precision on 32-bit machines. Geometry tracking in MCNP uses floating-point quantities without any special allowance for the fact that they are only approximations to the

mathematical real numbers that they represent. This turns out to be a safe practice if the floating-point numbers have 48 bits of precision but not with much less than 48. The accuracy of cross-section data is so low that they could be represented adequately by 32-bit floating-point values, and because most of the memory used by a typical MCNP problem is filled with cross-section tables, it is desirable to use 32-bit words for them. When 32-bit words are used for cross sections, problems sometimes fail to track.

The magnitude of a floating-point number cannot exceed about 10^{38} in most 32-bit machines. It has been necessary in writing some parts of MCNP to be careful that intermediate values do not exceed that limit. There are probably still sections of MCNP that can fail by trying to generate numbers greater than 10^{38}.

On some IBM machines, computation is significantly slower if double-precision values do not start on the correct sort of boundaries in memory. This peculiarity is accommodated without any system-dependent code by putting the floating-point entities ahead of the integer entities in most of the statically allocated common blocks. In /PBLCOM/ and /PB9COM/, where there are some additional constraints on the order, the integer entities are in even-length sequences, with a dummy variable included if necessary.

The vector capability of Cray computers is a major hardware peculiarity which might speed up MCNP if we could find a way to exploit it. The attempts made so far to vectorize MCNP have not been successful. They made it run more slowly. Part of the trouble is that Monte Carlo itself resists vectorization, especially with continuous-energy cross-section tables. Part of the trouble is that MCNP is a general-purpose program with a great many options that are implemented in hordes of IF statements. The one place in MCNP where there is some system-dependent code to facilitate vectorization is in subroutine TALSHF. The list-scoring parameter FTLS is affected by this bit of vectorization and has a special value in the Cray case. Only in rare problems does any of this make any significant difference.

The two-level memory on some CDC machines is a hardware peculiarity with a substantial impact on MCNP. The small size of the executable memory makes overlays necessary, and LEVEL statements have to be used to put some of the Common blocks into LCM or ECS.

The various systems use different file names to represent the user's terminal:

CTSS	TTY for both input and output
NOS and IBM	INPUT and OUTPUT
VMS	SYS$INPUT and SYS$OUTPUT
COS	terminal access not available

The FORTRAN standard allows I/O units to be preconnected, which means that MCNP must avoid using certain unit numbers. Fortunately the preconnected unit numbers in all systems that MCNP currently runs on are numbers less than 10 or greater than 99. To avoid them, MCNP uses unit numbers in the thirties, forties, and fifties.

The PRIME compiler allows a maximum of 64 labels in a computed GO TO statement. This peculiarity is accommodated without any system-dependent code by splitting the long computed GO TO statements into more than one statement.

CDC's FTN5 compiler does not provide for direct use of data in ECS as it does for LCM. The rather awkward scheme that we use to get around this limitation uses a subprogram function YSS that contains a MOVLEV call to fetch a single word from ECS. The only data that MCNP ever puts in ECS are the cross-section tables. They are completely processed one at a time by XACT while they are in SCM. After they are put in ECS, all references to them are read-only. The symbol YSS is used everywhere that cross-section tables are referenced in MCRUN. The function YSS is included in MCNP only on ECS machines. On other machines YSS is equivalenced to XSS, which is the usual symbol for the cross-section tables in dynamically allocated storage.

The FORTRAN standard does not specify the units for the length of the records of a direct-access file. Some systems define the length in bytes, some in words. This inconsistency does not affect the portability of MCNP. Direct access is used only for Type 2 cross-section files. The record length is read from the cross-section directory file and is entered explicitly in the input file to the auxiliary program MAKXSF, which writes the Type 2 files and the cross-section directory file. The question of the units occurs at the same time that the user chooses the size of the records, all in the context of the local system.

We have encountered bugs in compilers. Some of the comments in MCNP have CA in columns 1 and 2. These comments identify places where unusual programming has been done to get around compiler bugs.

System-dependent sections of code are set off by the preprocessor directives

*IF DEF,name ... *ENDIF or *IF DEF,name,n

See Appendix C for the names that are used and for how to use the preprocessors. As much as possible, we have tried to gather the system-dependent code in MCNP into only a few places, away from heavily mathematical parts of the program. One technique, exemplified by subroutines GRAFIX and GETIDT, is to write a subroutine to do just one or several closely related system-dependent tasks. A subroutine of this sort consists of several alternative sections of code, one for each of the different systems. When that technique is impractical, we have tried to concentrate system-dependent code into the main program and into the top

subroutines of the overlays. However, some system-dependent code is to be found almost anywhere. For example, MOVLEV, a CDC subroutine used in a few places for speed, has to be used where the speed is needed in subroutines DXTRAN, TALLYD, and BANKIT. Finally, coding practices forced on us by the limitations of certain systems, such as keeping all integer values within 32 bits, affect the entire program.

Some features of MCNP cannot be provided within the FORTRAN language. They are implemented by calling subroutines in local system libraries. These system-dependent features are not all available in all systems. The chart below shows the present availability. The geometry-plotting feature is a special case. Its availability depends more on the local availability of one of the three plotting packages, CGS, PLOT10, or DISSPLA, than on the nature of the computer system. DISSPLA has the added complication of not having a regular initialization procedure. The local installation has to insert its particular DISSPLA initialization call in MCNP subroutine DISSF.

System-Dependent Features

	CTSS	VMS	COS	NOS	IBM	PRIME
execute-line message	1	1	1	1		
interact with terminal	1	1	0	1		
terminal interrupt	1	1	0			
time interrupt	1	1				
control of amount of memory	1			?		
memory access pattern analysis	1					
fetch cross section files	1					
get date and time	1	1	1	1		
get computer time used	1	1	1	1		
get computer time remaining	1					
get setting of sense-switch 1	1		1	1		
get load date	1		1			
get and set charge code	1					
get production flag	1			1		
preserve the drop file	1					
predictable drop-file name	1					
inhibit warnings from library routines	1					
read ACE-format files	1	x	x	x	x	x
geometry plotting	1	1		1	1	
send graphics files to devices	1					

1 = available in MCNP
x = no reason to make it available
0 = impossible
? = available but may not work right
blank = we don't know yet how to provide the feature

V. *OVERLAYS*

On most systems MCNP is run without overlays, either because the memory is very large or because a virtual-memory operating system is used. However, on some systems overlays are necessary, so MCNP is set up as an overlaid program.

Most MCNP runs consist of problem setup followed by either geometry plotting or transport. This simple pattern is reflected in the first level of overlays, the top subroutines of which are

IMCN	problem setup
PLOT	geometry plotting
XACT	cross-section table processing
MCRUN	transport and output

XACT would logically be under IMCN, but since it requires a large memory space, is sometimes time consuming, and is not needed for geometry plotting, it is a separate overlay. The first level overlays are called by the main program of MCNP either directly or through subroutine OVERLAY, depending on the particular operating system.

A second level of overlays is provided in IMCN, PLOT, and MCRUN. They are called through subroutine SECOVL, which calls the top subroutine of a specified second level overlay either directly or through subroutine OVERLAY. In IMCN and PLOT each second level overlay is called at most once. In MCRUN one second level overlay does particle transport and is in memory most of the time. From time to time, the other second level overlay comes in to produce output.

VI. *COMMON BLOCKS*

Most of the common storage is in comdeck CM, which is used by all MCNP program units except some short mathematical or system-oriented subprograms. This common storage is divided into nine separate common blocks. Dynamically allocated storage is in common block /DAC/, separate from statically allocated storage. Fixed, variable, and ephemeral data are separated to simplify maintenance of subroutine TPEFIL, which writes and reads the RUNTPE file. Fixed data are defined in setting up the problem, are written to RUNTPE only once, and are not changed during transport. Variable data are changed during transport and have to be written to RUNTPE for each restart dump. Ephemeral data are needed only during problem setup or only during the current history and are not written to RUNTPE. Variable data that are backed up when a lost-particle history is rerun are separated from variable data that are not backed up. The particle description variables that have to be saved when a detector tally is made, when a DXTRAN particle is generated, and when a particle is banked are in common

block /PBLCOM/, which is separate from the rest of the ephemeral data. Character data are in a common block separate from the numerical data in accordance with the rules of FORTRAN. Tables of hard-wired data are in a separate block called /TABLES/. For speed, common block /PB9COM/, which is where the data in /PBLCOM/ are saved, is in LCM on CDC machines that have LCM. Dynamically allocated common block /DAC/ is in LCM in order to get enough space.

If any of the following common blocks is changed, the marker variable at the end of the block must remain at the end. The length parameter associated with the block may need to be changed. The value of the length parameter is the number of numeric storage units in the common block.

common block	marker variable	length parameter
/FIXCOM/	MFIXCM	LFIXCM
/VARCOM/	MVARCM	LVARCM
/NBVCOM/	MNBVCM	LNBVCM
/EPHCOM/	MEPHCM	LEPHCM
/PBLCOM/	MPBLCM	LPBLCM

The expression for each length parameter includes the parameter NDP2, which is the number of numeric storage units needed for a floating-point quantity. It has the value 1 on 60-bit and 64-bit machines and 2 on 32-bit machines. If /PBLCOM/ is changed in any way, the common block parallel to it, /PB9COM/, must also be changed in the same way.

First-level overlays IMCN, PLOT, and MCRUN each have one or more common blocks that contain some constant tables and some ephemeral data used only in that overlay.

VII. DYNAMICALLY ALLOCATED STORAGE

MCNP uses a limited form of dynamically allocated storage. The lengths and locations of all dynamically allocated arrays are defined during problem setup and are not changed during transport and output. All dynamically allocated storage, for both real (double-precision on 32-bit machines) and integer arrays, is in common block /DAC/. /DAC/ contains only one declared array, DAS. All of the dynamically allocated arrays are equivalenced to DAS. When any dynamically allocated array is referenced, an offset is included in the first subscript expression. The offset for each array is equal to the offset of the previous array plus the length of the previous array. Most of the arrays are included in three sets of arrays, one each for fixed, variable, and ephemeral data. The arrays used for tallying, TAL, and for nuclear data tables, XSS, follow at the end. The space TAL and XSS occupy is also used for some temporary arrays during

problem setup and geometry plotting. The lengths of most of the arrays are determined during the course of a preliminary reading of the INP file by subroutine PASS1. The offsets of those arrays are calculated in subroutine SETDAS. The INP file is then rewound and is read again by subroutine RDPROB. This time the data from INP are actually stored. The length of TAL is calculated in subroutine ITALLY. The length of XSS is calculated in subroutines under XACT.

The parameter NDP2 is used to make the appropriate adjustments to the offsets where an integer array follows a floating-point array or vice versa.

On systems that provide dynamic memory size adjustment, DAS is dimensioned relatively small, and /DAC/ is loaded as the last thing in memory. At several points during the problem setup, the memory size is adjusted to make /DAC/ big enough to hold the arrays whose lengths have been defined. On systems without dynamic memory size adjustment, mostly virtual-memory systems, the parameter MDAS, which is the length of DAS, has to be set before compilation to be large enough for the biggest problem planned to be run but not so large as to violate whatever technical or administrative constraints may exist at the site.

VIII. THE RUNTPE FILE

The RUNTPE file contains all the information needed to restart a problem in the continue-run mode. The RUNTPE file can also be read by postprocessing programs to pick up information about the problem. The MCPLOT postprocessor does just that.

The RUNTPE file is sequential and unformatted. It is written and read by subroutine TPEFIL. The first part of RUNTPE is a sequence of records containing fixed data for the problem. The rest of RUNTPE is a sequence of restart dumps, each consisting of a sequence of records containing variable data. The first dump is written immediately after the records of fixed data are written, before any transport calculations are done. Subsequent dumps are written from time to time during the initial run and during any continue-runs. If a continue-run is done with execute message item C, its dumps are written after the dump from which it started. If a continue-run is done with execute message item CN, its dumps are written after the fixed-data records.

Records in the Fixed-Data Part of the RUNTPE File

Identification Record

KOD*8	name of the code
VER*5	version identification
LODDAT*8	load date of the code
IDTM*19	machine designator, date and time
CHCD*10	charge code
PROBID*19	problem identification
PROBS*19	problem identification of surface source
AID*80	problem title
UFIL(3,6)*11	characteristics of user files
XLIST(MAXE)*10	list of names of cross-section tables
MXE	number of cross-section tables in the problem

Cross-section tables, MXE of them, one per record.

The contents of /FIXCOM/.

The part of /DAC/ that contains fixed data.

Records in a Restart Dump

Dump Identification Record

Current values of KOD, VER, LODDAT, IDTM, CHCD, and PROBID.
PROBID is always the same as in the initial identification record.

The contents of /VARCOM/ and /NBVCOM/ in one record.

The part of /DAC/ that contains variable data.

The part of /DAC/ that contains tallies, if any.

Endfile record, which is overwritten by the next dump.

Appendix E

GLOBAL CONSTANTS, VARIABLES, AND ARRAYS

This appendix contains information for users who need to modify MCNP. The first section is a dictionary of the symbolic names of the global entities in MCNP. The second section contains descriptions of some complicated arrays.

I. DICTIONARY OF SYMBOLIC NAMES

The global variables and arrays in MCNP are declared in COMMON statements that are in comdecks to reduce the bulk of the code and to simplify maintenance. The comdecks are copied into the MCNP program units in a preprocessor run before compilation. Some comdecks also contain PARAMETER statements that declare global named constants. Associated with each comdeck that has any common blocks is a BLOCK DATA subprogram that provides initial definitions for some of the entities in the common blocks. The arrangement of the common blocks and named constants in the comdecks and their related BLOCK DATA subprograms is as follows.

COMDECK ZC Double precision declaration and named constants

COMDECK CM with BLKDAT Common blocks for all overlays
Includes comdeck ZC

/FIXCOM/	Fixed common
/VARCOM/	Variable common (Backed up for lost tracks)
/NVBCOM/	Variable common (Not backed up)
/EPHCOM/	Ephemeral common
/PBLCOM/	Particle and collision data
/PB8COM/	For saving PBLCOM
/PB9COM/	For saving PBLCOM
/TABLES/	Tables of constant data
/CHARCM/	Character variables and arrays
/DAC/	Dynamically allocated common

COMDECK JC with IBLDAT Common blocks for the IMCN overlay
Named constants

/IMCCOM/	Constants and ephemeral data
/JMCCOM/	Character variables and arrays
/DAC/	Dynamically allocated common

COMDECK PC with PBLDAT Common blocks for the PLOT overlay
Named constants
/PLTCOM/ Constants and ephemeral data
/DAC/ Dynamically allocated common

COMDECK RC with RBLDAT Common blocks for the MCRUN overlay
/RUNCOM/ Constants and ephemeral data
/DAC/ Dynamically allocated common

The symbolic names of the global constants, variables, and arrays are listed alphabetically below. The dimension bounds (for arrays), the location, and a brief description is given for each entry. The adjustable dimension bound of each dynamically allocated array is indicated by a *. The location of each variable or array is the name of its common block with the slashes omitted. The location of each entity in the dynamically allocated common block /DAC/ also includes the two-letter designator of a comdeck, because /DAC/ is declared in several different comdecks. The location of each global named constant is given as the comdeck designator followed by -par.

The names of the entities in /PB8COM/ and /PB9COM/, which are where the contents of /PBLCOM/ are saved in various places in MCNP, are not included in the dictionary. The names are in most cases the same as the corresponding names in /PBLCOM/ but with an 8 or 9 appended. Their meanings are the same as the meanings of the corresponding entities in /PBLCOM/.

AID*80	CHARCM	Title card of the initial run
AID1*80	CHARCM	Title card of the current run
AIDS*80	CHARCM	Title card of the surface source write run
AJSH	IMCCOM	Coefficient for surface area of a torus
AMFP	RUNCOM	Mean free paths to detector or DXTRAN sphere
AMX(4,4,*)	PC-DAC	Matrices of surface coefficients from SCF
ANG(5)	EPHCOM	Surface normal and cosine of track direction
ARA(*)	CM-DAC	Areas of the surfaces in the problem
ARAS(*)	JC-DAC	Calculated areas of the surfaces
ATSA(*)	JC-DAC	Calculated areas or volumes of tally segments
ATWT(MEMAX)	FIXCOM	Atomic weights
AVGDN	CM-par	Avogadro's number
AWT(*)	JC-DAC	Atomic weights from AWTAB card
BASIS(9)	PLTCOM	Basis vectors for plotting
BBB(4,4)	IMCCOM	Transformation matrix in volume calculator
BBV(*)	JC-DAC	Equiprobable bins of a source function
BCW(2,4)	FIXCOM	Coefficients of surface source biasing cylinder

APPENDIX E

COMMON

BFR(2048)	PC-DAC	CGS graphics buffer
BND(2)	PLTCOM	Extents for plotting
CBWF	RUNCOM	Weight multiplier for source direction bias
CENTER(2)	PLTCOM	Center of plot zoom
CHCD*10	CHARCM	Charge code
CMP(MXDT)	PBLCOM	Collision multiplicity for OMCFE detectors
CMULT	PBLCOM	Collision multiplicity
CNM(NKCD)*5	JMCCOM	List of all legal input-card names
COE(6,2,*)	PC-DAC	Parametric coefficients of plot curves
COLOUT(2,11)	PBLCOM	Energy and cosine of particles from collision
CP0	RUNCOM	Computer time used by MCNP to start of MCRUN
CP1	RUNCOM	Computer time used after beginning MCRUN
CP2	RUNCOM	Computer time used so far
CPA	EPHCOM	Computer time used up to start of MCNP
CPK	NBVCOM	Computer time for settling in a KCODE problem
CPV	RUNCOM	Current time for time interrupt in VMS
CRS(*)	PC-DAC	Intersections of plot curves
CTME	EPHCOM	Computer time cutoff from CTME card
CTS	NBVCOM	Computer time used for transport in current problem including previous runs, if any
DAS(MDAS/NDP2)	CM-DAC	Dynamically allocated storage
DBCN(20)	NBVCOM	Debug controls from DBCN card
DDET	RUNCOM	Distance from collision point to detector
DDG(MIPT,MXDT)	FIXCOM	Controls for detector diagnostics
DDM(2,*)	CM-DAC	Size and history of largest score of each tally
DDN(23,*)	CM-DAC	Detector diagnostics
DDX(MIPT,2,MXDX)	NBVCOM	Controls for DXTRAN diagnostics
DEC(2,*)	CM-DAC	Detector contributions by cell
DEN(*)	CM-DAC	Mass densities of the cells
DFTINT	CM-par	Default interval between time interrupts
DLS	PBLCOM	Distance to next boundary
DMP	NBVCOM	Dump control from PRDMP card
DRC(17,2,*)	CM-DAC	Data saved for coincident detectors
DTC	PBLCOM	Distance to time cutoff
DTI(100)	RUNCOM	Positive distances to surfaces
DUMN1*8	CHARCM	Dummy name for user-specified file
DUMN2*8	CHARCM	Dummy name for user-specified file
DXC(2,*)	CM-DAC	DXTRAN contributions by cell
DXCP(MIPT,*)	CM-DAC	DXTRAN cell probabilities
DXD(MIPT,23,MXDX)	CM-DAC	DXTRAN diagnostics
DXL	PBLCOM	Distance to nearest DXTRAN sphere
DXW(MIPT,3)	FIXCOM	DXTRAN weight cutoffs
DXX(MIPT,5,MXDX)	FIXCOM	DXTRAN sphere parameters
EAA(*)	CM-DAC	Average values of source distributions

EBL(16)	TABLES	Energy bins for photon accounting See page 484
ECF(MIPT)	FIXCOM	Neutron and photon energy cutoffs
ECSS(*)	ECS1	Cross section tables in ECS
EGO	PBLCOM	Energy of the particle before last collision
EMCF(MIPT)	FIXCOM	Cutin energy for analog capture
EMX	FIXCOM	Maximum energy in problem from ERGN card
ERG	PBLCOM	Particle energy
ERGACE	PBLCOM	Raw energy extracted from cross-section table
ERGSAB(0:MAXE)	FIXCOM	Cutin energy for $S(\alpha,\beta)$ treatment
ESPL(MIPT,10)	FIXCOM	Controls for energy splitting
EWWG(*)	CM-DAC	Energy bins for weight-window generator
EXMS*80	CHARCM	Execute message
EXSAV(2)	PLTCOM	Saved extents
EXTENT(2)	PLTCOM	Extents for plotting
FACTOR	PLTCOM	Zoom factor for plotting
FDD(2,*)	CM-DAC	Inhibitors of source frequency duplication
FEBL(2,16)	VARCOM	Neutron-induced photons, binned See page 484
FES(33)	IMCCOM	Fission energy spectrum for KCODE source
FIM(MIPT,*)	CM-DAC	Neutron and photon cell importances
FIML	RUNCOM	Importance of the current cell
FIS	NBVCOM	Importance of the source cell
FIVE19	CM-par	Random number multiplier and default seed
FME(MEMAX)	FIXCOM	Atom fractions from M cards
FOR(MIPT,*)	CM-DAC	Controls for forced collisions
FPI	RUNCOM	Reciprocal of number of histories
FRC(*)	CM-DAC	Fraction of source cut off by energy limits
FSO(*)	CM-DAC	Fission source for KCODE
FTLS	CM-par	Criterion for list vs table scoring
GELEC	CM-par	Mass of the electron in Mev
GENR(*)	CM-DAC	Up to 6 records of particles entering each cell
GNEUT	CM-par	Mass of the neutron in Mev
GWT(*)	CM-DAC	Minimum gamma production weights
GWW(2,9,*)	CM-DAC	Terminated and tallied weights
HBLN(MAXV,2)*3	CHARCM	Names of SDEF and SSR source variables
HBLW(MAXW)*3	CHARCM	Names of SSW source variables
HCS(2)*7	CHARCM	"cell" and "surface"
HDR(MAXE)*10	JMCCOM	Nuclides for discrete reaction treatment
HFU(2)*11	CHARCM	Legal values of file attribute FORM
HITM*67	JMCCOM	Current item from input card
HLBL(4)	PLTCOM	Controls for labels on plots
HLIN*80	JMCCOM	Initial storage for newly read input line
HMM(MEMAX)*10	CHARCM	Cross section table identifiers from M cards
HMT(MXMTX)*10	CHARCM	Cross section table identifiers from MT cards
HNP(MIPT)*7	CHARCM	"neutron" and "photon"

HSD(2)*10	CHARCM	Legal values of file attribute ACCESS
HUGE	CM-par	A very large number
IAAFD(*)	CM-DAC	Array name for the fixed data part of /DAC/
IAAVD(*)	CM DAC	Array name for the variable data part of /DAC/
IAP	RUNCOM	Program number of the next cell
IAX	RUNCOM	Flag for presence of AXS vector
IBC	RUNCOM	Index of the tally cosine bin
IBE	RUNCOM	Index of the tally energy bin
IBNK(*)	CM-DAC	The bank
IBS	RUNCOM	Index of the tally segment bin
IBT	RUNCOM	Index of the tally time bin
IBU	RUNCOM	Index of the tally user bin
ICA	IMCCOM	Index of the type of the current input card
ICH*5	JMCCOM	Name in columns 1-5 of the current input card
ICHAN	EPHCOM	Terminal channel for TTY interupt on VMS
ICL	PBLCOM	Program number of the current cell
ICN	IMCCOM	Number in columns 1-5 of current input card
ICRN(3,*)	JC-DAC	Surfaces and label of each cell corner
ICS	EPHCOM	Flag for error on current input card
ICURS	PLTCOM	Cursor flag
ICURS1	PLTCOM	Flag for saving initial conditions for cursor
ICX	IMCCOM	Flag for asterisk on current input card
IDEFV(MAXV)	FIXCOM	Flags for presence of variable names on SDEF
IDET	RUNCOM	Index of the current detector
IDMP	EPHCOM	Number of the dump to start a continue run from
IDRC(MXDT)	FIXCOM	Links between master and slave detectors
IDTM*19	CHARCM	Machine designator and current date and time
IDTMS*19	CHARCM	IDTM of the surface source write run
IDUM(50)	VARCOM	Data from IDUM input card
IDV	PLTCOM	Designator of the plotting device
IDX	PBLCOM	Number of the current DXTRAN sphere
IET	RUNCOM	Index of the current $S(\alpha,\beta)$ table
IEX	PBLCOM	Index of the current cross section table
IFIP(MIPT)	IMCCOM	Flag for presence of IP card
IFL(*)	CM-DAC	Nodes at cell leavings, for tally flagging
IGWW	FIXCOM	Pointer to weight window generator tally bin
IINT(*)	JC-DAC	Surfaces crossed at the intersections
IITM	IMCCOM	Integer form of current item from input card
IKZ	FIXCOM	Number of KCODE cycles to skip before tallying
ILN	EPHCOM	Count of lines of input data
ILN1	EPHCOM	Saved count of lines of input data
IMD	RUNCOM	Indicator of monodirectional plane source
IME(2,MEMAX)	IMCCOM	Material numbers on cell and FM cards
IMT	FIXCOM	Number of times the surface source will occur

IMTX(MXMTX)	EPHCOM	Material numbers from MT cards
INDT	EPHCOM	Count of entries on MT cards
INIT	RUNCOM	Flag to advance starting random number
INK(MINK)	FIXCOM	Output controls from PRINT card
INP*8	CHARCM	Name of problem input file
INW	IMCCOM	Latch for other-side-cell-specification warning
IOID	IMCCOM	Flag for VOID card
IOVR	EPHCOM	Index of the current first level overlay
IPAC2(*)	CM-DAC	Flags used to distinguish between population and tracks entering cell
IPAN(*)	CM-DAC	Pointers into PAN for all the cells
IPL	IMCCOM	Pointer into RTP for current tally card
IPNT(2,21,0:*)	JC-DAC	Pointers into RTP See page 480
IPSC	PBLCOM	Type of PSC calculation to make
IPT	PBLCOM	Type of particle: 1=neutron, 2=photon
IPTAL(8,5,*)	CM-DAC	Guide to tally bins See page 475
IPTY	FIXCOM	Particle types to be written to surface source
IQC	PLTCOM	Index of current curve of current surface
IRC	IMCCOM	First column of data field of input line
IRS	IMCCOM	Index of the current source distribution
IRT	RUNCOM	Counter for renormalizing direction cosines
ISEF(2,*)	CM-DAC	Source position tries and rejections
ISIC(MAXF)	RUNCOM	Distribution used for each source variable
ISPACE	PBLCOM	Spacer to make the number of integers even
ISRP(7)	VARCOM	Processed source parameters from SRC card
ISS(*)	CM-DAC	Surfaces where input surface source is to start
ISSW	FIXCOM	Flag to cause surface source file to be written
IST	VARCOM	Where in FSO to store next KCODE source neutron
ISUB(NDEF)*8	CHARCM	Names of I/O files
ITAL	EPHCOM	Index of the current tally
ITASK	PLTCOM	Plotting task
ITDS(*)	CM-DAC	Tally specifications See page 476
ITERM	EPHCOM	Type of computer terminal
ITFXS	EPHCOM	Flag to indicate need for total-fission tables
ITI(100)	RUNCOM	Surface numbers associated with DTI values
ITOTNU	EPHCOM	Flag for total vs prompt nubar
ITTY	CM-par	I/O unit for terminal keyboard
IU1	CM-par	I/O unit for a scratch file
IU2	CM-par	I/O unit for another scratch file
IUB	CM-par	I/O unit for bank backup file
IUD	CM-par	I/O unit for directory of cross section tables
IUI	CM-par	I/O unit for problem input file
IUO	CM-par	I/O unit for problem output file
IUP	CM-par	I/O unit for intermediate file of plots

IUR	CM-par	I/O unit for file of restart dumps
IUS	CM-par	I/O unit for KCODE source file
IUSC	CM-par	I/O unit for surface source scratch file
IUSR	CM-par	I/O unit for surface source input file
IUSW	CM-par	I/O unit for surface source output file
IUX	CM-par	I/O unit for files of cross section tables
IVBU(*)	CM-DAC	Saved variable data for lost-particle reruns
IVC	RUNCOM	Flag for presence of VEC vector
IVDD(MAXF)	FIXCOM	For each dependent source variable, the number of the source variable depended upon
IVDIS(MAXV)	FIXCOM	Distribution number for each source variable
IVORD(MAXF)	FIXCOM	Source variable numbers in sampling order
IXAK	VARCOM	Where in FSO to get next KCODE source neutron
IXCOS(11)	PBLCOM	Pointer to cosine table for PSC calculation
IXRE	RUNCOM	Index of the collision reaction
JAP	RUNCOM	Program number of the next surface
JASR(*)	CM-DAC	Input surface source surfaces to be used
JASW(*)	JC-DAC	Surfaces from surface source input file
JBD	RUNCOM	Indicator for scoring flagged (or direct) bin
JDSP	PLTCOM	Flag for sending graphics files to devices
JEPHCM	EPHCOM	Array name for the whole common block
JESP	PLTCOM	Type of paper to use on electrostatic plotter
JEV	RUNCOM	Count of event-log lines printed
JFCN	EPHCOM	Flag indicating CN is in the execute message
JFIXCM	FIXCOM	Array name for the whole common block
JFL(*)	CM-DAC	Nodes of surface crossings, for tally flagging
JFQ(8,0:*)	CM-DAC	Order for printing tally results
JFR80	PLTCOM	Type of film to use for plotting
JNBVCM	NBVCOM	Array name for the whole common block
JOVR(NOVR)	EPHCOM	Flags for first level overlays to be executed
JPBLCM	PBLCOM	Array name for the common block See page 472
JPTAL(8,*)	CM-DAC	Basic tally information See page 475
JRWB(13)	RUNCOM	PWB columns corresponding to values of NTER
JSCN(*)	CM-DAC	Source comments
JSD(4,33)	IMCCOM	Flags for distributions that need space in SSO
JSF(MJSF)	TABLES	Numerical names of built-in source functions
JSS(*)	CM DAC	Surfaces for surface source output file
JST(2,*)	PC-DAC	Stack of points in the current piece of cell
JSU	PBLCOM	Program number of the current surface
JTF(8,*)	CM-DAC	Indices for fluctuation charts See page 479
JTLS	RUNCOM	Count of the scores in the current history
JTLX	FIXCOM	Latch for the TALLYX warning message
JTR(*)	CM-DAC	Transformation numbers from surface cards
JTTY	CM-par	I/O unit for terminal printer or CRT

JUI	IMCCOM	Unit number of the current input file
JVARCM	VARCOM	Array name for the whole common block
JVC(*)	CM-DAC	Vector numbers from the VECT card
JXS(32,MAXE)	FIXCOM	Blocks of pointers into cross section tables
KAW(*)	JC-DAC	Values of Z*1000+A from the AWTAB card
KCL(102,*)	PC-DAC	Cell numbers of grid points in the plot window
KCSF	VARCOM	Flag for KCODE source overlap
KCT	NBVCOM	Number of KCODE cycles to run
KCY	NBVCOM	Current KCODE cycle
KDB	RUNCOM	Flag for lost particle or long history
KDUP(*)	JC-DAC	List of input cards for detecting duplicates
KFL	FIXCOM	Flag for cell or surface tally flagging
KFM(*)	PC-DAC	Type of curve each surface makes in plot plane
KITM	IMCCOM	Type of current item from input card
KLIN*80	CHARCM	Input line currently being processed
KNOD	NBVCOM	Dump number
KNODS	FIXCOM	Last dump in the surface source write run
KNRM	FIXCOM	Type of normalization of KCODE tallies
KOD*8	CM-par	Name of the code (MCNP)
KODS*8	CHARCM	Name of the code that wrote surface source file
KONRUN	EPHCOM	Continue-run flag
KOV	EPHCOM	Flag to print lost-particle information
KOVR(4,2)*5	CHARCM	Names of the first level overlays
KPROD	EPHCOM	Flag for production status
KRFLG	RUNCOM	Flag to do event printing
KRN	PBLCOM	Current random number count for banked particle
KRQ(7,NKCD)	IMCCOM	Attributes of all types of input data cards
KSD(21,*)	CM-DAC	Source distribution information See page 470
KSF(29)*3	CHARCM	List of all legal surface-type symbols
KSR	RUNCOM	Number of seconds left before job time limit
KST(*)	CM-DAC	Surface-type numbers of all the surfaces
KTCR(2,MAXE)	RUNCOM	Current indices of energy grids See page 472
KTL(NTALMX,2)	IMCCOM	Amount of storage needed for segment divisors
KTLS	NBVCOM	Flag for list vs table scoring
KUFIL(2,6)	FIXCOM	Unit numbers and record lengths of user files
KXS(MAXE)	FIXCOM	Indices of the cross section tables on RUNTPE
KZKF	NBVCOM	Flag for tallying in KCODE problems
LAJ(*)	CM-DAC	Cells on the other sides of the surfaces in LJA
LAMX	PLTCOM	Offset for AMX array
LARA	FIXCOM	Offset for ARA array
LARS	IMCCOM	Offset for ARAS array
LAT	IMCCOM	Total storage needed for tally segment divisors
LATS	IMCCOM	Offset for ATSA array
LAWFLG(4)	NBVCOM	Latches for obsolete-law warning messages

LAWT	IMCCOM	Offset for AWT array
LBB	RUNCOM	Particles per record in bank backup file
LBBV	IMCCOM	Offset for BBV array
LBFR	PLTCOM	Offset for BFR array
LBNK	FIXCOM	Offset for IBNK array
LCA(*)	CM-DAC	For each cell, a pointer into LJA and LCAJ
LCAJ(*)	CM-DAC	For each surface in LJA, a pointer into the list of other-side cells in LJA
LCL(*)	PC-DAC	List of cells bounded by the current surface
LCOE	PLTCOM	Offset for COE array
LCRS	PLTCOM	Offset for CRS array
LDDN	FIXCOM	Offset for DDN array
LDEC	FIXCOM	Offset for DEC array
LDEN	FIXCOM	Offset for DEN array
LDQ	EPHCOM	Flag to preserve or destroy the drop file
LDR	FIXCOM	Flag for presence of coincident detectors
LDRC	FIXCOM	Offset for DRC array
LDUP	IMCCOM	Offset for KDUP array
LDXC	FIXCOM	Offset for DXC array
LDXD	FIXCOM	Offset for DXD array
LDXP	FIXCOM	Offset for DXCP array
LEAA	FIXCOM	Offset for EAA array
LEPHCM	CM-par	Length of /EPHCOM/
LEWG(MIPT)	FIXCOM	Offsets for EWWG array
LFATL	EPHCOM	Flag to run in spite of fatal errors
LFCD	FIXCOM	Length of fixed dynamically allocated storage
LFCL(*)	CM-DAC	Cells where fission is treated like capture
LFDD	FIXCOM	Offset for FDD array
LFIM	FIXCOM	Offset for FIM array
LFIXCM	CM-par	Length of /FIXCOM/
LFLL	EPHCOM	Current length of /DAC/
LFOR	FIXCOM	Offset for FOR array
LFRC	FIXCOM	Offset for FRC array
LFSO	FIXCOM	Offset for FSO array
LGC(101)	EPHCOM	Logical expression for the current point with respect to a particular cell
LGNR	FIXCOM	Offset for GENR array
LGWT	FIXCOM	Offset for GWT array
LGWW(MIPT)	FIXCOM	Offsets for GWW array
LICC	IMCCOM	Length of /DAC/ during execution of IMCN
LICR	IMCCOM	Offset for ICRN array
LIFL	FIXCOM	Offset for IFL array
LIIN	IMCCOM	Offset for IINT array
LIPA	FIXCOM	Offset for IPAN array

LIPN	IMCCOM	Offset for IPNT array
LIPT	FIXCOM	Offset for IPTAL array
LISE	FIXCOM	Offset for ISEF array
LISS	FIXCOM	Offset for ISS array
LIT	IMCCOM	Length of ITDS array
LITD	FIXCOM	Offset for ITDS array
LIVB	FIXCOM	Offset for IVBU array
LJA(*)	CM-DAC	Logical geometrical definitions of all cells
LJAR	FIXCOM	Offset for JASR array
LJAV(*)	JC-DAC	Logical geometrical definition of current cell
LJAW	FIXCOM	Offset for JASW array
LJFL	FIXCOM	Offset for JFL array
LJFQ	FIXCOM	Offset for JFQ array
LJPT	FIXCOM	Offset for JPTAL array
LJSC	FIXCOM	Offset for JSCN array
LJSS	FIXCOM	Offset for JSS array
LJST	PLTCOM	Offset for JST array
LJSV(*)	JC-DAC	List of the surfaces of the current cell
LJTF	FIXCOM	Offset for JTF array
LJTR	FIXCOM	Offset for JTR array
LJVC	FIXCOM	Offset for JVC array
LKAW	IMCCOM	Offset for KAW array
LKCL	PLTCOM	Offset for KCL array
LKFM	PLTCOM	Offset for KFM array
LKSD	FIXCOM	Offset for KSD array
LKST	FIXCOM	Offset for KST array
LLAJ	FIXCOM	Offset for LAJ array
LLAV	IMCCOM	Offset for LJAV array
LLCA	FIXCOM	Offset for LCA array
LLCJ	FIXCOM	Offset for LCAJ array
LLCL	PLTCOM	Offset for LCL array
LLCT	FIXCOM	Offset for LOCCT array
LLFC	FIXCOM	Offset for LFCL array
LLJA	FIXCOM	Offset for LJA array
LLSA	IMCCOM	Offset for LSAT array
LLSC	FIXCOM	Offset for LSC array
LLSE	FIXCOM	Offset for LSE array
LLSG	PLTCOM	Offset for LSG array
LLST	FIXCOM	Offset for LOCST array
LLSV	IMCCOM	Offset for LJSV array
LMAT	FIXCOM	Offset for MAT array
LME(MIPT,MEMAX)	FIXCOM	For each material, a list of the indices of the cross section tables for neutrons and photons
LMLL	FIXCOM	Offset for MLL array

LMT(MEMAX)	FIXCOM	For each material, a list of the indices of the applicable $S(\alpha,\beta)$ tables
LNBVCM	CM-par	Length of /NBVCOM/
LNCL	FIXCOM	Offset for NCL array
LNCS	PLTCOM	Offset for NCS array
LNP	FIXCOM	Location of NPA in JPBLCM
LNPW	FIXCOM	Offset for NPSW array
LNSF	FIXCOM	Offset for NSF array
LNSL	FIXCOM	Offset for NSL array
LNSR	FIXCOM	Offset for NSLR array
LNTB	FIXCOM	Offset for NTBB array
LOCCT(MIPT,*)	CM-DAC	Cell-tally locators See page 477
LOCDT(2,MXDT)	FIXCOM	Detector-tally locators See page 476
LOCST(MIPT,*)	CM-DAC	Surface-tally locators See page 477
LODDAT*8	CHARCM	Date when the code was loaded
LODS*8	CHARCM	LODDAT of code that wrote surface source file
LOST(2)	NBVCOM	Controls for handling lost particles
LOVR	EPHCOM	Current second level overlay
LPAC	FIXCOM	Offset for PAC array
LPAN	FIXCOM	Offset for PAN array
LPB8CM	CM-par	Length of /PB8COM/
LPB9CM	CM-par	Length of /PB9COM/
LPBB	FIXCOM	Length of JPBLCM for banking
LPBD	FIXCOM	Length of JPBLCM to save in TALLYD
LPBLCM	CM-par	Length of /PBLCOM/
LPBX	FIXCOM	Length of JPBLCM to save in DXTRAN
LPC2	FIXCOM	Offset for IPAC2 array
LPCC	FIXCOM	Offset for PCC array
LPLB	PLTCOM	Offset for PLB array
LPRB	IMCCOM	Offset for PRB array
LPWB	FIXCOM	Offset for PWB array
LQAX	FIXCOM	Offset for QAX array
LQMX	PLTCOM	Offset for QMX array
LRHO	FIXCOM	Offset for RHO array
LRKP	FIXCOM	Offset for RKPL array
LRSC	IMCCOM	Offset for RSCRN array
LRSN	IMCCOM	Offset for RSINT array
LRT	IMCCOM	Length of RTP array
LRTP	IMCCOM	Offset for RTP array
LSAT(*)	JC-DAC	For each segmented tally, a pointer into ATSA
LSB	RUNCOM	Latch for the count of bank overflows
LSC(*)	CM-DAC	For each surface, a pointer into SCF
LSCF	FIXCOM	Offset for SCF array
LSCQ	IMCCOM	Offset for SCFQ array

LSE(*)	CM-DAC	Cells where source particles have appeared
LSFB	IMCCOM	Offset for SFB array
LSG(*)	PC-DAC	Kind of line to plot for each segment of curve
LSPEED	PLTCOM	Baud rate of the plotting terminal display
LSPF	FIXCOM	Offset for SPF array
LSQQ	FIXCOM	Offset for SQQ array
LSSO	FIXCOM	Offset for SSO array
LTAL	FIXCOM	Offset for TAL array
LTD	FIXCOM	Length of TDS array
LTDS	FIXCOM	Offset for TDS array
LTFC	FIXCOM	Offset for TFC array
LTMP	FIXCOM	Offset for TMP array
LTTH	FIXCOM	Offset for TTH array
LVARCM	CM-par	Length of /VARCOM/
LVCD	FIXCOM	Length of the fixed and variable parts of /DAC/
LVEC	FIXCOM	Offset for VEC array
LVLS	IMCCOM	Offset for VOLS array
LVOL	FIXCOM	Offset for VOL array
LWNS	FIXCOM	Offset for WNS array
LWWE(MIPT)	FIXCOM	Offsets for WWE array
LWWF(MIPT)	FIXCOM	Offsets for WWF array
LXS	FIXCOM	Length of XSS array
LXSS	FIXCOM	Offset for XSS array
LZST	PLTCOM	Offset for ZST array
M1C	IMCCOM	General purpose variable for PASS1 and RDPROB
M2C	IMCCOM	General purpose variable for PASS1 and RDPROB
M3C	IMCCOM	General purpose variable for PASS1 and RDPROB
M4C	IMCCOM	General purpose variable for PASS1 and RDPROB
MAT(*)	CM-DAC	Material numbers of the cells
MXDX	CM-par	Maximum number of DXTRAN spheres
MAXE	CM-par	Maximum number of cross section tables
MAXF	CM-par	Number of sampleable source variables
MAXV	CM-par	Number of SDEF source variables
MAXW	CM-par	Number of SSW source variables
MBB	RUNCOM	Number of bank particles actually in memory
MDAS	CM-par	Initial length of /DAC/
MDC	RUNCOM	Flag indicating a dump is due to be written
MEMAX	CM-par	Size of tables for M-card data
MEPHCM	EPHCOM	Marker variable at end of /EPHCOM/
MFIXCM	FIXCOM	Marker variable at end of /FIXCOM/
MINK	CM-par	Length of INK array
MIPT	CM-par	Number of kinds of particles the code can run
MIX	EPHCOM	Number of entries in HMM and FME
MJSF	CM-par	Length of JSF array

MLAJ	FIXCOM	Length of LAJ array
MLJA	FIXCOM	Length of LJA array
MLL(2,*)	CM-DAC	Pointers to the portions of HMM, FME, LME, and LMT for the material in each cell
MNBVCM	NBVCOM	Marker variable at end of /NBVCOM/
MNK	EPHCOM	Flag to indicate maximum printing is wanted
MODE	FIXCOM	Problem mode from MODE card
MPBLCM	PBLCOM	Marker variable at end of /PBLCOM/
MPC	RUNCOM	Flag indicating that printing is due to be done
MRL	FIXCOM	Number of source points in FSO
MSD	FIXCOM	Number of source distributions
MSEB	CM-par	Maximum number of equiprobable source bins
MSRK	FIXCOM	Maximum number of source points in FSO
MSSC	IMCCOM	Length of source comments array JSCN
MSUB(NDEF)*10	CHARCM	Default names of files
MVARCM	VARCOM	Marker variable at end of /VARCOM/
MXA	FIXCOM	Number of cells in the problem
MXAFM	FIXCOM	Number of cells plus pseudocells for FM cards
MXAFS	FIXCOM	Number of cells plus pseudocells for FS cards
MXC	CM-par	Length of each element of XSCRD
MXDT	CM-par	Maximum number of detectors
MXE	FIXCOM	Number of cross section tables in the problem
MXF	FIXCOM	Total number of tally bins, possibly adjusted upward to avoid LCM bank conflicts
MXF0	FIXCOM	Total number of bins for all tallies
MXF2	FIXCOM	2*MXF
MXF3	FIXCOM	3*MXF
MXIT	IMCCOM	Longest input geometry definition for any cell
MXJ	FIXCOM	Number of surfaces in the problem
MXMTX	CM-par	Size of tables for MT-card data
MXT	FIXCOM	Number of cell-temperature time bins
MXTR	CM-par	Maximum number of surface transformations
MXXS	IMCCOM	Length of SPF and WNS
NAW	IMCCOM	Number of atomic weights from AWTAB card
NBHWM	VARCOM	Largest number of particles ever in the bank
NBMX	CM-par	Number of particles IBNK has room for
NBNK	EPHCOM	Number of particles in the bank
NBOV	NBVCOM	Count of bank overflows
NBT(MIPT)	VARCOM	Total number of neutrons and photons banked
NBY	VARCOM	Flag for tallies beyond last bin
NCEL	PLTCOM	Number of cells bounded by the current surface
NCH(MIPT)	EPHCOM	Counts of neutron and photon collisions
NCL(*)	CM-DAC	Problem numbers of the cells
NCOMP	IMCCOM	Count of # characters on cell cards

NCP	PBLCOM	Count of collisions per track
NCPAR(MIPT,NKCD)	IMCCOM	Largest cell parameter n, -1 if none
NCPARF	IMCCOM	Number of cell parameter cards on cell cards
NCRN	IMCCOM	Number of corners in the current cell
NCRS	PLTCOM	Length of LSG and CRS arrays
NCS(*)	PC-DAC	Number of curves where surface meets plot plane
NCT(MIPT)	VARCOM	Total numbers of neutron and photon collisions
NDE	EPHCOM	Value of execute-message item DBUG n
NDEF	CM-par	Number of file names
NDET(MIPT)	FIXCOM	Numbers of neutron and photon detectors
NDP2	CM-par	Number of numeric storage units needed to store a floating-point value
NDRR(MAXE)	VARCOM	List of discrete-reaction rejections
NDTT	FIXCOM	Total number of detectors in the problem
NDUP(3)	IMCCOM	Number of cards in each input data block
NDX(MIPT)	FIXCOM	Numbers of neutron and photon DXTRAN spheres
NERR	NBVCOM	Count of lost particles
NETB(2)	VARCOM	Counts of numbers of times energy > EMX
NFER	NBVCOM	Count of fatal errors found by IMCN or XACT
NFRAME	PLTCOM	Number of pictures in the PIX file
NGWW(MIPT)	FIXCOM	Number of weight-window-generator energy bins
NII	IMCCOM	Number of interpolated values to make; -1 for J
NISS	FIXCOM	Number of histories in input surface source
NITM	IMCCOM	Length of current item from input card
NJSR	FIXCOM	Number of surfaces in JASR
NJSS	FIXCOM	Number of surfaces in JSS
NJSW	IMCCOM	Number of surfaces in JASW
NJSX	IMCCOM	Number of surfaces in ISS
NKCD	JC-par	Number of different types of input cards
NKXS	FIXCOM	Count of cross section tables written on RUNTPE
NLAJ	NBVCOM	Number of other-side cells in LAJ
NLB	PLTCOM	Number of surface labels on the plot
NLJA	FIXCOM	Number of entries in LJA
NLSE	NBVCOM	Number of cells in the LSE list
NLT	RUNCOM	Number of entries in DTI
NMAT	IMCCOM	Number of materials in the problem
NNPOS	FIXCOM	Index of first position variable to be sampled
NODE	PBLCOM	Number of nodes in track from source to here
NORD	FIXCOM	Number of source variables to be sampled
NOVOL	IMCCOM	Flag to inhibit volume calculation
NOVR	CM-par	Number of primary overlays
NP1	FIXCOM	Number of histories in surface source write run
NPA	PBLCOM	Number of tracks in the same bank location
NPC(20)	NBVCOM	NPS for tally fluctuation charts See page 479

NPD	NBVCOM	NPS step in tally fluctuation chart
NPLB	PLTCOM	Length of PLB array
NPN	FIXCOM	Length of adjustable dimension of PAN
NPNM	NBVCOM	Count of times neutron-reaction MT not found
NPP	NBVCOM	Number of histories to run, from NPS card
NPPM	NBVCOM	Count of times photon-production MT not found
NPS	VARCOM	Count of source particles started
NPSR	NBVCOM	History number last read from surface source
NPSW(*)	CM-DAC	For each surface source surface, the last history in which a track crossed it
NQSS	VARCOM	Number of histories read from surface source
NQSW	VARCOM	Number of histories written to surface source
NQW	IMCCOM	Type of input card: neutron, photon or neither
NRC	RUNCOM	Count of restarts in the run
NRCD	FIXCOM	Number of values in a surface-source record
NRN	NBVCOM	Count of pseudorandom numbers generated
NRRS	VARCOM	Number of tracks read from surface source
NRSS	FIXCOM	Number of tracks on input surface source file
NRSW	VARCOM	Number of tracks written to surface source
NSA	VARCOM	Source particles yet to be done in this cycle
NSC	FIXCOM	Number of surface coefficients in SCF
NSF(*)	CM-DAC	Problem numbers of the surfaces
NSJV	IMCCOM	Length of cell definition in LJAV
NSKK	NBVCOM	Number of histories in first IKZ KCODE cycles
NSL(10,*)	CM-DAC	Summary information for surface source file
NSLR(10,*)	JC-DAC	Summary information from surface source file
NSPH	FIXCOM	Flag for spherical output surface source
NSR	FIXCOM	Source type
NSRC	IMCCOM	Number of entries on SRC card
NSRCK	FIXCOM	Nominal size of the KCODE source
NSS	VARCOM	Count of source points stored for the next cycle
NSSI(8)	VARCOM	Numbers of rejected surface source tracks
NST	EPHCOM	Reasons why the run is terminating
NSV	IMCCOM	Number of surfaces in LJSV
NTAL	FIXCOM	Number of tallies in the problem
NTALMX	JC-par	Maximum number of tallies
NTBB(5,*)	CM-DAC	Counts of scores beyond the last bin
NTC	NBVCOM	Control variable for time check
NTC1	NBVCOM	Second control variable for time check
NTEK	PLTCOM	Number of pictures drawn on display terminal
NTER	RUNCOM	Type of termination of the track
NTII	RUNCOM	Indicator of multiple time interrupts
NTL(0:NTALMX)	IMCCOM	Tally numbers from tally input cards
NTSS	VARCOM	Number of surface source tracks accepted

NTX	RUNCOM	Number of calls of TALLYX in user bins loop
NTY(MAXE)	FIXCOM	Type of each cross section table
NTYN	PBLCOM	Type of reaction in current collision
NVEC	FIXCOM	Number of vectors on VECT card
NVS(MAXV)	TABLES	Number of values for each source variable
NWC	IMCCOM	Count of items on current input card
NWER	NBVCOM	Count of warning messages printed
NWSB	VARCOM	Count of source weights below cutoff
NWSE	VARCOM	Count of source energies below cutoff
NWSG	VARCOM	Count of source weights above weight window
NWSL	VARCOM	Count of source weights below weight window
NWST	VARCOM	Count of source times greater than cutoff
NWW(MIPT)	FIXCOM	Number of weight-window energy bins
NWWS(MIPT,10)	VARCOM	Like NWSG and NWSL but binned
NXP	PLTCOM	Number of intersections in CRS
NXS(16,MAXE)	FIXCOM	Blocks of descriptors of cross section tables
NXSC	IMCCOM	Number of XSn cards
NZIP	NBVCOM	Detector scores lost to zero importance
NZIX	NBVCOM	DXTRANs lost to zero importance
NZIY(MXDX,MIPT)	NBVCOM	DXTRANs lost to zero importance
ONE	CM-par	Floating-point constant 1.
ORIGIN(3)	PLTCOM	Origin for plotting
ORSAV(3)	PLTCOM	Saved origin
OSUM(3)	NBVCOM	Sums used to calculate K(eff)
OSUM2(3,3)	NBVCOM	Sums used to calculate covariances of K(eff)
OUTP*8	CHARCM	Name of problem output file
PAC(MIPT,10,*)	CM-DAC	Activity in each cell See page 481
PAN(MIPT,6,*)	CM-DAC	Activity of each nuclide See page 483
PAX(MIPT,6,16)	VARCOM	Ledger of creation and loss See page 480
PBL	PBLCOM	Array name for the common block See page 472
PBS	FIXCOM	Angle for cylindrical window biasing
PCC(3,*)	CM-DAC	Neutron-induced photons, by cell See page 484
PIE	CM-par	π
PIX*8	CHARCM	Name of intermediate file of plots
PLB(*)	PC-DAC	Locations and widths of surface labels
PLE	RUNCOM	Macroscopic cross section of current cell
PLMX(4,4)	PLTCOM	Plot matrix
PMF	RUNCOM	Distance to next collision
PRB(*)	JC-DAC	Probabilities for equiprobable-bin iteration
PRN	NBVCOM	Print control from PRDMP card
PROBID*19	CHARCM	Problem identification
PROBS*19	CHARCM	PROBID of the surface source write run
PSC	RUNCOM	Probability density for scattering toward a detector or DXTRAN sphere

PWB(MIPT,15,*)	CM-DAC	Weight-balance tables See page 482
PXYZ(3)	PLTCOM	Data from plot keywords PX,PY,PZ
QAX(MIPT,*)	CM-DAC	Exponential transform parameters for each cell
QMX(3,3,2,*)	PC-DAC	Curves where surfaces intersect the plot plane
QPL	RUNCOM	Adjusted macroscopic cross section
RANB	EPHCOM	Upper part of pseudorandom number
RANI	NBVCOM	Upper part of RIJK
RANJ	NBVCOM	Lower part of RIJK
RANK	EPHCOM	Saved value of RANI
RANL	EPHCOM	Saved value of RANJ
RANS	EPHCOM	Lower part of pseudorandom number
RDUM(50)	VARCOM	Data from RDUM input card
RES	PC-par	Plot resolution
RFQ(10)*57	CHARCM	Partial formats for termination messages
RHO(*)	CM-DAC	Atom densities of the cells
RIJK	NBVCOM	Starting random number for the current history
RITM	IMCCOM	Real form of current item from input card
RKK	NBVCOM	Collision estimate of K(eff)
RKPL(5,*)	CM-DAC	K(eff) and removal lifetime, for plotting
RLT(2)	VARCOM	Sums of lifetimes in current KCODE cycle
ROM	RUNCOM	Radius of sphere around OMCFE detector
RSCRN(2,*)	JC-DAC	R and S coordinates of cell corners
RSINT(2,*)	JC-DAC	R and S coordinates of surface intersections
RSSA*8	CHARCM	Name of surface source input file
RSSP	FIXCOM	Radius of spherical surface source
RSUM(2)	NBVCOM	Sums used to calculate removal lifetime
RSUM2(2,2)	NBVCOM	Sums used to calculate covariance of lifetime
RTCR(10,MAXE)	RUNCOM	Current interpolated cross sections See page 472
RTP(*)	JC-DAC	Tally-card data See page 479
RUNTPE*8	CHARCM	Name of file of restart dumps
SCF(*)	CM-DAC	Surface coefficients for all surfaces
SCFQ(5,*)	JC-DAC	Q-form of surface coefficients
SDIR(9)	FIXCOM	Parameters for source directional biasing
SFB(*)	JC-DAC	Probabilities of the source input groups
SFF(3,MAXF)	RUNCOM	Current values of source variables
SLITE	CM-par	Speed of light
SMUL(3)	VARCOM	Tally of neutron multiplication
SNIT	FIXCOM	Surface source splitting or RR factor
SPARE1	PBLCOM	Spare banked variable for user modifications
SPARE2	PBLCOM	Spare banked variable for user modifications
SPARE3	PBLCOM	Spare banked variable for user modifications
SPF(4,*)	CM-DAC	Source probability distributions See page 471
SQQ(12,*)	CM-DAC	Coefficients of the built-in source functions
SRC(51)	FIXCOM	Source parameters from SRC card

SRCTP*8	CHARCM	Name of KCODE source file
SRP(19)	FIXCOM	Processed source parameters from SRC card
SRV(3,MAXV)	FIXCOM	Explicit or default values of source variables
SSB(10)	EPHCOM	Surface source input buffer
SSO(*)	CM-DAC	Equiprobable bins for source distributions
SSR	PBLCOM	Neutron speed relative to target nucleus
STLS	NBVCOM	Sum of JTLS**2 for all histories so far
SUMK(3)	VARCOM	Sums of fission weight in current KCODE cycle
SWTM	IMCCOM	Minimum weight of source particles
SWTX	IMCCOM	Minimum source weight for obsolete sources
TAL(*)	CM-DAC	Tally scores accumulation See page 474
TALB(8,2)	TABLES	Bins for detector and DXTRAN diagnostics
TBLTMP(MAXE)	FIXCOM	Temperatures of the cross section tables
TCO(MIPT)	FIXCOM	Neutron and photon time cutoffs
TDC	RUNCOM	Time of writing latest dump to RUNTPE
TDS(*)	CM-DAC	Tally specifications See page 477
TFC(3,20,*)	CM-DAC	Tally fluctuation charts See page 479
THETA	PLTCOM	Zoom angle for plotting
THGF(0:50)	FIXCOM	Table of the thermal cross section function
THSAV	PLTCOM	Saved zoom angle
TLC	RUNCOM	Time of writing latest problem summary to OUTP
TMAV(MIPT,3)	VARCOM	Tallies of time to termination
TME	PBLCOM	Time at the particle position
TMP(*)	CM-DAC	Temperatures of the cells
TOTM	RUNCOM	Total microscopic cross section
TPD(7)	PBLCOM	Thermal collision data for PSC calculation
TPP(20)	EPHCOM	General-purpose scratch storage
TRF(14,MXTR)	FIXCOM	Surface transformations from TR cards
TSC	RUNCOM	Time of writing latest source to SRCTP
TTH(*)	CM-DAC	Time bins for cell temperatures
TTN	RUNCOM	Temperature of the current cell
TWAC	VARCOM	Total weight accepted from surface source file
TWSS	VARCOM	Total weight read from surface source file
UFIL(3,6)*11	CHARCM	Name, access, and form of each user file
UOLD	PBLCOM	Reference direction cosine with X-axis
UUU	PBLCOM	Particle direction cosine with X-axis
VCO(55)	RUNCOM	Form factors for photon scattering
VEC(3,*)	CM-DAC	Vectors from the VECT card
VEL	PBLCOM	Speed of the particle
VER*5	CM-par	Code version identification
VERS*5	CHARCM	Version of code that wrote surface source file
VIC(21)	RUNCOM	Form factors for photon scattering
VOL(*)	CM-DAC	Volumes of the cells in the problem
VOLD	PBLCOM	Reference direction cosine with Y-axis

VOLS(2,*)	JC-DAC	Calculated volumes of the cells
VTR(3)	PBLCOM	Velocity of the target nucleus
VVV	PBLCOM	Particle direction cosine with Y-axis
WC1(MIPT)	FIXCOM	First weight cutoff
WC2(MIPT)	FIXCOM	Second weight cutoff
WCO(55)	RUNCOM	Form factors for photon scattering
WCS1(MIPT)	FIXCOM	First weight cutoff modified by SWTM
WCS2(MIPT)	FIXCOM	Second weight cutoff modified by SWTM
WGT	PBLCOM	Particle weight
WGTS(2)	NBVCOM	Range of actual source weights
WNS(2,*)	CM-DAC	Actual frequencies of source sampling
WOLD	PBLCOM	Reference direction cosine with Z-axis
WSSA*8	CHARCM	Name of surface source output file
WSSI(7)	VARCOM	Weights of rejected surface source tracks
WTO	NBVCOM	Weight of each KCODE source point
WWE(*)	CM-DAC	Weight-window energy bins
WWF(*)	CM-DAC	Lower weight bounds for weight window
WWG(7)	FIXCOM	Controls for the weight window generator
WWP(MIPT,5)	FIXCOM	Weight-window controls
WWW	PBLCOM	Particle direction cosine with Z-axis
XLIST(MAXE)*10	CHARCM	Identifiers of cross section tables
XSCRD(MAXE)*MXC	CHARCM	Directory information for cross section tables
XSDIR*8	CHARCM	Name of directory of cross section tables
XSS(*)	CM-DAC	Cross section tables
XXX	PBLCOM	X-coordinate of the particle position
YSS	RC-DAC	Cross section tables See page 446
YYY	PBLCOM	Y-coordinate of the particle position
ZERO	CM-par	Floating-point constant 0.
ZST(*)	PC-DAC	Data buffer for PIX file
ZZZ	PBLCOM	Z-coordinate of the particle position

II. SOME IMPORTANT COMPLICATED ARRAYS

A. Source Arrays

<u>KSD(21,MSD) Array</u> Information About Each Source Distribution

KSD(LKSD+J,K) contains information of type J about source probability distribution K, as listed below.

<u>J</u>
1 problem name of the distribution
2 index of built-in function, if any

3 length of comment in JSCN
4 number of value sets from SI or DS card
5 flag for discrete distribution: L, S, F, Q, or T option
6 flag for distribution of distributions: S or Q option
7 flag for dependent distribution: DS rather than SI
8 flag for DS Q
9 flag for DS T
10 flag for SP V
11 flag for SI F
12 index of the variable of the distribution
13 offset into SPF
14 offset into SSO
15 offset into JSCN
16 offset into WNS
17 number of equiprobable bins in each group, if any
18 flag for biased distribution: SB card present
19 flag for interpolated distribution: A option
20 number of values on SP and/or SB card
21 number of values per bin, including tag from Q or T option

SPF(4,MXXS) Array   Source Probability Distributions

Each source distribution that is not just an unbiased function has a section of SPF. If the distribution is a histogram, the four rows of SPF contain

row
1 values of the variable (triples for POS, AXS, or VEC)
2 cumulative probability of each bin, possibly biased
3 weight factor to compensate for the bias
4 not used

If the distribution is linearly interpolated, the four rows contain

row
1 values of the variable (never triple)
2 unbiased probability density
3 biased probability density, if any
4 cumulative probability for sampling which bin

The above definitions are for the final SPF table as used in MCRUN. In IMCN the cumulative probabilities start out as probability per bin and the distributions may not yet be normalized.

B. Transport Arrays

PBL(11) and JPBLCM(LPBLCM) Arrays   Particle and Collision Descriptors

PBL and JPBLCM are equivalenced to the /PBLCOM/ common block. PBL is used when it is convenient to refer in an indexed fashion to the coordinates and direction cosines that are at the beginning of /PBLCOM/. JPBLCM is used when it is necessary to copy all or a large portion of /PBLCOM/ to or from some other place. Part of /PBLCOM/ is copied to /PB9COM/ at the beginning of DXTRAN and TALLYD and is retrieved at the end. Banking a particle consists of copying part of /PBLCOM/ to the next block of space in IBNK, and getting a particle from the bank is the reverse. The number of variables that have to be copied depends on the purpose and on whether there are any OMCFE detectors in the problem. The variables in /PBLCOM/ are ordered such that the variables that are copied are always in an unbroken sequence. Three variables, LPBX, LPBD, and LPBB, contain the length of the portion of JPBLCM that is to be copied. LPBX is used in DXTRAN, LPBD is used in TALLYD, and LPBB is used in BANKIT. All three are defined in LBLOCS and may be redefined in NEWCD1 if there are any OMCFE detectors in the problem.

KTCR(2,MAXE) and RTCR(10,MAXE) Arrays   Interpolated Cross Sections

When interpolated values of cross sections are calculated at the current particle energy, they are stored in KTCR and RTCR for possible use later in the calculation of the details of the collision. The values stored in KTCR(I,J) and RTCR(I,J) are as follows:

For neutron cross sections, class C, D, or Y
EGO = neutron energy in laboratory frame
ERG = neutron energy in target-at-rest frame

KTCR

1 index in cross-section table for EGO
2 index in cross-section table for ERG

RTCR

1 table interpolation factor for EGO
2 table interpolation factor for ERG
3 absorption cross section for EGO
4 total cross section for EGO at temperature of table
5 total cross section for EGO at cell temperature
6 EGO
7 cell temperature
8 fission cross section

9
10 number of neutrons emitted by fission

For neutron $S(\alpha,\beta)$ cross sections, class T

KTCR

1 index in inelastic cross-section table
2 index in elastic cross-section table

RTCR

1 inelastic interpolation factor
2
3
4 elastic interpolation factor
5
6 neutron energy
7 inelastic cross section plus elastic cross section
8 inelastic cross section
9
10

For photon cross sections, class P

RTCR

1 incoherent scattering cross section
2 incoherent plus coherent scattering cross section
3 incoherent plus coherent plus photoelectric cross section
4 total cross section
5 photon heating number
6 photon energy
7
8
9
10

C. Tally Arrays

The tallying facilities in MCNP are very flexible. The places in the code where tally scoring is done are very heavily used. The arrays required for flexible and efficient tallying are numerous and complicated. The main tally arrays, grouped by function, are listed below. Arrays in parentheses are not discussed separately but are mentioned in the discussion of the preceding array.

Accumulation of scores: TAL
Controls: JPTAL, IPTAL, LOCDT, ITDS (LOCCT, LOCST), TDS

APPENDIX E
Arrays

Fluctuation charts: TFC (JTF, NPC)
Initiation: RTP (IPNT)

TAL(MXF3) Array   Tally Scores Accumulation

TAL is in dynamically allocated storage with offset LTAL. LTAL is usually not explicit in the subscript of TAL because the values of the various pointers into TAL include LTAL. TAL is divided into three blocks, each of length MXF. Unless list scoring is in effect (see below), tally scores made during the course of a history are added into tally bins in the first block. At the end of each history the scores in the first block are added into corresponding places in the second block, their squares are added into the third block, and the first block is zeroed. Whenever printed output is called for, the sums in the second block and the sums of squares in the third block are used to calculate and print the tally estimates and their estimated errors.

Each of the three blocks in TAL is divided into sections of various lengths, one for each tally in the problem. Each section is an eight-dimensional array of tally bins. The storage sequence is as if the section of TAL were an eight-dimensional FORTRAN array. The order of the eight dimensions, corresponding to a right-to-left reading of the dimensions of a FORTRAN array, the kind of bins each dimension represents, and the input cards that define them are as follows.

1	cell, surface, or detector bins	F
2	all vs flagged or all vs direct	CF, SF or F
3	user bins	FU
4	segment bins	FS
5	multiplier bins	FM
6	cosine bins	C
7	energy bins	E
8	time bins	T

The number of bins in each dimension is determined by rules set forth in the descriptions of the input cards in Chapter 3.

An alternative way of entering scores into the first block is automatically used if the number of scores per history is sufficiently small compared to the size of the block. Only the first of the three blocks in TAL is affected. The procedure is as follows. Index JTLS is incremented by 2, the score is entered at TAL(LTAL+JTLS-1), and the location where the score would otherwise have gone is entered at TAL(LTAL+JTLS). At the end of the history, scores with the same location are consolidated, the scores and their squares are added into the second

and third blocks, and JTLS is set to zero. This technique is called list scoring. The scoring described previously is called table scoring. The reason for using list scoring is speed. It is used in only a small minority of problems but can in some cases make a big difference in running time.

<u>JPTAL(8,NTAL) Array</u> Basic Tally Information

JPTAL(LJPT+J,K) contains integer information of type J about tally K. Each pointer in JPTAL includes the offset of the array pointed into.

<u>J</u>
1 problem number of the tally
2 tally type: 1, 2, 4, 5, 6, or 7
3 particle type: 1 for neutron, 2 for photon, 3 for both
4 1 for an asterisk on F card, 0 if none
5 offset in the first block in TAL of the section for tally K
6 location of the tally comment in ITDS
7 location in TAL of the tally fluctuation chart bin
8 1 for a point detector, 2 for a ring detector, 3 for a
   OMCFE detector, 0 if not a detector tally

<u>IPTAL(8,5,NTAL) Array</u> Guide to Tally Bins

IPTAL(LIPT+I,J,K) contains information of type J about the bins of type I of tally K. The eight bin types I are defined above under TAL. The information types J are listed below, subject to the exceptions noted. Each pointer in IPTAL includes the offset of the array pointed into.

<u>J</u>
1 offset in TDS or ITDS of specifications for the bins. If
   there is just one unbounded bin, the value is zero.
   Exceptions
   I=2: for cell or surface tally, location in ITDS of flagging cells
      for detector tally, the number of direct bins (0 or 1)
   I=4: program number of pseudocell for segmenting surfaces
2 offset in TDS of bin multipliers
   Exceptions
   I=1: no meaning
   I=2: cell or surface tally: location in ITDS of flagging surfaces
      detector tally: offset in TDS of cell contributions
   I=3: location in TDS of the dose function

I=4: offset in TDS of the table of segment divisors

3 number of bins, which is never less than one

4 number of bins including a total bin whether there actually is a total bin or not

Exceptions

I=1 and I=2 have no meaning.

5 coefficients for calculating the location of a bin, given the eight bin indices

LOCDT(2,MXDT) Array   Detector-Tally Locators

LOCDT(1,J) is the program number of the tally of which detector J is a part. LOCDT(2,J) is the offset in the first block of TAL of the seven-dimensional array where scores for detector J are made.

ITDS(LIT) Array   Tally Specifications

ITDS contains blocks, in no particular order and accessed only through pointers, that contain some of the specifications of the tallies of the problem. ITDS is in dynamically allocated storage with offset LITD. LITD is usually not explicit in the subscript of ITDS because the values of the various pointers into ITDS include LITD.

Tally Comment

The value of JPTAL(LJPT+6,K) is the location in ITDS of the comment for tally K. The first element of the comment is the number of additional elements in the comment. Each line of 67 characters is contained in 23 elements of ITDS, packed 3 characters per element. The packing uses the ICHAR function and a shift factor of 256. The characters are unpacked and processed by the CHAR function before being printed.

Flagging Cells and Surfaces

The values of IPTAL(LIPT+2,1,K) and the values of IPTAL(LIPT+2,2,K) are the locations in ITDS of lists of the program numbers of flagging cells and flagging surfaces, respectively, for tally K. The first item of each list is the number of cells or surfaces in the list.

Cell and Surface Bins

The value of IPTAL(LIPT+1,1,K) is the offset in ITDS of the description of the cell or surface bins of cell or surface tally K. The structure of the description is

$P_1 \quad P_2 \ldots P_N \quad n_1 \quad l_{11} \quad l_{21} \ldots l_{n1} \quad n_2 \quad l_{12} \quad l_{22} \ldots l_{n2} \ldots$

where

N = number of cell or surface bins in tally K
P_i = pointer to specifications for bin i
n_i = number of cells or surfaces in bin i
l_{ji} = program number of a cell or surface in bin i

Cell and Surface Tally Pointers

The value of LOCCT(I,J) if J is a cell – or LOCST(I,J) if J is a surface – is the location in ITDS of a table which locates the sections of TAL where tally scoring is done when a particle of type I passes through cell or surface J. The table is organized this way:

$N \quad T_1 \quad m_1 \quad L_{11} \quad L_{21} \ldots L_{m1} \ldots T_N \quad m_N \quad L_{1N} \quad L_{2N} \ldots L_{mN}$

where

N = number of tallies for particle type I which include cell or surface J
T_i = program number of a tally
m_i = number of bins that involve cell or surface J
L_{ji} = cell or surface bin number

TDS(LTD) Array   Tally Specifications

TDS contains blocks, in no particular order and accessed only through pointers, that contain some of the specifications of the tallies of the problem. TDS is in dynamically allocated storage with offset LTDS. LTDS is usually not explicit in the subscript of TDS because the values of the various pointers into TDS include LTDS.

Detector Bins

For detector tally K, the value of IPTAL(LIPT+1,1,K) is the offset in TDS of the description of the detector bins. The description contains the information from the F card, modified for faster use in TALLYD. Five elements of TDS are used for each detector:

	Point detector	OMCFE	Ring detector
1	X	X	a
2	Y	Y	r
3	Z	Z	1, 2, or 3 for x, y, or z
4	R	R	R
5	$\lvert 2\pi R^3/3 \rvert$	cell	$\lvert 2\pi R^3/3 \rvert$

APPENDIX E
Arrays

Cell Contributions

For detector tally K, the value of IPTAL(LIPT+2,2,K) is the offset in TDS of the table of cell contributions. The information in the table is exactly as it is on the PD card.

Simple Bins and Multipliers

The value of IPTAL(LIPT+I,1,K) for I = 3, 6, 7, or 8 is the offset in TDS of a table of bins for tally K. The information in the table is as it came from the corresponding input card except that any T or NT on the card does not appear in the table. The value of IPTAL(LIPT+I,2,K) for I = 6, 7, or 8 is the offset in TDS of a table of bin multipliers for tally K. The information in the table is exactly as it is on the input card.

Segment Bin Divisors

For cell or surface tally K, the value of IPTAL(LIPT+4,2,K) is the offset in TDS of the table of segment bin divisors. Except for a type 1 tally without any SD card, the table exists even if there is no FS card. The table is a two-dimensional array. One dimension is for cell or surface bins and the other is for the segment bins. The segment bin index changes faster. If segment bin divisors are not provided on an SD card, they are calculated or derived from VOL or AREA data, if possible, by MCNP according to the tally type:

tally type	2	4	6	7
divisor	area	volume	mass	mass

Multiplier Bins

The value of IPTAL(LIPT+5,2,K) is the offset in TDS of a table of the constant multipliers for the multiplier bins from the FM card of tally K. If there is anything more on the FM card than just a constant multiplier for each bin, the value of IPTAL(LIPT+5,1,K) is the offset in TDS of a table of bin descriptions:

$$N \quad P_1 \quad P_2 \ldots P_N \quad I_1 \quad n_1 \quad R_{11} \quad R_{21} \ldots R_{n1} \quad I_2 \quad n_2 \quad R_{12} \quad R_{22} \ldots R_{n2} \ldots$$

where

N = number of P's.

P_i = pointer to the description of a bin or attenuator. If the FM card has only a constant for some bin, then $P_i = 0$ for that bin. If the FM card has C m but nothing more for a bin, (which makes it a track-count bin), then $P_i = -1$. If P_i points to an attenuator which appears inside parentheses on the FM card, it is negative.

I_i = for a regular bin, the program number of a cell, possibly

a pseudocell, that contains the material m specified on the FM card. For an attenuator, I_i = −1.

n_i = for a regular bin, the number of entries, including both reaction numbers and operators, in the bin description. If the list of reaction numbers in the bin includes the elastic or the total cross section, n_i is negative. For an attenuator, n_i is the number of entries, including material numbers and superficial-density values. If a regular bin appears on the FM card within parentheses that also contain an attenuator, n_i has 10000000 added to it for an attenuator to the right of the bin and 20000000 for an attenuator to the left.

R_{ji} = for a regular bin, a reaction number or operator. The sum operator, indicated by a colon on the FM card, is stored here as the value 100003. For an attenuator, the R_{ji} are alternating cell numbers and superficial-density values.

Dose Function

The value of IPTAL(LIPT+3,2,K) is the location in TDS of the dose function table for tally K. The first element in the table is the length N. It is followed by the N values of the energy and then the N values of the function. N is preceded by an indicator of the type of interpolation: 0 for log-log, 1 for lin-log, 2 for log-lin, and 3 for lin-lin.

TFC(3,20,NTAL) Array   Tally Fluctuation Charts

The value of TFC(LTFC+I,J,K) is the tally value (I=1), the error (I=2), and the figure of merit (I=3) for line J of the tally fluctuation chart for tally K. The tally bin involved is designated by the eight indices in JTF(LJTF+I,K) for I = 1 to 8. The number of histories run at the point where the entries for a line were calculated is stored in NPC(J). Initially a line is calculated every 1000 histories. When the 20th line is generated, the history increment is doubled. When the time comes to generate the 21st line, the odd-numbered lines are eliminated, the data in line J are moved to line J/2 for J = 2 to 20 by 2, and the new data are put in line 11.

RTP(LRT) Array   Information from Tally Input Cards

The information from most tally input cards is stored without much modification in temporary array RTP. Numbers are stored as is. Special characters are encoded. After all the input cards have been read, subroutine ITALLY sets up the permanent tally control arrays from the

information in RTP. The main reason for this two-step process is that some of the control arrays depend in a complicated way on information from more than one input card. It is simpler to generate the control arrays with all the input data available at the same time than to do it as the cards are read.

Pointer array IPNT(2,21,0:NTAL) is defined as the tally cards are read. The information from tally card type J of tally K begins at RTP(LRTP+IPNT(LIPN+1,J,K)) and occupies IPNT(LIPN+2,J,K) elements of RTP. The tally card type numbers J are given in KRQ(3,N) for each type N of input card. KRQ(3,N) is defined by DATA statements in block data subprogram IBLDAT. KRQ(3,N) is zero for nontally input cards. There is no tally card type 1. IPNT(LIPN+1,1,K) is used for bits that reflect T or NT on certain cards and indicate whether a total bin needs to be included. The value of IPNT(LIPN+1,2,K) is 1, 2, 3, 4, or 5, depending on whether the F card for the tally has blank, X, Y, Z, or W with the F, and it is negative if there is an asterisk on that card.

D. Accounting Arrays

MCNP regularly collects and prints data on the behavior of the particles transported through the problem geometry. This is accounting information which shows what MCNP actually did, in contrast to the tallies which are estimates of physically measurable quantities. The accounting information is essential to a user who is trying to make his problem run faster. The arrays where the accounting data are collected and the titles of the tables where they are printed are as follows.

PAX	Ledger of Net Creation and Loss
PAC	Problem Activity in Each Cell (Print Table 126)
PWB	Weight Balance in Each Cell (Print Table 130)
PAN	Activity of Each Nuclide in Each Cell (Print Table 140)
PCC, FEBL	Summary of Photons Produced in Neutron Collisions

<u>PAX(2,6,16) Array</u> Ledger of Net Creation and Loss

The value of PAX(I,J,K) is the total of type J data for particle type I and mechanism K.

<u>J</u>	
1	number of tracks created
2	weight created
3	energy created

4 number of tracks terminated
5 weight terminated
6 energy terminated

K	Creation Mechanism	Loss Mechanism
1	source	escape
2	scattering (p-annihilation)	scattering (fluorescence)
3	fission (from neutrons)	capture
4	(n,xn) (2Xfluorescence)	energy cutoff
5	forced collision	time cutoff
6	weight cutoff	weight cutoff
7	weight window	weight window
8	cell importance	cell importance
9	energy importance	energy importance
10	DXTRAN	DXTRAN
11	exponential transform	exponential transform
12	spare (scattering)	dead fission (pair production)
13–16	spares	spares

Photon mechanisms that differ from mechanisms for neutrons are given in parentheses.

For the printed table, the weight totals are divided by the number of histories and the energy totals are divided by the total weight of source particles.

PAC(2,10,MXA) Array Problem Activity in Each Cell

The value of PAC(LPAC+I,J,K) is the total of type J data for particle type I in cell K. If a particle becomes lost, a small amount of erroneous information gets added into PAC.

J	
1	number of tracks entering cell K
2	population of cell K: the number of tracks, including source tracks, entering for the first time
3	number of collisions in cell K
4	weight entering collisions
5	energy * time interval in cell K * weight
6	energy * path length * weight
7	path length in cell K
8	mean free path * path length * weight
9	time interval * weight
10	path length * weight

APPENDIX E
Arrays

The quantities printed are

Tracks Entering = PAC(LPAC+I,1,K)
Population = PAC(LPAC+I,2,K)
Collisions = PAC(LPAC+I,3,K)
Collisions * weight (per history) = PAC(LPAC+I,4,K) / number of histories
Number Weighted Energy = PAC(LPAC+I,5,K) / PAC(LPAC+I,9,K)
Flux Weighted Energy = PAC(LPAC+I,6,K) / PAC(LPAC+I,10,K)
Average Track Weight (Relative) = PAC(LPAC+I,10,K) * importance of cell K / [PAC(LPAC+I,7,K) * importance of source cell]
Average Track MFP = PAC(LPAC+I,8,K) / PAC(LPAC+I,10,K)

PWB(2,15,MXA) Array   Weight Balance in Each Cell

The value of PWB(LPWB+I,J,K) is the net weight change of type J for particle type I in cell K. If a particle becomes lost, a small amount of erroneous information gets added into PWB. Table values are divided by the number of histories before being printed.

J	Table Heading	
	External	
1	Source	weight of created source particles
2	Entering	weight of particles entering cell K
3	Time Cutoff	weight of particles killed by time cutoff
4	Energy Cutoff	weight of particles killed by energy cutoff
5	Exiting	weight of particles exiting cell K
	Physical (neutrons)	
6	Fission Loss	weight of neutrons lost to fission
7	Fission Neutrons	weight of fission neutrons produced
8	(n,xn)	weight of new tracks produced by other inelastic processes
9	Capture	weight lost to capture
	Physical (photons)	
6	From Neutrons	weight of neutron-induced photons
7	Pair Production	net weight created by pair production
8	Fluorescence	net weight created by double fluorescence
9	Capture	weight lost to capture
	Variance Reduction	
10	Weight Cutoff	net weight change due to weight cutoff
11	Cell Importance	net weight change due to splitting and

		Russian roulette in importance sampling
12	Exponential Transform	net weight change due to exponential transform
13	DXTRAN	net weight change due to DXTRAN
14	Energy Importance	net weight change due to energy splitting and Russian roulette
15	Weight Window	net weight change due to weight-window Russian roulette

PAN(2,6,NPN) Array Activity of Each Nuclide in Each Cell

The value of PAN(LPAN+I,J,IPAN(LIPA+K)+N−1) is the total of type J data for particle type I for the Nth nuclide in cell K. IPAN(LIPA+M+1) = IPAN(LIPA+M) + number of nuclides in the material of cell M. IPAN(LIPA+1) = 1 and NPN = IPAN(LIPA+MXA+1)−1. If a particle becomes lost, a small amount of erroneous information gets added into PAN.

J (for neutrons)
1 number of collisions with Nth nuclide of cell K
2 weight entering collisions
3 weight lost to capture
4 weight gain by fission
5 weight gain by other inelastic processes
6 unused

J (for photons)
1−3 same as for neutrons
4 number of neutron-induced photons
5 weight of neutron-induced photons
6 energy * weight of neutron-induced photons

The quantities printed are
Total Collisions = PAN(LPAN+I,1,L)
Collisions * Weight = PAN(LPAN+I,2,L) / number of histories
Weight Lost to Capture = PAN(LPAN+1,3,L) / number of histories
Weight Gain by Fission = PAN(LPAN+1,4,L) / number of histories
Weight Gain by (n,xn) = PAN(LPAN+1,5,L) / number of histories
Total From Neutrons = PAN(LPAN+2,4,L)
Weight from Neutrons = PAN(LPAN+2,5,l) / number of histories
Avg Photon Energy = PAN(LPAN+2,6,L) / PAN(LPAN+2,5,L)

APPENDIX E
Arrays

PCC(3,MXA) Array   Summary of Photons Produced in Neutron Collisions

The value of PCC(LPCC+J,K) is the total of type J data for cell K. If a particle becomes lost, a small amount of erroneous information may be added into PCC.

J
1 number of neutron-induced photons
2 weight of neutron-induced photons
3 weight * energy of neutron-induced photons

The quantities printed are
Number of Photons = PCC(LPCC+1,K)
Weight Per Source Neutron = PCC(LPCC+2,K) / number of histories
Energy Per Source Neutron = PCC(LPCC+3,K) / number of histories
Avg Photon Energies = PCC(LPCC+3,K) /PCC(LPCC+2,K)
Energy/ Per Source Neutron = PCC(LPCC+3,K) /
[cell mass * number of histories]
Weight/Neutron Collision = PCC(LPCC+2,K) / PAC(LPAC+1,4,K)
Energy/Neutron Collision = PCC(LPCC+3,K) / PAC(LPAC+1,4,K)

FEBL(2,16) Array   Summary of Photons Produced in Neutron Collisions

The value of FEBL(J,K) is the total of type J data for photon energy bin K. The energy bin bounds are in array EBL(16) in common block /TABLES/.

J
1 number of neutron-induced photons
2 weight of neutron-induced photons

The quantities printed are
Number of Photons = FEBL(1,K)
Number Frequency = FEBL(1,K) / PAX(2,1,3)
Weight of Photons = FEBL(2,K) / number of histories
Weight Frequency = FEBL(2,K) / PAX(2,2,3)

Appendix F

NUCLEAR DATA FORMATS

Nuclear data libraries for all versions of MCNP through Version 2D were nonportable binary libraries in ACE format (A Compact ENDF), generated and read by nonstandard I/O routines. Our other data libraries – photon, dosimetry, and thermal – were also accessed by MCNP using these nonstandard (but extremely fast) Los Alamos routines.

With the advent of a portable code (Version 3), portable nuclear data libraries became a necessity. Therefore, MCNP now reads from three *types* of nuclear data tables:

1. Type 1 – standard formatted tables (sequential, 80 characters per record). These portable libraries are used to transmit data from one installation to another. They are very bulky and must be read slowly. Probably few installations will use Type 1 tables in MCNP directly but rather will generate Type 2 tables from Type 1 tables (using the MAKXSF code– see Appendix C).
2. Type 2 – standard unformatted tables. These are direct-access, binary files that are locally generated from Type 1 tables. They are not portable. Type 2 tables are used most because they are more compact and faster to read than Type 1 tables.
3. Type 3 – nonstandard tables. At present, the only Type 3 tables are the direct-access, binary, nonportable ACE libraries that remain as an option to users at Los Alamos and on the MFE network.

Previous versions of the MCNP manual have described the format of Type 3 (ACE) libraries only. In this Appendix, we will be concerned with Type 1, Type 2, and Type 3 tables.

Nuclear data tables exist for five *classes* of data: continuous-energy neutron, discrete-reaction neutron, photon interaction, neutron dosimetry, and $S(\alpha,\beta)$ thermal. A user should think of a nuclear data table as an entity that contains evaluation-dependent information about one of the five *classes* of data for a specific target isotope, element, or material. For how the data are used in MCNP, a user does not need to know whether a particular table is in Type 1, Type 2, or Type 3 format. For any ZAID, the data contained on Type 1, Type 2, and Type 3 tables are identical.

When we refer to nuclear data libraries, we are talking about a series of nuclear data tables all contained on one file. All tables on a single library must be of the same *type* but not necessarily of the same *class*. There is no reason, other than convenience, for having nuclear data

libraries; MCNP could read exclusively from self-contained, individual nuclear data tables.

I. XSDIR – NUCLEAR DATA DIRECTORY FILE

MCNP determines where to find nuclear data tables for each ZAID in a problem based on information contained in a system-dependent directory file, XSDIR. All nuclear data tables supported by Group X-6 are represented on XSDIR.

The directory file has seven or more entries for each table. They are:

1. Name of the table	character * 10
2. Main atomic weight	real
3. File name	character * 8
4. Access route	character * 70
5. File type	integer
6. Record number	integer
7. Table length	integer
8. Record length	integer
9. Number of entries per record	integer
10. Temperature	real
11.+ Pairs of ZA and atomic weight	integer, real

1. Name of the Table. This is usually the ZAID: 3 characters for Z, 3 characters for A (or something else), a decimal point, 2 characters for evaluation identification, and a tenth character used to identify continuous energy tables by the letter C, discrete-reaction tables by D, dosimetry tables by Y, $S(\alpha,\beta)$ thermal tables by T, and photon tables by P. For the $S(\alpha,\beta)$ tables, the first 6 characters are a mnemonic character string.

2. Main Atomic Weight. The atomic weight here must be exactly the same as it appears in the cross-section table. It is used by the initiation part of MCNP to convert between atomic density and mass density. Later, in XACT, it is checked against the value in the cross-section table.

3. File Name. The file name is a string of up to eight characters in a form allowed by the local installation.

4. Access Route. The access route is a string of up to 70 characters which tells how to get hold of the file if it is not already accessible. At Los Alamos, it is a CFS path name. Elsewhere it might be a tape label. If there is no access route, this entry should be a zero.

5. File Type. 1, 2, or 3.

6. Record Number. For Type 1 and Type 2 files, the record number is the number of the first record of the table. For a Los Alamos Type 3 file (ACE file), this entry is the disk address of the first word of the table.

7. Table Length. A nuclear data table consists of two blocks of information. The first block is a collection of pointers, counters, and character information. The second block is a solid sequence of numbers. For Type 1 and Type 2 tables, the table length is the length of the second block. For ACE tables, it is the total length. The table length is needed so XACT can adjust the field length to make room for the largest table in the problem.

8. Record Length. If a file has direct access, it has a processor-dependent attribute called the record length. For unformatted files, the record length on NOS is the number of words in the record, allowing one word for each integer or real item, two for each complex or double-precision item, and one for every 10 characters in character items. For CTSS, COS, and IBM, it is the number of 8-bit bytes in the record.

9. Number of Entries per Record. This is the number of items per record in the second block of the table. It applies only to Type 2 files. For some systems, including FTN5/NOS, it is the same as the record length.

10. Temperature. The temperature (in MeV) at which a neutron table was processed may be entered here. MCNP might someday use this entry in some sort of automatic cross-section table selection scheme.

11. Pairs of ZA and Atomic Weight. These are required for photon tables only and are used only in Mode 2 problems. An entry is necessary for each isotope of the element. As with the main atomic weight, the atomic weights here must match the entries in the cross-section table exactly.

The directory file is sequential and formatted with 80-character records containing free-field entries delimited by blanks. If a table requires more than one record, the continuation is indicated by a + at the end of any record that is followed by a continuation record. If an entry is irrelevant, it should be a zero. Irrelevant entries at the end may be omitted.

II. NUCLEAR DATA TABLES

The remainder of this Appendix is designed for the user who wishes to know a great deal about how data are stored in nuclear data tables and in MCNP. We will first describe how to find a specific table on a Type 1, Type 2, or Type 3 library. Then we will document the detailed format of the various blocks of information for each *class* of data.

Before getting into too much detail, we must introduce the concept that three arrays are associated with each nuclear data table. The NXS array contains various counters and flags. The JXS array contains pointers. The third array, XSS, contains all of the data.

Regardless of the format (Type) of a specific table, the data that constitute the NXS, JXS, and XSS arrays are all contained within the table. The arrays will be manipulated internally by MCNP. Within a nuclear data table, the counter and pointer arrays are dimensioned to NXS(16) and JXS(32). In MCNP the same arrays are dimensioned to NXS(16,MAXE) and JXS(32,MAXE), with MAXE currently set equal to 50. The XSS array is part of dynamically allocated common.

To locate data for a specific table (external to MCNP) it is necessary to extract several parameters associated with that table from the directory file, XSDIR. The file name obviously indicates the name of the library that the table is stored on. Other important parameters from the viewpoint of this Appendix are file type, record number (IRN), table length (ITL), and number of entries per record (NER).

A. Locating Data on a Type 1 Table

The record number from XSDIR, IRN, is the number of the first record of the table. Since Type 1 tables are 80-character card-image files, the record number is the line number at which the table begins. The first 12 records (lines) contain miscellaneous information as well as the NXS and JXS arrays. The format is as follows:

Record (Line) Number			
Relative	Absolute	Contents	Format
1	IRN	HZ,AW(0),TZ,HD	A10,2E12.0,1X,A10
2	IRN+1	HK,HM	A70,A10
3–6	IRN+2	(IZ(I),AW(I),I=1,16)	4(I7,F11.0)
7–8	IRN+6	(NXS(I),I=1,16)	8I9
9–12	IRN+8	(JXS(I),I=1,32)	8I9

The variables are defined in Tables F.1–F.3.

The XSS array immediately follows the JXS array. All data from the XSS array are read into MCNP with a 4E20.0 format. (When Type 1 tables are created, floating-point numbers are written in 1PE20.12 format and integers are written in I20 format.) The length of the XSS array is given by the table length, ITL, in the directory (also by NXS(1) in the table itself). The number of records required for the XSS array is (ITL+3)/4. A Type 1 library is shown in Figure F.1.

Layout of a Type 1 Library

Starting Record (Line) Number	Number of Records	Contents
$IRN_1=1$	12	misc. including NXS_1, JXS_1
IRN_1+12	$(ITL_1+3)/4$	XSS_1
IRN_2	12	misc. including NXS_2, JXS_2
IRN_2+12	$(ITL_2+3)/4$	XSS_2
. . .	. . .	. . .
IRN_n	12	misc. including NXS_n, JXS_n
IRN_n+12	$(ITL_n+3)/4$	XSS_n

n=number of tables contained on library
IRN_i, ITL_i are the record numbers and tables lengths from XSDIR

Figure F.1

Table F.1
Definition of the NXS Array

NTY	1 or 2 Continuous energy or Discrete reaction	3 Dosimetry	4 Thermal	5 Photon
NXS(1)	Length of second block of data	Length of second block of data	Length of second block of data	Length of second block of data
NXS(2)	ZA=1000*Z+A	ZA=1000*Z+A	IDPNI=inelastic scattering mode	Z
NXS(3)	NES=number of energies		NIL=inelastic dimensioning parameter	NES=number of energies
NXS(4)	NTR=number of reactions excluding elastic	NTR=number of reactions	NIEB=number of inelastic exiting energies	NFLO=length of the fluorescence data divided by 4
NXS(5)	NR=number of reactions having secondary neutrons excluding elastic		IDPNC=elastic scattering mode	
NXS(6)	NTRP=number of photon production reactions		NCL=elastic dimensioning parameter	
NXS(7)			IFENG=secondary energy mode	
NXS(8)				
......				
......				
......				
NXS(16)				

Note that many variables are not used, allowing for easy expansion in the future.

Table F.2
Definition of the JXS Array

NTY	1 or 2 Continuous energy or Discrete reaction	3 Dosimetry	4 Thermal	5 Photon
JXS(1)	ESZ=location of energy table	LONE=location of first word of table	ITIE=location of inelastic energy table	ESZG=location of energy table
JXS(2)	NU=location of fission nu data		ITIX=location of inelastic cross sections	JINC=location of incoherent form factors
JXS(3)	MTR=location of MT array	MTR=location of MT array	ITXE=location of inelastic energy/angle distributions	JCOH=location of coherent form factors
JXS(4)	LQR=location of Q-value array		ITCE=location of elastic energy table	JFLO=location of fluorescence data
JXS(5)	TYR=location of reaction type array		ITCX=location of elastic cross sections	LHNM=location of heating numbers
JXS(6)	LSIG=location of table of cross-section locators	LSIG=location of table of cross-section locators	ITCA=location of elastic angular distributions	
JXS(7)	SIG=location of cross sections	SIGD=location of cross sections		
JXS(8)	LAND=location of table of angular distribution locators			
JXS(9)	AND=location of angular distributions			

Table F.2 (cont)

NTY	1 or 2 Continuous energy or Discrete reaction	3 Dosimetry	4 Thermal	5 Photon
JXS(10)	LDLW=location of table of energy distribution locators			
JXS(11)	DLW=location of energy distributions			
JXS(12)	GPD=location of photon production data			
JXS(13)	MTRP=location of photon production MT array			
JXS(14)	LSIGP=location of table of photon production cross section locators			
JXS(15)	SIGP=location of photon production cross sections			
JXS(16)	LANDP=location of table of photon production angular distribution locators			
JXS(17)	ANDP=location of photon production angular distributions			

Table F.2 (cont)

NTY	1 or 2 Continuous energy or Discrete reaction	3 Dosimetry	4 Thermal	5 Photon
JXS(18)	LDLWP=location of table of photon production energy dist ribution locators			
JXS(19)	DLWP=location of photon production energy distributions			
JXS(20)	YP=location of table of yield multipliers			
JXS(21)	FIS=location of total fission cross section			
JXS(22)	END=location of last word of this table	END=location of last word of this table		
JXS(23)				
......				
......				
......				
JXS(32)				

Note that many variables are not used, allowing for easy expansion in the future.

All pointers in the JXS array refer to locations in the XSS array.

JXS(1) always points to the first entry in the second block of data.

Table F.3
Definition of Miscellaneous Variables on Data Tables

HZ - 10 character name (ZAID) of table. The form of HZ is

ZZZAAA.nnC	continuous-energy neutron
ZZZAAA.nnD	discrete-reaction neutron
ZZZAAA.nnY	dosimetry
XXXXXX.nnT	thermal $S(\alpha,\beta)$
ZZZ000.nnP	photon

where ZZZ is the atomic number
AAA is the mass number
XXXXXX for thermal data is a Hollerith name or abbreviation of the material
nn is the evaluation identifier

AW(0) - atomic weight ratio; the atomic weight divided by the mass of a neutron

TZ - temperature at which the data were processed (in MeV)

HD - 10-character date when data were processed

HK - 70-character comment

HM - 10-character MAT identifier

(IZ(I),AW(I),I=1,16) - 16 pairs of ZZZAAAs and atomic weight ratios. The pairs are found only on photon tables and exist for all isotopes of the element for which neutron transport data are available. IZs alone are entered on thermal tables to indicate for which isotope(s) the scattering data are appropriate.

B. Locating Data on a Type 2 Table

A standard unformatted file consists of many records, each with NER entries, where NER is the number of entries per record defined on XSDIR. A Type 2 nuclear data table consists of one record that contains pointers, counters, and character information, followed by one or more records containing the XSS array.

The information contained in the first record for each table is the same as that contained in the first twelve lines of a Type 1 table described above. The variables, in order, are HZ, AW(0), TZ, HD, HK, HM, (IZ(I),AW(I),I=1,16), (NXS(I),I=1,16), (JXS(I),I=1,32). The variables are defined in Tables F.1-F.3. HZ, HD, and HM are 10-character variables and

HK is a 70-character variable. Floating-point variables may be double precision in some cases. The number of words contained in this "package" of information is therefore different for different computing systems. The remainder of the first record is empty.

The next NREC records (NREC≥1) contain the XSS data array, with NREC=(ITL+NER-1)/NER, where ITL is the table length.

A Type 2 library is shown in Figure F.2.

Layout of a Type 2 Library

Record Number	Contents	
IRN_1 = 1	misc. including NXS_1, JXS_1	
2	XSS_1	$NER<ITL_1 \leq 2*NER$
3	XSS_1 (cont)	
IRN_2 = 4	misc. including NXS_2, JXS_2	
5	XSS_2	$ITL_2 \leq NER$
. .	. .	
IRN_n = MAX-3	misc. including NXS_n, JXS_n	
MAX-2	XSS_n	
MAX-1	XSS_n (cont)	$2*NER<ITL_n \leq 3*NER$
MAX	XSS_n (cont)	

(Records per table are examples only)

n=number of tables contained on library
MAX=number of records contained on library
IRN_i, ITL_i, NER are the record numbers, table lengths, and entries per record from XSDIR

Figure F.2

C. Locating Data on a Type 3 Table

Type 3 tables are compatible with MCNP only for continuous-energy or discrete-reaction neutron data. Therefore the following discussion does not apply to photon, dosimetry, or thermal tables.

The disk address of the first word of the table is IRN, the record number from XSDIR. The JXS, NXS, and XSS arrays are interspersed throughout the next ITL words. ITL is the total table length and not simply the length of the XSS array, as in Type 1 and Type 2 tables.

The first 40 words in a Type 3 table contain pointers, counters, and descriptive information that can be related to some of the parameters in the JXS and NXS arrays. The relationships are as follows:

Disk Address		Relationship to	Disk Address		Relationship to
Relative	Absolute	MCNP Variable	Relative	Absolute	MCNP Variable
1	IRN	NXS(2),HZ*	13	IRN+12	JXS(10)+41**
2	IRN+1	AW(0)	14	IRN+13	JXS(11)+41**
3	.	NXS(4)	15	.	JXS(12)+41**
4	.	NXS(3)	16	.	JXS(4)+41**
5	.	NXS(5)	17	.	*,***
6		Not used;JXS(1)=1	18		JXS(2)+41**
7		JXS(3)+41**	19		JXS(21)+41**
8		JXS(5)+41**	20		NXS(1)+41
9		JXS(6)+41**			or NXS(1)+61***
10		JXS(7)+41**	24–30		HK
11		JXS(8)+41**	31		HM
12		JXS(9)+41**	40		HD

* The first nine characters of HZ are from relative location 1 (of the form ZZAAA.nn). If the value in relative location 17 is greater than or equal to zero, the tenth character of HZ is C; if the value is less than zero, the tenth character of HZ is D.

** If the value in any of these locations is zero, the corresponding value in the JXS array is also zero.

*** If the absolute value of relative location 17 is 100, NXS(1) is 61 less than the value in relative location 20; otherwise NXS(1) is 41 less than the value in relative location 20.

MCNP variables are defined in Tables F.1–F.3.

The XSS array begins with XSS(1) in relative disk address 43 and initially extends for a total of 2*NXS(3) words. This is followed by a record marker on the Type 3 table which is ignored in the XSS array. The XSS array continues with XSS(2*NXS(3)+1) at relative disk address 44+2*NXS(3) and extends for a total of either

NXS(1)−2*NXS(3) words if absolute value of relative location 17 is not equal to 100

or JXS(12)+599−NXS(3) words if absolute value of relative location 17 is equal to 100.

If there are no expanded photon production data for this isotope (that is, if the absolute value of relative location 17 is not equal to 100), then the table is complete at this point. Otherwise the next 20 words contain more counters and pointers needed for the NXS and JXS arrays. They are

Disk Address Relative	Absolute	Relationship to MCNP Variable
1	IRN+642+NXS(3)+JXS(12)	
2	IRN+643+NXS(3)+JXS(12)	
3	.	NXS(6)
7	.	JXS(13)−580−JXS(12)−NXS(3)
9	.	JXS(14)−580−JXS(12)−NXS(3)
10		JXS(15)−580−JXS(12)−NXS(3)
11		JXS(16)−580−JXS(12)−NXS(3)
12		JXS(17)−580−JXS(12)−NXS(3)
13		JXS(18)−580−JXS(12)−NXS(3)
14		JXS(19)−580−JXS(12)−NXS(3)
19		JXS(20)−580−JXS(12)−NXS(3)

The XSS array continues immediately with XSS(JXS(12)+NXS(3)+600) at relative disk address IRN+662+NXS(3)+JXS(12) and extends for a total of NXS(1)−JXS(12)−NXS(3)−599 words. Regardless of whether expanded photon production data exist, the total length of XSS is NXS(1).

A dictionary in the first 200 words of a Type 3 library is no longer used by MCNP. The first 198 words contain pairs of ZAIDs and absolute locations of data for each ZAID. Word 199 contains the library name and word 200 is the date when the library was created.

A Type 3 library is shown in Figure F.3.

Layout of a Type 3 Library

Absolute Disk Location	Contents
1 to 200	dictionary; no longer used by MCNP
IRN_1 to IRN_1+ITL_1-1	NXS_1, JXS_1, XSS_1
IRN_2 to IRN_2+ITL_2-1	NXS_2, JXS_2, XSS_2
. . .	. . .
IRN_n to IRN_n+ITL_n-1	NXS_n, JXS_n, XSS_n

n=number of tables contained on library
IRN_i, ITL_i are the record numbers and table lengths from XSDIR

Figure F.3

D. Locating Nuclear Data Tables in MCNP

The NXS and JXS arrays exist in MCNP for each nuclear data table. The information contained in the (2-dimensional) arrays in MCNP mirrors the information contained in NXS and JXS (1-dimensional) on the individual tables. The current dimensions are NXS(16) and JXS(32) on the nuclear data tables and NXS(16,50) and JXS(32,50) in MCNP. In the code, the arrays are usually referenced as NXS(I,IEX) and JXS(I,IEX), where IEX is the index to a particular table.

The data from all cross-section tables used in an MCNP problem are in the XSS array, a part of dynamically allocated common. The data from the first table appear first, followed by the data from the second table, etc., as shown in Figure F.4. The pointers in the JXS array indicate absolute locations in the XSS array.

Diagram of Nuclear Data Storage in MCNP

common shared with other information	Nuclear Data Table 1	Nuclear Data Table 2	...	Nuclear Data Table n

XSS

Figure F.4

The definitions of the variables in the NXS and JXS arrays (Tables F.1 and F.2) are the same in MCNP as on a nuclear data table with one slight exception (for discrete-reaction neutron tables, NXS(16,IEX) is used in MCNP as an indicator of whether discrete tables in a problem have cross sections tabulated on identical energy grids). Although the definitions of the variables are the same, the contents are generally not. Pointers in the JXS array are pointing to locations in the MCNP internal XSS array that are different from the locations in the nuclear data table XSS array. Flags in the NXS array will generally retain the same value in MCNP. Counters in the NXS array may retain the same value, primarily depending on the degree to which MCNP is able to expunge data for a particular problem.

E. Individual Data Blocks

Several blocks of data exist for every cross-section table. The format of an individual block is essentially the same in MCNP as on a nuclear data table. In either case, the absolute location of a data block in the XSS array is determined by pointers in the JXS array. The specific data blocks available for a particular table are a function of the *class* of data. We next describe the detailed format of individual data blocks for each *class* of data.

F. Data Blocks for Neutron Transport Tables

The format of individual data blocks found on neutron transport tables is identical for continuous-energy (NTY=1) and discrete-reaction (NTY=2) tables. Therefore, the format for both are described in this section. There are 21 different data blocks for neutron transport tables; 12 involve neutron-in neutron-out data, 9 involve neutron-in photon-out data. The format of four of the neutron-photon blocks is identical to corresponding neutron-neutron blocks. Therefore, only 17 unique data blocks need be described. All data blocks are now listed with a brief description of their contents and the table numbers in which their formats are detailed.

**Note: In the tables that follow these descriptions, it is understood that NXS(I) or JXS(I) really means NXS(I,IEX) or JXS(I,IEX) when locating data blocks in MCNP.

1. ESZ Block – contains the main energy grid for the table and the total, absorption, and elastic cross sections as well as the average heating numbers. The ESZ Block always exists. See Table F.4.

2. NU Block – contains prompt and/or total $\bar{\nu}$ as a function of incident neutron energy. The NU Block exists only for fissionable isotopes (that is, if JXS(2)≠0). See Table F.5.

3. MTR Block – contains list of ENDF/B MT numbers for all neutron reactions other than elastic scattering. The MTR Block exists for all isotopes that have reactions other than elastic scattering (that is, all isotopes with NXS(4)≠0). See Table F.6.

4. LQR Block – contains list of kinematic Q-values for all neutron reactions other than elastic scattering. The LQR Block exists if NXS(4)≠0. See Table F.7.

5. TYR Block – contains information about the type of reaction for all neutron reactions other than elastic scattering. Information for each reaction includes the number of secondary neutrons and whether secondary neutron angular distributions are in the laboratory or center-of-mass system. The TYR Block exists if NXS(4)≠0. See Table F.8.

6. LSIG Block – contains list of cross-section locators for all neutron reactions other than elastic scattering. The LSIG Block exists if NXS(4)≠0. See Table F.9.

7. SIG Block – contains cross sections for all reactions other than elastic scattering. The SIG Block exists if NXS(4)≠0. See Table F.10.

8. LAND Block – contains list of angular-distribution locators for all reactions producing secondary neutrons. The LAND Block always exists. See Table F.11.

9. AND Block – contains angular distributions for all reactions producing secondary neutrons. The AND Block always exists. See Table F.12.

10. LDLW Block – contains list of energy distribution locators for all reactions producing secondary neutrons except for elastic scattering. The LDLW Block exists if NXS(5)≠0. See Table F.13.

11. DLW Block – contains energy distributions for all reactions producing secondary neutrons except for elastic scattering. The DLW Block exists if NXS(5)≠0. See Table F.14.

12. GPD – contains the total photon production cross section tabulated on the ESZ energy grid and a 30X20 matrix of secondary photon energies. The GPD Block exists only for those evaluations that provide

coupled neutron/photon information (that is, if JXS(12)≠0). See Table F.15.

13. MTRP Block – contains list of MT numbers for all photon production reactions. (We will use the term "photon production reaction" for any information describing a specific neutron-in photon-out reaction.) The MTRP Block exists if NXS(6)≠0. See Table F.6.

14. LSIGP Block – contains list of cross-section locators for all photon production reactions. The LSIGP Block exists if NXS(6)≠0. See Table F.9.

15. SIGP Block – contains cross sections for all photon production reactions. The SIGP Block exists if NXS(6)≠0. See Table F.16.

16. LANDP Block – contains list of angular-distribution locators for all photon production reactions. The LANDP Block exists if NXS(6)≠0. See Table F.17.

17. ANDP Block – contains photon angular distributions for all photon- production reactions. The ANDP Block exists if NXS(6)≠0. See Table F.18.

18. LDLWP Block – contains list of energy-distribution locators for all photon production reactions. The LDLWP Block exists if NXS(6)≠0. See Table F.13.

19. DLWP Block – contains photon energy distributions for all photon production reactions. The DLWP Block exists if NXS(6)≠0. See Table F.14.

20. YP Block – contains list of MT identifiers of neutron reaction cross sections required as photon production yield multipliers. The YP Block exists if NXS(6)≠0. See Table F.19.

21. FIS Block – contains the total fission cross section tabulated on the ESZ energy grid. The FIS Block exists if JXS(21)≠0. See Table F.20.

Table F.4
ESZ Block

Location in XSS	Parameter	Description
JXS(1)	E(I),I=1,NXS(3)	Energies
JXS(1)+NXS(3)	σ_t(I),I=1,NXS(3)	Total cross sections
JXS(1)+2*NXS(3)	σ_a(I),I=1,NXS(3)	Total absorption cross sections
JXS(1)+3*NXS(3)	σ_{el}(I),I=1,NXS(3)	Elastic cross sections
JXS(1)+4*NXS(3)	H_{ave}(I),I=1,NXS(3)	Average heating numbers

Table F.5
NU Block

There are three possibilities for the NU Block:

1.	XSS(JXS(2))=0	no NU Block
2.	XSS(JXS(2))>0	either prompt $\bar{\nu}$ or total $\bar{\nu}$ is given. The NU array begins at location XSS(KNU) where KNU=JXS(2).
3.	XSS(JXS(2))<0	both prompt $\bar{\nu}$ and total $\bar{\nu}$ are given. The prompt NU Array begins at XSS(KNU) where KNU=JXS(2)+1; the total NU array begins at XSS(KNU) where KNU=JXS(2)+ABS(XSS(JXS(2)))+1.

The NU Array has two forms if it exists:

a) Polynomial function form of NU Array:

Location in XSS	Parameter	Description
KNU	LNU=1	Polynomial function flag
KNU+1	NC	Number of coefficients
KNU+2	C(I),I=1,NC	Coefficients

$$\bar{\nu}(E) = \sum_{I=1}^{NC} C(I)*E**(I-1) \qquad E \text{ in MeV}$$

b) Tabular data form of NU array:

Location in XSS	Parameter	Description
KNU	LNU=2	Tabular data flag
KNU+1	NR	Number of interpolation regions
KNU+2	NBT(I),I=1,NR	ENDF interpolation parameters
KNU+2+NR	INT(I),I=1,NR	If NR=0, NBT and INT are omitted and linear-linear interpolation is used.
KNU+2+2*NR	NE	Number of energies
KNU+3+2*NR	E(I),I=1,NE	Tabular energy points
KNU+3+2*NR+NE	$\bar{\nu}$(I),I=1,NE	Corresponding values of $\bar{\nu}$

Table F.6
MTR, MTRP Blocks

Location in XSS	Parameter	Description
LMT	MT_1	First ENDF reaction available
LMT+1	MT_2	Second ENDF reaction available
.	.	.
.	.	.
.	.	.
LMT+NMT−1	MT_{NMT}	Last ENDF reaction available

where LMT=JXS(3) for MTR Block
LMT=JXS(13) for MTRP Block
NMT=NXS(4) for MTR Block
NMT=NXS(6) for MTRP Block

Note: For MTR Block: MT_1, MT_2, ... are standard ENDF MT numbers, that is, MT=16=(n,2n); MT=17=(n,3n); etc.
For MTRP Block: the MT values are somewhat arbitrary. To understand the scheme used for numbering the photon production MTs, it is necessary to realize that in ENDF/B format, more than one photon can be produced by a particular neutron reaction that is itself specified by a single MT. Each of these photons is produced with an individual energy-dependent cross section. For example, MT 102 (radiative capture) might be responsible for 40 photons, each with its own cross section, angular distribution, and energy distribution. We need 40 photon MTs to represent the data; the MTs are numbered 102001, 102002, ... , 102040. Therefore, if ENDF/B MT "N" is responsible for "M" photons, we shall number the photon MTs 1000*N+1, 1000*N+2, ... , 1000*N+M.

Table F.7
LQR Block

Location in XSS	Parameter	Description
JXS(4)	Q_1	Q-value of reaction MT_1
JXS(4)+1	Q_2	Q-value of reaction MT_2
.	.	.
.	.	.
.	.	.
JXS(4)+NXS(4)−1	$Q_{NXS(4)}$	Q-value of reaction $MT_{NXS(4)}$

Note: The MT_i's are given in the MTR Block.

Table F.8
TYR Block

Location in XSS	Parameter	Description
JXS(5)	TY_1	Neutron release for reaction MT_1
JXS(5)+1	TY_2	Neutron release for reaction MT_2
.	.	.
.	.	.
.	.	.
JXS(5)+NXS(4)−1	$TY_{NXS(4)}$	Neutron release for reaction $MT_{NXS(4)}$

Note: the possible values of TY_i are ±1, ±2, ±3, ±4, 19, 0. The sign indicates the system for scattering: negative = CM system; positive = LAB system. Thus if TY_i = −3, three neutrons are released for reaction MT_i, and the data on the cross-section tables used to determine the exiting neutrons' angles are given in the CM system.

TY_i=19 indicates fission. The number of secondary neutrons released is determined from the fission $\bar{\nu}$ data found in the NU Block.

TY_i=0 indicates absorption (ENDF reactions MT>100); no neutrons are released.

The MT_i's are given in the MTR Block.

Table F.9
LSIG, LSIGP Blocks

Location in XSS	Parameter	Description
LXS	$LOCA_1$=1	Loc. of cross sections for reaction MT_1
LXS+1	$LOCA_2$	Loc. of cross sections for reaction MT_2
.	.	.
.	.	.
.	.	.
LXS+NMT−1	$LOCA_{NMT}$	Loc. of cross sections for reaction MT_{NMT}

where LXS=JXS(6) for LSIG Block
LXS=JXS(14) for LSIGP Block
NMT=NXS(4) for LSIG Block
NMT=NXS(6) for LSIGP Block

Note: All locators are relative to JXS(7) for LSIG or JXS(15) for LSIGP. The MT_i's are given in the MTR Block for LSIG or the MTRP Block for LSIGP.

Table F.10
SIG Block

Location in XSS	Description
$JXS(7)+LOCA_1-1$	Cross-section array* for reaction MT_1
$JXS(7)+LOCA_2-1$	Cross-section array* for reaction MT_2
.	.
.	.
.	.
$JXS(7)+LOCA_{NXS(4)}-1$	Cross-section array* for reaction $MT_{NXS(4)}$

*The i^{th} array has the form:

Location in XSS	Parameter	Description
$JXS(7)+LOCA_i-1$	IE_i	Energy grid index for reaction MT_i
$JXS(7)+LOCA_i$	NE_i	Number of consecutive entries for MT_i
$JXS(7)+LOCA_i+1$	$\sigma_i[E(K)], K=IE_i, IE_i+NE_i-1$	Cross sections for reaction MT_i

Note: The values of $LOCA_i$ are given in the LSIG Block. The energy grid E(K) is given in the ESZ Block. The energy grid index IE_i corresponds to the first energy in the grid at which a cross section is given. The MT_i's are defined in the MTR Block.

Table F.11
LAND Block

Location in XSS	Parameter	Description
JXS(8)	$LOCB_1=1$	Loc. of angular dist. data for elastic scattering
JXS(8)+1	$LOCB_2$	Loc. of angular dist. data for reaction MT_1
.	.	.
.	.	.
.	.	.
JXS(8)+NXS(5)	$LOCB_{NXS(5)+1}$	Loc. of angular dist. data for reaction $MT_{NXS(5)}$

Note: All locators ($LOCB_i$) are relative to JXS(9). If $LOCB_i=0$, no angular distribution data are given for this reaction, and isotropic scattering is assumed in either the LAB or CM system. Choice of LAB or CM system depends upon value for this reaction in the TYR Block. The MT_i's are given in the MTR Block.

Table F.12
AND Block

Location in XSS	Description
JXS(9)+$LOCB_1$−1	Angular distribution array* for elastic scattering
JXS(9)+$LOCB_2$−1	Angular distribution array* for reaction MT_1
.	.
.	.
JXS(9)+$LOCB_{NXS(5)+1}$−1	Angular distribution array* for reaction $MT_{NXS(5)}$

Note: The values of $LOCB_i$ are given in the LAND Block. If $LOCB_i$=0, no angular distribution array is given and scattering is isotropic in either the LAB or CM system. Choice of LAB or CM system depends on value in the TYR Block. The MT_i's are given in the MTR Block.

*The i^{th} array has the form:

Location in XSS	Parameter	Description
JXS(9)+$LOCB_i$−1	NE	Number of energies at which angular distributions are tabulated.
JXS(9)+$LOCB_i$	E(J),J=1,NE	Energy grid
JXS(9)+$LOCB_i$+NE	LC(J),J=1,NE	Location of tables associated with energies E(J)
JXS(9)+LC(1)−1	P(1,K),K=1,33	32 equiprobable cosine bins for scattering at energy E(1)
JXS(9)+LC(2)−1	P(2,K),K=1,33	32 equiprobable cosine bins for scattering at energy E(2)
.	.	.
.	.	.
JXS(9)+LC(NE)−1	P(NE,K),K=1,33	32 equiprobable cosine bins for scattering at energy E(NE)

Note: All values of LC(J) are relative to JXS(9). If LC(J)=0, no table is given for energy E(J) and scattering is isotropic in the coordinate system indicated by entry in the TYR Block.

Table F.13
LDLW, LDLWP Blocks

Location in XSS	Parameter	Description
LED	$LOCC_1$	Loc. of energy dist. data for reaction MT_1
LED+1	$LOCC_2$	Loc. of energy dist. data for reaction MT_2
.	.	.
.	.	.
LED+NMT−1	$LOCC_{NMT}$	Loc. of energy dist. data for reaction MT_{NMT}

where LED=JXS(10) for LDLW Block
LED=JXS(18) for LDLWP Block
NMT=NXS(5) for LDLW Block
NMT=NXS(6) for LDLWP Block

Note: All locators are relative to JXS(11) for LDLW or JXS(19) for LDLWP. The MT_i's are given in the MTR Block for LDLW or MTRP Block for LDLWP.

Table F.14
DLW,DLWP Blocks

Location in XSS	Description
$JED+LOCC_1-1$	Energy distribution array* for reaction MT_1
$JED+LOCC_2-1$	Energy distribution array* for reaction MT_2
.	.
.	.
.	.
$JED+LOCC_{NMT}-1$	Energy distribution array* for reaction MT_{NMT}

where JED=JXS(11) for DLW
JED=JXS(19) for DLWP
NMT=NXS(5) for DLW
NMT=NXS(6) for DLWP

Note: Values of $LOCC_i$ are given in the LDLW and LDLWP Blocks. Values of MT_i are given in the MTR and MTRP Blocks.

*The i^{th} array has the form:

Location in XSS	Parameter	Description
$LDIS+LOCC_i-1$	LNW_1	Location of next law. If LNW_1=0, then law LAW_1 is used regardless of other circumstances.
$LDIS+LOCC_i$	LAW_1	Name of this law
$LDIS+LOCC_i+1$	$IDAT_1$	Location of data for this law relative to LDIS
$LDIS+LOCC_i+2$	NR	Number of interpolation regions to define law applicability regime
$LDIS+LOCC_i+3$	NBT(I),I=1,NR	ENDF interpolation parameters.
$LDIS+LOCC_i+3+NR$	INT(I),I=1,NR	If NR=0, NBT and INT are omitted and linear-linear interpolation is used.

LDIS+$LOCC_i$+3+2*NR	NE	Number of energies
LDIS+$LOCC_i$+4+2*NR	E(I),I=NE	Tabular energy points
LDIS+$LOCC_i$+4 +2*NR+NE	P(I),I=1,NE	Probability of law validity. If the particle energy, E, is E<E(1) then P(E)=P(1) If E>E(NE) then P(E)=P(NE). If more than 1 law is given, then LAW_1 is used only if ξ<P(E) where ξ is a random number between 0 and 1.
LDIS+$IDAT_1$-1	LDAT(I),I=1,L**	Law data array for LAW_1. The length, L, of the law data array, LDAT, is determined from parameters within LDAT. The various law data arrays, LDAT, for each law, LAW_i, are given in the following tables.
LDIS+LNW_1-1	LNW_2	Location of next law
LDIS+LNW_1	LAW_2	Name of this law
LDIS+LNW_1+1	$IDAT_2$	Location of data for this law
.	.	.
.	.	.
.	.	.

etc.

where LDIS=JXS(11) for DLW
LDIS=JXS(19) for DLWP

Note: The locators $LOCC_i$ are defined in the LDLW Block or the LDLWP Block.
All locators (LNW_i , $IDAT_i$) are relative to LDIS.

**We now define the format of the LDAT array for each law. Laws 2 and 4 are used to describe the spectra of secondary photons from neutron collisions. All laws except for Law 2 are used to describe the spectra of scattered neutrons. In the following tables we provide relative locations of data in the LDAT array rather than absolute locations in the XSS array. The preceding table defines the starting location of the LDAT array within the XSS array.

a. LAW_i=1 Tabular Equiprobable Energy Bins (From ENDF Law 1)

Location	Parameter	Description
LDAT(1)	NR	Interpolation scheme
LDAT(2)	NBT(I),I=1,NR	between tables of E_{out}

LDAT(2+NR)	INT(I),I=1,NR	If NR=0 or if INT(I) ≠1 (histogram), linear-linear interpolation is used
LDAT(2+2*NR)	NE	Number of incident energies tabulated
LDAT(3+2*NR)	E_{in}(I),I=1,NE	List of incident energies for which E_{out} is tabulated
LDAT(3+2*NR+NE)	NET	Number of outgoing energies in each E_{out} table
LDAT(4+2*NR+NE)	E_{out_1}(I),I=1,NET E_{out_2}(I),I=1,NET . . $E_{out_{NE}}$(I),I=1,NET	E_{out} tables are NET boundaries of NET-1 equally likely energy intervals. Linear-linear interpolation is used between intervals.

b. LAW_i=2 Discrete Photon Energy

Location	Parameter	Description
LDAT(1)	LP	Indicator of whether the photon is a primary or nonprimary photon
LDAT(2)	EG	Photon energy (if LP=0 or LP=1), or Binding energy (if LP=2)

If LP=0 or LP=1, the photon energy is EG

If LP=2, the photon energy is EG+(AWR)/(AWR+1)*EN
where AWR is the atomic weight ratio
and EN is the neutron energy

c. LAW_i=3 Level Scattering (From ENDF Law 3)

$$\mathrm{LDAT(1)} = \left(\frac{A+1}{A}\right) |Q| \qquad \mathrm{LDAT(2)} = \left(\frac{A}{A+1}\right)^2$$

$$E_{out}^{CM} = \mathrm{LDAT(2)*(E-LDAT(1))}$$

where E_{out}^{CM} = outgoing center-of-mass energy
E = incident energy
A = atomic weight ratio
Q = Q-value

The outgoing neutron energy in the laboratory system, E_{out}^{LAB}, is

$$E_{out}^{LAB}=E_{out}^{CM}+\left\{E+2\mu_{cm}(A+1)(EE_{out}^{CM})^{1/2}\right\}/(A+1)^2,$$

where μ_{cm}=cosine of the center-of-mass scattering angle.

d. LAW_i=4 Continuous Tabular Distribution (From ENDF Law 1)

Location	Parameter	Description
LDAT(1)	NR	Number of interpolation regions
LDAT(2)	NBT(I),I=1,NR	ENDF interpolation parameters. If NR=0,
LDAT(2+NR)	INT(I),I=1,NR	NBT and INT are omitted and linear-linear interpolation is used.
LDAT(2+2*NR)	NE	Number of energies at which distributions are tabulated
LDAT(3+2*NR)	E(I),I=1,NE	Incident neutron energies
LDAT(3+2*NR+NE)	L(I),I=1,NE	Locations of distributions (relative to JXS(11) or JXS(18))
Data for E(1) (let K=3+2*NR+2*NE):		
LDAT(K)	INTT	Interpolation scheme for subsequent data INTT=1 histogram distribution INTT=2 linear-linear distribution
LDAT(K+1)	NP	Number of points in the distribution
LDAT(K+2)	EOUT(I),I=1,NP	Outgoing energy grid
LDAT(K+2+NP)	PDF(I),I=1,NP	Probability density function
LDAT(K+2+2*NP)	CDF(I),I=1,NP	Cumulative density function
Data for E(2):		
.	.	.
.	.	.

e. LAW_i=5 General Evaporation Spectrum (From ENDF Law 5)

Location	Parameter	Description
LDAT(1)	NR	Interpolation scheme between T's
LDAT(2)	NBT(I),I=1,NR	
LDAT(2+NR)	INT(I),I=1,NR	
LDAT(2+2*NR)	NE	Number of incident energies tabulated
LDAT(3+2*NR)	E(I),I=1,NE	Incident energy table
LDAT(3+2*NR+NE)	T(I),I=1,NE	Tabulated function of incident energies
LDAT(3+2*NR+2*NE)	NET	Number of X's tabulated
LDAT(4+2*NR+2*NE)	X(I),I=1,NET	Tabulated probabilistic function

E_{out} = X(ξ)*T(E), where X(ξ) is a randomly sampled table of X's, and E is the incident energy.

f. $LAW_i=7$ Simple Maxwell Fission Spectrum (From ENDF Law 7)

Location	Parameter	Description
LDAT(1)	NR	Interpolation scheme between T's
LDAT(2)	NBT(I),I=1,NR	
LDAT(2+NR)	INT(I),I=1,NR	
LDAT(2+2*NR)	NE	Number of incident energies tabulated
LDAT(3+2*NR)	E(I),I=1,NE	Incident energy table
LDAT(3+2*NR+NE)	T(I),I=1,NE	Tabulated T's
LDAT(3+2*NR+2*NE)	U	Restriction energy

$$f(E \rightarrow E_{out}) = C*SQRT(E_{out}) \exp(-E_{out}/T(E))$$

with restriction $0 \leq E_{out} \leq E-U$

$$C = T^{-3/2}\left[\frac{\sqrt{\pi}}{2} \operatorname{erf}\left(\sqrt{(E-U)/T}\right) - \sqrt{(E-U)/T}\, e^{-(E-U)/T}\right]^{-1}$$

g. $LAW_i=9$ Evaporation Spectrum (From ENDF Law 9)

Location	Parameter	Description
LDAT(1)	NR	Interpolation scheme between T's
LDAT(2)	NBT(I),I=1,NR	
LDAT(2+NR)	INT(I),I=1,NR	
LDAT(2+2*NR)	NE	Number of incident energies tabulated
LDAT(3+2*NR)	E(I),I=1,NE	Incident energy table
LDAT(3+2*NR+NE)	T(I),I=1,NE	Tabulated T's
LDAT(3+2*NR+2*NE)	U	Restriction energy

$$f(E \rightarrow E_{out}) = C*E_{out}\exp(-E_{out}/T(E))$$

with restriction $0 \leq E_{out} \leq E-U$

$$C = T^{-2}\left[1-e^{-(E-U)/T}\left(1+(E-U)/T\right)\right]^{-1}$$

h. $LAW_i=11$ Energy Dependent Watt Spectrum (From ENDF Law 11)

Location	Parameter	Description
LDAT(1)	NR_a	Interpolation scheme between a's
LDAT(2)	$NBT_a(I),I=1,NR_a$	
LDAT($2+NR_a$)	$INT_a(I),I=1,NR_a$	

LDAT($2+2*NR_a$)	NE_a	Number of incident energies tabulated for $a(E_{in})$ table
LDAT($3+2*NR_a$)	$E_a(I),I=1,NE_a$	Incident energy table
LDAT($3+2*NR_a+NE_a$)	$a(I),I=1,NE_a$	Tabulated a's
let $L=3+2*(NR_a+NE_a)$		
LDAT(L)	NR_b	Interpolation scheme between b's
LDAT(L+1)	$NBT_b(I),I=1,NR_b$	
LDAT($L+1+NR_b$)	$INT_b(I),I=1,NR_b$	
LDAT($L+1+2*NR_b$)	NE_b	Number of incident energies tabulated for $b(E_{in})$ table
LDAT($L+2+2*NR_b$)	$E_b(I),I=1,NE_b$	Incident energy table
LDAT($L+2+2*NR_b+NE_b$)	$b(I),I=1,NE_b$	Tabulated b's
LDAT($L+2+2*NR_b+2*NE_b$)	U	Rejection energy

$$f(E\rightarrow E_{out}) = C_o \exp[-E_{out}/a(E)\sinh[b(E)*E_{out}]^{1/2}$$

with restriction $0 \le E_{out} < E-U$

This law is sampled by the rejection scheme in LA-5061-MS (R11, pg. 45).

i. $LAW_i=22$ Tabular Linear Functions (from UK Law 2)

Location	Parameter	Description
LDAT(1)	NR	Interpolation parameters
LDAT(2)	NBT(I),I=1,NR	which are not used by MCNP
LDAT(2+NR)	INT(I),I=1,NR	(histogram interpolation is assumed)
LDAT(2+2*NR)	NE	Number of incident energies tabulated
LDAT(3+2*NR)	$E_{in}(I),I=1,NE$	List of incident energies for E_{out} tables
LDAT(3+2*NR+NE)	LOCE(I),I=1,NE	Locators of E_{out} tables (relative to JXS(11))
Data for $E_{in}(1)$ (Let L=3+2*NR+2*NE):		
LDAT(L)	NF_1	if $E_{in}(I) \le E < E_{in}(I+1)$
LDAT(L+1)	$P_1(K),K=1,NF_1$	and ξ is a random
LDAT($L+1+NF_1$)	$T_1(K),K=1,NF_1$	number [0,1] then if
LDAT($L+1+2*NF_1$)	$C_1(K),K=1,NF_1$	
Data for $E_{in}(2)$:		$\sum_{k=1}^{k=K} P_1(K) < \xi \le \sum_{k=1}^{k=K+1} P_1(K)$
.	.	$E_{out} = C_1(K)*(E-T_1(K))$
.	.	
.	.	

j. LAW_i=24 (From UK Law 6)

Location	Parameter	Description
LDAT(1)	NR	Interpotation parameters
LDAT(2)	NBT(I),I=1,NR	which are not used by MCNP
LDAT(2+NR)	INT(I),I=1,NR	(histogram interpolation is assumed)
LDAT(2+2*NR)	NE	Number of incident energies
LDAT(3+2*NR)	E_{in}(I),I=1,NE	List of incident energies for which T is tabulated
LDAT(3+2*NR+NE)	NET	Number of outgoing values in each table
LDAT(4+2*NR+NE)	T_1(I),I=1,NET	Tables are NET boundaries
	T_2(I),I=1,NET	of NET-1 equally likely
	.	intervals. Linear-linear
	.	interpolation is used between
	T_{NE}(I),I=1,NET	intervals.

$E_{out} = T_K(I)*E$

where $T_K(I)$ is sampled from the above tables

E is the incident neutron energy

Table F.15

GPD Block

Location in XSS	Parameter	Description
JXS(12)	σ_γ(I),I=1,NXS(3)	Total photon production cross section
JXS(12)+NXS(3)	EG(1,K),K=1,20	20 equally likely outgoing photon energies for incident neutron energy E < EN(2)
JXS(12)+NXS(3)+20	EG(2,K),K=1,20	20 equiprobable outgoing photon energies for incident neutron energy EN(2) ≤ E < EN(3)
.	.	.
.	.	.
.	.	.
JXS(12)+NXS(3)+580	EG(30,K),K=1,20	20 equiprobable outgoing photon energies for incident neutron energy E ≥ EN(30)

Notes: (1) The discrete incident neutron energy array in MeV is EN(J),J=1,30: 1.39E-10, 1.52E-7, 4.14E-7, 1.13E-6, 3.06E-6, 8.32E-6, 2.26E-5, 6.14E-5, 1.67E-4, 4.54E-4, 1.235E-3,

3.35E-3, 9.23E-3, 2.48E-2, 6.76E-2, .184, .303, .500, .823,
1.353, 1.738, 2.232, 2.865, 3.68, 6.07, 7.79, 10., 12.,
13.5, 15.
(2) The equiprobable photon energy matrix is used only for those older tables that do not provide expanded photon production data.

Table F.16
SIGP Block

Location in XSS	Description
JXS(15)+$LOCA_1$-1	Cross-section array* for reaction MT_1
JXS(15)+$LOCA_2$-1	Cross-section array* for reaction MT_2
.	.
.	.
JXS(15)+$LOCA_{NXS(6)}$-1	Cross-section array* for reaction $MT_{NXS(6)}$

*The i^{th} array has two possible forms, depending on the first word in the array:

(a) If MFTYPE=12 (Yield Data)

Location in XSS	Parameter	Description
JXS(15)+$LOCA_i$-1	MFTYPE	12
JXS(15)+$LOCA_i$	MTMULT	Neutron MT whose cross section should multiply the yield
JXS(15)+$LOCA_i$+1	NR	Number of interpolation regions
JXS(15)+$LOCA_i$+2	NBT(I),I=1,NR	ENDF interpolation parameters. If NR=0,
JXS(15)+$LOCA_i$+2+NR	INT(I),I=1,NR	NBT and INT are omitted and linear-linear interpolation is used.
JXS(15)+$LOCA_i$+2+2*NR	NE	Number of energies at which the yield is tabulated
JXS(15)+$LOCA_i$+3+2*NR	E(I),I=1,NE	Energies
JXS(15)+$LOCA_i$+3+2*NR+NE	Y(I),I=1,NE	Yields

$$\sigma_{\gamma,i} = Y(E) * \sigma_{MTMULT}(E)$$

(b) If MFTYPE=13 (Cross-Section Data)

Location in XSS	Parameter	Description
JXS(15)+$LOCA_i$-1	MFTYPE	13
JXS(15)+$LOCA_i$	IE	Energy grid index
JXS(15)+$LOCA_i$+1	NE	Number of consecutive entries
JXS(15)+$LOCA_i$+2	$\sigma_{\gamma,i}$[E(K)],K=IE,IE+NE-1	Cross sections for reaction MT_i

Note: The values of $LOCA_i$ are given in the LSIGP Block. The energy grid E(K) is given in the ESZ Block. The MT_i's are defined in the MTRP Block.

Table F.17
LANDP Block

Location in XSS	Parameter	Description
JXS(16)	$LOCB_1=1$	Loc. of angular dist. data for reaction MT_1
JXS(16)+1	$LOCB_2$	Loc. of angular dist. data for reaction MT_2
.	.	.
.	.	.
.	.	.
JXS(16)+NXS(6)−1	$LOCB_{NXS(6)}$	Loc. of angular dist. data for reaction $MT_{NXS(6)}$

Note: All locators ($LOCB_i$) are relative to JXS(17). If $LOCB_i=0$, there are no angular distribution data given for this reaction and isotropic scattering is assumed in the LAB system. MT_i's are defined in the MTRP Block.

Table F.18
ANDP Block

Location in XSS	Description
$JXS(17)+LOCB_1-1$	Angular distribution array* for reaction MT_1
$JXS(17)+LOCB_2-1$	Angular distribution array* for reaction MT_2
.	.
.	.
.	.
$JXS(17)+LOCB_{NXS(6)}-1$	Angular distribution array* for reaction $MT_{NXS(6)}$

Note: The values of $LOCB_i$ are given in the LANDP Block. If $LOCB_i=0$, then no angular distribution array is given and scattering is isotropic in the LAB system. The MT_i's are given in the MTRP Block.

*The i^{th} array has the form:

Location in XSS	Parameter	Description
$JXS(17)+LOCB_i-1$	NE	Number of energies at which angular distributions are tabulated.
$JXS(17)+LOCB_i$	E(J),J=1,NE	Energy grid

$JXS(17)+LOCB_i+NE$	$LC(J),J=1,NE$	Location of tables associated with energies $E(J)$
$JXS(17)+LC(1)-1$	$P(1,K),K=1,33$	32 equiprobable cosine bins for scattering at energy $E(1)$
$JXS(17)+LC(2)-1$	$P(2,K),K=1,33$	32 equiprobable cosine bins for scattering at energy $E(2)$
.	.	.
.	.	.
.	.	.
$JXS(17)+LC(NE)-1$	$P(NE,K),K=1,33$	32 equiprobable cosine bins for scattering at energy $E(NE)$

Note: All values of $LC(J)$ are relative to $JXS(17)$. If $LC(J)=0$, no table is given for energy $E(J)$ and scattering is isotropic in the LAB system.

Table F.19
YP Block

Location in XSS	Parameter	Description
JXS(20)	NYP	Number of neutron MTs to follow
JXS(20)+1	MTY(I),I=1,NYP	Neutron MTs

Note: The MTY array contains all neutron MTs that are required as photon-production yield multipliers (See Table F.16). MCNP needs this information when expunging data.

Table F.20
FIS Block

Location in XSS	Parameter	Description
JXS(21)	IE	Energy grid index
JXS(21)+1	NE	Number of consecutive entries
JXS(21)+2	$\sigma_f[E(K)],K=IE,IE+NE-1$	Total fission cross sections

Note: The FIS Block generally is not provided on individual data tables because the total fission cross section is a redundant quantity [that is, $\sigma_{f,tot}(E) = \sigma_{n,f}(E) + \sigma_{n,n'f}(E) + \sigma_{n,2nf}(E) + \sigma_{n,3nf}(E)$]. MCNP forms the FIS Block if conditions warrant (for example, for KCODE calculations, coupled neutron photon calculations, etc.). The energy grid E(K) is given in the ESZ Block.

G. Data Blocks for Dosimetry Tables

Dosimetry tables (NTY=3) provide cross sections that are useful as response functions with the FM feature in MCNP. They can never be used for actual neutron transport. Therefore, there is a much more limited set of information available on dosimetry tables than on neutron transport tables (NTY=1 or 2). In fact, only three blocks of data exist on dosimetry tables. The three blocks are now listed along with the table numbers in which their formats are detailed.

1. MTR Block – contains a list of the MT numbers for all reactions provided on the table. The MTR Block always exists on dosimetry tables. The format of the block is identical to that of the MTR Block previously described for neutron transport tables. See Table F.6.

2. LSIG Block – contains a list of cross-section locators for all reactions provided on the table. The LSIG Block always exists on dosimetry tables. The format of the block is identical to that of the LSIG Block previously described for neutron transport tables. See Table F.9.

3. SIGD Block – contains (energy, cross-section) pairs for all reactions provided on the table. The LSIGP Block always exists on dosimetry tables. See Table F.21.

Table F.21
SIGD Block

Location in XSS	Description
$JXS(7)+LOCA_1-1$	Cross-section array* for reaction MT_1
$JXS(7)+LOCA_2-1$	Cross-section array* for reaction MT_2
.	.
.	.
$JXS(7)+LOCA_{NXS(4)}-1$	Cross-section array* for reaction $MT_{NXS(4)}$

*the i^{th} array is of the form:

Location in XSS	Parameter	Description
$JXS(7)+LOCA_i-1$	NR	Number of interpolation regions
$JXS(7)+LOCA_i$	NBT(I),I=1,NR	ENDF interpolation parameters. If NR=0,
$JXS(7)+LOCA_i+NR$	INT(I),I=1,NR	NBT and INT are omitted and linear-linear interpolation is assumed.
$JXS(7)+LOCA_i+2*NR$	NE	Number of (energy,cross section) pairs
$JXS(7)+LOCA_i+1+2*NR$	E(I),I=1,NE	Energies
$JXS(7)+LOCA_i+1+2*NR+NE$	σ(I),I=1,NE	Cross sections

Note: The locators ($LOCA_i$) are provided in the LSIG Block. The MT_i's are given in the MTR Block.

H. Data Blocks for Thermal $S(\alpha,\beta)$ Tables

Data from thermal $S(\alpha,\beta)$ tables (NTY=4) provide a complete representation of thermal neutron scattering by molecules and crystalline solids. Cross sections for elastic and inelastic scattering are found on the tables (typically for neutron energies below 4 eV). A coupled energy angle representation is used to describe the spectra of inelastically scattered neutrons. Angular distributions for elastic scattering are also provided.

Four unique blocks of data are associated with $S(\alpha,\beta)$ tables. We now briefly describe each of the four data blocks and give the table numbers in which their formats are detailed.

1. ITIE Block – contains the energy-dependent inelastic scattering cross sections. The ITIE Block always exists. See Table F.22.
2. ITCE Block – contains the energy-dependent elastic scattering cross sections. The ITCE Block exists if JXS(4)≠0. See Table F.23.
3. ITXE Block – contains coupled energy angle distributions for inelastic scattering. The ITXE Block always exists. See Table F.24.
4. ITCA Block – contains angular distributions for elastic scattering. The ITCA Block exists if JXS(4)≠0 and NXS(6)≠−1. See Table F.25.

Table F.22
ITIE Block

Location in XSS	Parameter	Description
JXS(1)	NE_{in}	Number of inelastic energies
JXS(1)+1	$E_{in}(I), I=1,NE_{in}$	Energies
JXS(1)+1+NE_{in}	$\sigma_{in}(I), I=1,NE_{in}$	Inelastic cross sections

Note: JXS(2)=JXS(1)+1+NE_{in}
Linear-linear interpolation is assumed between adjacent energies.

Table F.23
ITCE Block

Location in XSS	Parameter	Description
JXS(4)	NE_{el}	Number of elastic energies
JXS(4)+1	$E_{el}(I), I=1,NE_{el}$	Energies
JXS(4)+1+NE_{el}	$P(I), I=1,NE_{el}$	(See Below)

If $NXS(5) \neq 4$:

$\sigma_{el}(I)=P(I)$, with linear-linear interpolation between points

If $NXS(5)=4$:

$\sigma_{el}(E)=P(I)/E$, for $E_{el}(I) \leq E < E_{el}(I+1)$

Note: $JXS(5)=JXS(3)+1+NE_{el}$

Table F.24

ITXE Block

For NXS(2)=3 (equally-likely cosines; currently the only scattering mode allowed for inelastic angular distributions)

Location in XSS	Parameter	Description
JXS(3)	$E_1^{OUT}[E_{in}(1)]$	First of NXS(4) equally-likely outgoing energies for inelastic scattering at $E_{in}(1)$
JXS(3)+1	$\mu_I(1\text{-->}1)$, I=1,NXS(3)+1	Equally-likely discrete cosines for scattering from $E_{in}(1)$ to $E_1^{OUT}[E_{in}(1)]$
JXS(3)+2+NXS(3)	$E_2^{OUT}[E_{in}(1)]$	Second of NXS(4) equally-likely outgoing energies for inelastic scattering at $E_{in}(1)$
JXS(3)+3+NXS(3)	$\mu_I(1\text{-->}2)$, I=1,NXS(3)+1	Equally-likely discrete cosines for scattering from $E_{in}(1)$ to $E_2^{OUT}[E_{in}(1)]$
.	.	.
.	.	.
.	.	.
JXS(3)+(NXS(4)-1)* (NXS(3)+2)	$E_{NXS(4)}^{OUT}[E_{in}(1)]$	Last of NXS(4) equally-likely outgoing energies for inelastic scattering at $E_{in}(1)$
JXS(3)+(NXS(4)-1)* (NXS(3)+2)+1	$\mu_I(1\text{-->}NXS(4))$, I=1,NXS(3)+1	Equally-likely discrete cosines for scattering from $E_{in}(1)$ to $E_{NXS(4)}^{OUT}[E_{in}(1)]$
.	.	.
. (Repeat for all remaining values of E_{in})		.
.	.	.

Note: Incident inelastic energy grid $E_{in}(I)$ is given in ITIE Block. Linear-linear interpolation is assumed between adjacent values of E_{in}.

Table F.25

ITCA Block

Location in XSS	Parameter	Description
JXS(6)	$\mu_I[E_{el}(1)]$, I=1,NXS(6)+1	Equally-likely discrete cosines for elastic scattering at $E_{el}(1)$
JXS(6)+NXS(6)+1	$\mu_I[E_{el}(2)]$,	Equally-likely discrete cosines for

	$l=1,NXS(6)+1$	elastic scattering at $E_{el}(2)$
.	.	.
.	.	.
.	.	.
$JXS(6)+(NE_{el}-1)*(NXS(6)+1)$	$\mu_l[E_{el}(NE_{el})]$, $l=1,NXS(6)+1$	Equally-likely discrete cosines for elastic scattering at $E_{el}(NE_{el})$

Note: Incident elastic energy grid $E_{el}(I)$ and number of energies NE_{el} are given in ITCE Block. Linear-linear interpolation is assumed between adjacent values of E_{el}.

I. Data Blocks for Photon Transport Tables

Only five data blocks are found on photon transport tables (NTY=5). Information contained on the blocks includes: cross sections for coherent and incoherent scattering, pair production, and the photoelectric effect; scattering functions and form factors that modify the differential Klein-Nishina and Thomson cross sections; energy deposition data; and fluorescence data. The five data blocks are now listed, along with brief descriptions and table numbers where detailed formats may be found.

1. ESZG Block - contains the coherent, incoherent, photoelectric, and pair production cross sections, all tabulated on a common energy grid. The ESZG Block always exists. See Table F.26.

2. JINC Block - contains the incoherent scattering functions that are used to modify the differential Klein-Nishina cross section. The JINC Block always exists. See Table F.27.

3. JCOH Block - contains the coherent form factors that are used to modify the differential Thomson cross section. The JCOH Block always exists. See Table F.28.

4. JFLO Block - contains fluorescence data. The JFLO Block exists if $NXS(4)\neq 0$. See Table F.29.

5. LHNM Block - contains average heating numbers. The LHNM Block always exists. See Table F.30.

Table F.26
ESZG Block

Location in XSS	Parameter	Description
JXS(1)	$\ln[E(I),I=1,NXS(3)]$	Logarithms of energies
JXS(1)+NXS(3)	$\ln[\sigma_{IN}(I),I=1,NXS(3)]$	Logarithms of incoherent cross sections
JXS(1)+2*NXS(3)	$\ln[\sigma_{CO}(I),I=1,NXS(3)]$	Logarithms of coherent cross sections
JXS(1)+3*NXS(3)	$\ln[\sigma_{PE}(I),I=1,NXS(3)]$	Logarithms of photoelectric cross sections
JXS(1)+4*NXS(3)	$\ln[\sigma_{PP}(I),I=1,NXS(3)]$	Logarithms of pair production cross sections

Note: Linear-linear interpolation is performed on the logarithms as stored, resulting in effective log-log interpolation for the cross sections. If a cross section is zero, a value of 0.0 is stored on the data table.

Table F.27
JINC Block

Location in XSS	Parameter	Description
JXS(2)	$FF_{INC}(I)$, I=1,21	Incoherent scattering functions

Note: The scattering functions for all elements are tabulated on a fixed set of v(I), where v is the momentum of the recoil electron (in inverse angstroms). The grid is: v(I),I=1,21 / 0. , .005 , .01 , .05 , .1 , .15 , .2 , .3 , .4 , .5 , .6 , .7 , .8 , .9 , 1. , 1.5 , 2. , 3. , 4. , 5. , 8. /
Linear-linear interpolation is assumed between adjacent v(I).
The constants v(I) are stored in the VIC array in common block RBLDAT.

Table F.28
JCOH Block

Location in XSS	Parameter	Description
JXS(3)	$FFINT_{COH}(I)$, I=1,55	Integrated coherent form factors
JXS(3)+55	$FF_{COH}(I)$, I=1,55	Coherent form factors

Note: The form factors for all elements are tabulated on a fixed set of v(I), where v is the momentum transfer of the recoil electron (in inverse angstroms). The grid is: v(I),I=1,55 / 0. , .01 , .02 , .03 , .04 , .05 , .06 , .08 , .10 , .12 , .15 , .18 , .20 , .25 , .30 , .35 , .40 , .45 , .50 , .55 , .60 , .70 , .80 , .90 , 1.0 , 1.1 , 1.2 , 1.3 , 1.4 , 1.5 , 1.6 , 1.7 , 1.8 , 1.9 , 2.0 , 2.2 , 2.4 , 2.6 , 2.8 , 3.0 , 3.2 , 3.4 , 3.6 , 3.8 , 4.0 , 4.2 , 4.4 , 4.6 , 4.8 , 5.0 , 5.2 , 5.4 , 5.6 , 5.8 , 6.0 /
The integrated form factors are tabulated on a fixed set of $v(I)^2$, where the v(I) are those defined above. See LA-5157-MS for a description of the integrated form factors and the sampling technique used in MCNP.
The constants v(I) are stored in the VCO array. The constants $v(I)^2$ are stored in the WCO array. Both arrays are in common block RBLDAT.

Table F.29
JFLO Block

Location in XSS	Parameter	Description
JXS(4)	e(1),Φ(1),Y(1),F(1)	(See Below)
JXS(4)+4	e(2),Φ(2),Y(2),F(2)	(See Below)
.	.	.
.	.	.
.	.	.
JXS(4)+4*(NXS(4)-1)	e[NXS(4)],Φ[NXS(4)],Y[NXS(4)],F[NXS(4)]	(See Below)

A complete description of the parameters given in this block may be found in LA-5240-MS. Briefly:

e(I) are the edge energies
Φ(I) are relative probabilities of ejection from various shells
Y(I) are yields
and F(I) are fluorescent energies.

Table F.30
LHNM Block

Location in XSS	Parameter	Description
JXS(5)	H_{ave}(I),I=1,NXS(3)	Average heating numbers

Note: Log-log interpolation is performed between adjacent heating numbers. The units of H_{ave} are MeV per collision. Heating numbers are tabulated on the energy grid given in the ESZG Block.

Appendix G

CROSS-SECTION LIBRARIES

This appendix is divided into three sections. The first section lists some of the more frequently used ENDF B reaction types that can be used with the FMn input card. Secondly, the currently available $S(\alpha,\beta)$ identifiers for the MTm card are listed in Table G.1. Thirdly, a brief description of the available libraries is provided and a complete list of the evaluations in these libraries is given.

I. *ENDF B REACTION TYPES*

The following list includes some of the more useful reactions for use with the FMn input card, but it is not the complete ENDF B list. The complete list can be found in the ENDF B manual.

R	Microscopic Cross-Section Description
1	Total
2	Elastic
16	(n,2n)
17	(n,3n)
18	Total fission (n,fx) if and only if MT=18 is used to specify fission in the original evaluation.
19	(n,f)
20	(n,n′f)
21	(n,2nf)
22	(n,n′)α
38	(n,3nf)
51	(n,n′) to 1st excited state
52	(n,n′) to 2nd excited state
.	.
.	.
.	.
90	(n,n′) to 40th excited state
91	(n,n′) to continuum
102	(n,γ)
103	(n,p)
104	(n,d)

105	(n,t)
106	(n,^{3}He)
107	(n,α)

In addition, for files based on ENDF B-V *only*, the following special reactions are available:

203	total proton production
204	total deuterium production
205	total tritium production
206	total ^{3}He production
207	total alpha production

Notes:

(1) The R number for tritium production varies from nuclide to nuclide and evaluation to evaluation. For ^{6}Li and ^{7}Li

^{6}Li(n,α)t	R = 107	AWRE, ENDF B-III,ENDF B-IV, and Los Alamos Sublibrary evaluations
^{6}Li(n,t)α	R = 105	All ENDL evaluations and ENDF B-V
^{7}Li(n,n′)t,α	R = 33	ENDL evaluations
^{7}Li(n,n′)α,t	R = 22	AWRE evaluation
^{7}Li(n,n′)α,t	R = 91	ENDF B-III, ENDF B-IV, and ENDF B-V evaluations
^{7}Li(n,n′)α,t	R = 205	T-2 evaluation (3007.55)

(2) The nomenclature between MCNP and ENDF B is inconsistent in that MCNP refers to the number of the reaction type as R whereas ENDF B uses MT. They are one and the same, however. The problem arises since MCNP has an MT input card used for the S(α,β) thermal treatment.

(3) The user looking for total production of p, d, t, ^{3}He, and ^{4}He should be warned that in some evaluations, such processes are represented using reactions with R (or MT) numbers other than the standard ones given in the above list. This is of particular importance with the so-called "pseudolevel" representation of certain reactions which take place in light isotopes. For example, the ENDF B-V evaluation of carbon includes cross sections for the (n,n′3α) reaction in R = 52 to 58. The user interested in particle production from light isotopes should contact X-6 to check for the existence of pseudolevels and thus possible deviations from the above standard reaction list.

II. S(α,β) IDENTIFIERS FOR THE MTm CARD

Table G.1
Thermal S(α,β) Cross-Section Tables

ZAID	Description	*Isotopes	Temperature (K)
LWTR.01T	Light water	1001	300
LWTR.02T	Light water	1001	400
LWTR.03T	Light water	1001	500
LWTR.04T	Light water	1001	600
LWTR.05T	Light water	1001	800
POLY.01T	Polyethylene	1001	300
H/ZR.01T	^{1}H in ZrHx	1001	300
H/ZR.02T	^{1}H in ZrHx	1001	400
H/ZR.04T	^{1}H in ZrHx	1001	600
H/ZR.05T	^{1}H in ZrHx	1001	800
H/ZR.06T	^{1}H in ZrHx	1001	1200
BENZ.01T	Benzene	1001,6000,6012	300
BENZ.02T	Benzene	1001,6000,6012	400
BENZ.03T	Benzene	1001,6000,6012	500
BENZ.04T	Benzene	1001,6000,6012	600
BENZ.05T	Benzene	1001,6000,6012	800
HWTR.01T	Heavy water	1002	300
HWTR.02T	Heavy water	1002	400
HWTR.03T	Heavy water	1002	500
HWTR.04T	Heavy water	1002	600
HWTR.05T	Heavy water	1002	800
BE.01T	Beryllium metal	4009	300
BE.04T	Beryllium metal	4009	600
BE.05T	Beryllium metal	4009	800
BE.06T	Beryllium metal	4009	1200
BEO.01T	Beryllium oxide	4009,8016	300
BEO.04T	Beryllium oxide	4009,8016	600
BEO.05T	Beryllium oxide	4009,8016	800
BEO.06T	Beryllium oxide	4009,8016	1200

GRPH.01T	Graphite	6000,6012	300
GRPH.04T	Graphite	6000,6012	600
GRPH.05T	Graphite	6000,6012	800
GRPH.06T	Graphite	6000,6012	1200
GRPH.07T	Graphite	6000,6012	1600
GRPH.08T	Graphite	6000,6012	2000
ZR/H.01T	Zr in ZrHx	40000	300
ZR/H.02T	ZR in ZrHx	40000	400
ZR/H.04T	Zr in ZrHx	40000	600
Zr/H.05T	Zr in ZrHx	40000	800
Zr/H.06T	Zr in ZrHx	40000	1200

---

*Isotopes for which the $S(\alpha,\beta)$ data are valid.

III. MCNP CROSS-SECTION LIBRARIES

Discussion of the Monte Carlo cross-section libraries is given in two parts: (1) description of the available libraries, and (2) available visual information to aid in cross-section selection. A list of the library contents then follows.

A. Library Description

Continuous-energy (or pointwise) neutron cross-section data for use in Monte Carlo calculations are available to MCNP from several libraries. The physics of the interactions is contained in these neutron cross-section sets. For each element or nuclide there is a set of numbers detailing (a) through which processes the interactions take place, (b) at which angles scattered neutrons are likely to emerge, and (c) with how much energy the scattered neutrons are likely to emerge. There is also information about photon production data, the spectra of photons produced, and the amount of energy deposited in heating. These numbers are arranged in ACE format described in Appendix F.

The cross-section data depend on incident neutron energy and are tabulated at a number of energy points sufficiently dense that linear-linear interpolation at intermediate energies represents the desired quantity to within certain specified tolerances.

Each cross-section set is generated from an evaluated data set and is then stored with a unique identifier (the ZAID number) on one

of the several libraries. The evaluated data come from many sources: ENDF/B-III, -IV, -V,[46] ENDL-73, -75, -76, -79, -85,[4] AWRE,[47] as well as special Los Alamos evaluations.

Because of the linear-linear interpolation requirement as well as the detail with which the data are given in some evaluations, many of the ACE-formatted cross-section sets can be quite lengthy. For this reason , many (but not all) of the cross-section sets are also available in pseudomultigroup form. The individual cross sections have been averaged over 262 groups (263 energy boundaries) using a flat weight function. For $E_i \leq E < E_{i+1}$, MCNP uses $\sigma(E) = \sigma_i$. In the vernacular we refer to these cross sections as "discrete-reaction cross sections." One great advantage of using the discrete cross sections is that the computer storage required for cross sections is markedly reduced. Reducing storage is particularly important when running in a timesharing environment and obviously when the fully continuous sets are too large for the computer. It should be emphasized that the 262-group treatment applies only to the neutron cross sections; the secondary energy and angular distributions are identical to those on the original pointwise data file from which the discrete-reaction set was generated.

The discrete cross sections are accessed whenever a DRXS input card is used (see page 248). If a ZAID number is listed on the DRXS card, the cross sections are read off the corresponding discrete cross-section library rather than off the continuous-energy library. From the list in Table G.2 of this Appendix, you can determine which continuous-energy cross-section sets have discrete-reaction counterparts.

There are many pointwise and discrete cross-section libraries available in ACE format for MCNP. These libraries are contained on both public and CFS files and are referenced on the cross-section directory XSDIR. It is not necessary for the user to know whether a particular ZAID is on a CFS or public file because the code will automatically access the appropriate library based on the ZAID designated.

In general, cross sections (whether continuous or discrete) based on ENDF/B-V evaluations are believed to be the best available to MCNP users at this time. Therefore, the default cross sections for MCNP are based upon the ENDF/B-V evaluations, although in some cases the ENDF/B-V sets are supplemented by more recent evaluations from T-2 (the Applied Nuclear Science Group). The ENDF cross sections are denoted by the alphanumeric suffixes .50C, .51C, and .50D, which represent the pointwise, thinned-pointwise, and discrete-reaction cross sections, respectively. T-2 cross-section evaluations are denoted by the .55 suffix. The cross-section file denoted by the .50D

suffix is a discrete version of the ENDF continuous-energy cross-section file (.50C). The file identified by the .51C suffix is a thinned version of .50C. It is a continuous-energy cross-section file containing the same set of nuclides as the .50C file but with cross sections and angular distributions specified at far fewer energies. If there are several cross-section sets available for the same nuclide and you are uncertain which one to choose, we advise you initially to select one of the three mentioned above. X-6 can also answer any questions about the various sets and help you with your selection

Files ENDL79 and DRL79 are continuous-energy and discrete sets based on Howerton's 1979 evaluations from Livermore. Howerton has recently updated these files (ENDL85, suffixes .35C and .35D) but many of the nuclides have yet to be processed. Thus the complete data set is currently unavailable to MCNP users.

Several libraries are available that contain older cross-section evaluations. BMCCS is the library that previously was RMCCS when Version 2C was current. Correspondingly, D9 is the previous version of DRMCCS. The files AMCCS, XMCCS, and UMCCS contain miscellaneous evaluations from various sources. Nuclides in these libraries should be used cautiously.

Table G.2 at the end of this Appendix lists all the cross-section sets maintained by X-6 on the standard Monte Carlo neutron libraries as of December 19, 1985. The current version of the list is a computer file called XSLIST and is available under the CFS path /X6XS/CTSS. The first line in XSLIST contains its creation date. If your listing is outdated, XSLIST can be sent to a line printer or to microfiche. If a set is not listed for a nuclide or at a particular temperature you need, contact X-6 and we will try to get it for you. The entries in each of the columns of Table G.2 can be described as follows:

ZAID – the ZAID is the nuclide identification number with the form
ZZZAAA.nnX
where ZZZ is the atomic number,
AAA is the mass number (000 for naturally occurring elements),
nn is the neutron cross-section identifier
X=C for continuous-energy neutron tables
X=D for discrete-reaction tables

FILE – name of one of the cross-section libraries maintained by X-6

SOURCE - indicates where the particular evaluation originated:

ENDF/B (Versions III, IV, and V) is the Evaluated Nuclear Data File,[46] an American effort coordinated by the National Nuclear Data Center at Brookhaven National Laboratory. The evaluations are updated periodically by evaluators from all over the country.

ENDL (73,75,76,79, and 85) is the Evaluated Nuclear Data Library[4] compiled by R. J. Howerton and his group at the Lawrence Livermore National Laboratory. The number indicates from which year's library a particular evaluation was taken.

LAMDF - the Los Alamos Master Data File,[21] vintage 1970. Evaluations in this data file were drawn together from many sources, including Los Alamos, the AWRE at Harwell, England, and the Lawrence Livermore Laboratory.

LASL-SUB - The Los Alamos Sublibrary is a collection of special evaluations prepared by Group T-2 at Los Alamos. These are described in special Los Alamos reports such as Reference 48.

DNA - Defense Nuclear Agency special version of the iron evaluation specified correctly as MAT= 4180 MOD 2.

FOSTER - special evaluations representative of the average fission products for ^{235}U and ^{239}Pu. This work was carried out in Group T-2, Los Alamos, and is described in Reference 7.

WEBSTER - special purpose evaluation for ^{12}C prepared from a translation into ENDF/B format of the ^{12}C data used at the Lawrence Livermore Laboratory by W. W. Webster.

MAT - for ENDF/B, MAT is the material identifier for a particular evaluation. For the ENDL-73, 75, and 76 libraries, a MAT was assigned for each nuclide to be compatible with ENDF/B procedures. ENDL-79 and 85 contain no MAT identifiers. For evaluations from the Los Alamos Master Data File, the SID identifier (Table VI of Reference 21) is listed in the MAT column.

TYPE - CONT indicates a continuous-energy cross-section set; DISC indicates a discrete-reaction cross-section set.

TEMP – the temperature (in Kelvin) at which the data were processed. The temperature enters into the processing only through the Doppler broadening of cross sections. Doppler broadening, in the current context, refers to a change in cross section resulting from thermal motion (translation, rotation, and vibration) of nuclei in a target material. Doppler broadening is done on all cross sections for incident neutron (nonrelativistic energies) on a target at some temperature (TEMP) in which the free-atom approximation is valid.

In general an increase in the temperature of the material containing neutron-absorbing nuclei in a homogeneous system results in Doppler broadening of resonances and an increase in resonance absorption. Furthermore, a constant cross section at zero K goes to 1/v behavior as the temperature increases. You should not only use the best evaluations but also use evaluations that are at temperatures approximating temperatures in your application. Contact X-6 for guidance or generation of specific temperature libraries. All ENDF/B-V evaluations on the standard libraries (that is, pointwise, thinned-pointwise, and discrete) were processed at room temperature (300 K).

GPD "Yes" means that gamma production data exist; "No" means that such data do not exist. Between sets with gamma production data, there is a further distinction represented as follows: "P" indicates sets with a pointwise representation of the energy dependence of the gamma production cross section; "H" indicates that the energy dependence of the gamma production cross section is represented as a histogram over 30 energy groups. The "H" type of representation is found on cross-section sets processed before 1977, approximately. Before then, gamma production information was incorporated into MCNP calculations using gamma production matrices calculated in a 30-neutron group and 12-gamma group structure. When the ACE format was changed, the total cross section was calculated from the matrices and used in histogram form so old problems could track. In the case of discrete cross-section sets in Table G.2, the "P" or "H" refers to the type of data in the original file before processing into discrete form. In many older evaluations gamma production data frequently was not included. Thus, gamma production matrices from other quite different sources might be used with the neutron cross-section data. In Table G.2 the asterisk indicates those data sets for which the gamma production data source differs from the source of the neutron cross-section data listed in the column labeled SOURCE. Those materials that have photon production information given in expanded ACE format (see Version 2D Newsletter) are indicated by the notation (E).

LENGTH - the total length of a particular cross-section file in decimal. It is understood that the actual storage requirement in an MCNP problem will often be less because certain data unneeded for a problem will be deleted.

NUBAR - for fissionable material, NUBAR indicates the type of fission nubar data available. PROMPT means that only prompt nubar data are given; TOTAL means that only total nubar data are given; BOTH means that prompt and total nubar are given.

ARROW - the arrow, which precedes certain nuclides listed in the table, is a new feature and indicates what we believe to be the best available representation or evaluation of a specific nuclide when more than one evaluation is listed. You should note that because of size limitations on public files the evaluation designated by the arrow may not be the default evaluation.

Finally, you may introduce a cross-section library of your own by using the XS input card.

B. VISUAL INFORMATION

To assist in cross-section selection, microfiche cards are available that display in Hollerith the BCD information for all isotopes in the Monte Carlo libraries. XSLIST is an index of the cross sections that are on each of the libraries and follows immediately.

Graphic information for some older cross-section libraries is available in References 52 and 53 where graphs of the cross sections for all reactions are given. Plotting programs are available to compare the cross-section data of the original evaluations (such as ENDF/B or ENDL) and the ACE representation thereof. Other plotting packages are being developed to help the MCNP user in cross-section selection.

Table G.2
Cross Sections Maintained by X-6
as of 4/15/86

ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
z = 1 ************** hydrogen **************								
** h-1 **								
1001.01c	xmccs	endl-73	501	cont	0	yes h*	3250	
1001.02c	xmccs	lamdf	5990	cont	0	yes h*	2011	
1001.03c	xmccs	endf/b-iii	1148	cont	0	yes h	2459	
1001.04c	bmccs	endf/b-iv	1269	cont	0	yes p	2459	
1001.04d	d9	endf/b-iv	1269	disc	0	yes p	2914	
1001.30c	amccs	endl-76	7101	cont	0	yes p	1444	
1001.31c	endl79	endl-79		cont	0	no	2496	
1001.31d	drl79	endl-79		disc	0	no	2082	
——> 1001.50c	rmccs	endf/b-v	1301	cont	300	yes p(e)	2827	
1001.50d	drmccs	endf/b-v	1301	disc	300	yes p(e)	3236	
1001.51c	endf5t	endf/b-v	1301	cont	300	yes p(e)	2827	
** h-2 **								
1002.01c	xmccs	endl-73	502	cont	0	yes h*	2122	
1002.02c	bmccs	lamdf	2254	cont	0	yes h*	3007	
1002.02d	d9	lamdf	2254	disc	0	yes h*	4172	
1002.03c	xmccs	enbf/b-iii	1120	cont	0	yes h*	1963	
1002.04c	amccs	endf/b-iv	1120	cont	0	yes p	2144	
1002.31c	endl79	endl-79		cont	0	no	1926	
1002.31d	drl79	endl-79		disc	0	no	2630	
1002.50c	endf5p	endf/b-v	1302	cont	300	yes p(e)	4048	
1002.50d	dre5	endf/b-v	1302	disc	300	yes p(e)	4747	
1002.51c	endf5t	endf/b-v	1302	cont	300	yes p(e)	3978	
——> 1002.55c	rmccs	group t-2	120	cont	300	yes p(e)	6042	
1002.55d	drmccs	group t-2	120	disc	300	yes p(e)	5404	
** h-3 **								
1003.01c	xmccs	endl-73	503	cont	0	no	1111	
1003.02c	umccs	lamdf	2252	cont	0	no	1483	
1003.03c	bmccs	endf/b-iv	1169	cont	300	no	2114	
1003.03d	d9	endf/b-iv	1169	disc	300	no	2702	
1003.30c	amccs	endl-76	7103	cont	0	no	1026	
1003.31c	endl79	endl-79		cont	0	no	1361	
1003.31d	drl79	endl-79		disc	0	no	2259	
——> 1003.50c	rmccs	endf/b-v	1169	cont	300	no	2469	
1003.50d	drmccs	endf/b-v	1169	disc	300	no	2848	
1003.51c	endf5t	endf/b-v	1169	cont	300	no	2434	

ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
z = 2 ************** helium **								
** he-nat **								
2000.01c	bmccs	endf/b-iii	1088	cont	0	no	1705	
2000.01d	d9	endf/b-iii	1088	disc	0	no	2108	
** he-3 **								
2003.01c	xmccs	endl-73	504	cont	0	no	1320	
2003.02c	xmccs	landf	7220	cont	0	no	818	
2003.03c	bmccs	endf/b-iii	1146	cont	0	no	1517	
2003.03d	d9	endf/b-iii	1146	disc	0	no	2048	
2003.31c	endl79	endl-79		cont	0	no	1743	
2003.31d	drl79	endl-79		disc	0	no	2313	
——> 2003.50c	rmccs	endf/b-v	1146	cont	300	no	2361	
2003.50d	drmccs	endf/b-v	1146	disc	300	no	2653	
2003.51c	endf5t	endf/b-v	1146	cont	300	no	2361	
** he-4 **								
2004.01c	umccs	endl-73	505	cont	0	no	1217	
2004.02c	xmccs	landf	22	cont	0	no	1162	
2004.03c	bmccs	endf/b-iv	1270	cont	300	no	2407	
2004.03d	d9	endf/b-iv	1270	disc	300	no	2202	
2004.30c	amccs	endl-76	7105	cont	0	no	1267	
2004.31c	endl79	endl-79		cont	0	no	1548	
2004.31d	drl79	endl-79		disc	0	no	2408	
——> 2004.50c	rmccs	endf/b-v	1270	cont	300	no	3102	
2004.50d	drmccs	endf/b-v	1270	disc	300	no	2692	
2004.51c	endf5t	endf/b-v	1270	cont	300	no	2682	
z = 3 ************** lithium **								
** li-6 **								
3006.01c	xmccs	landf	2214	cont	0	yes h*	3443	
3006.02c	xmccs	endf/b-iii	1115	cont	0	yes h	3997	
3006.04c	xmccs	endf/b-iv	1271	cont	0	yes p	4615	
3006.10c	bmccs	lasl-sub	101	cont	0	yes p	8294	
3006.10d	d9	lasl-sub	101	disc	0	yes p	6742	
3006.30c	xmccs	endl-76	7106	cont	0	yes p	2940	
3006.31c	endl79	endl-79		cont	0	no	5928	
3006.31d	drl79	endl-79		disc	0	no	5012	
——> 3006.50c	rmccs	endf/b-v	1303	cont	300	yes p(e)	9993	
3006.50d	drmccs	endf/b-v	1303	disc	300	yes p(e)	8777	
3006.51c	endf5t	endf/b-v	1303	cont	300	yes p(e)	9196	

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
** li-7 **									
	3007.01c	xmccs	endl-73	507	cont	0	yes h*	2787	
	3007.02c	xmccs	lamdf	2215	cont	0	yes h*	3097	
	3007.03c	xmccs	endf/b-iii	1116	cont	0	yes h	3562	
	3007.05c	bmccs	endf/b-iv	1272	cont	0	yes p	3751	
	3007.05d	d9	endf/b-iv	1272	disc	0	yes p	3801	
	3007.31c	endl79	endl-79		cont	0	no	2721	
	3007.31d	drl79	endl-79		disc	0	no	3215	
	3007.50c	endf5p	endf/b-v	1272	cont	300	yes p(e)	4925	
	3007.50d	dre5	endf/b-v	1272	disc	300	yes p(e)	4996	
	3007.51c	endf5t	endf/b-v	1272	cont	300	yes p(e)	4925	
——>	3007.55c	rmccs	group t-2	3007	cont	300	yes p(e)	13232	
	3007.55d	drmccs	group t-2	3007	disc	300	yes p(e)	12708	

z = 4 ************** beryllium **

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
** be-9 **									
	4009.01c	xmccs	endl-73	509	cont	0	yes h*	3843	
	4009.02c	umccs	endf/b-iii	1154	cont	0	yes h*	3321	
	4009.03c	bmccs	lasl-sub	104	cont	300	yes p	7885	
	4009.03d	d9	lasl-sub	104	disc	300	yes p	6622	
	4009.31c	endl79	endl-79		cont	0	no	7737	
	4009.31d	drl79	endl-79		disc	0	no	7329	
——>	4009.50c	rmccs	endf/b-v	1304	cont	300	yes p(e)	8947	
	4009.50d	drmccs	endf/b-v	1304	disc	300	yes p(e)	8817	
	4009.51c	endf5t	endf/b-v	1304	cont	300	yes p(e)	8073	

z = 5 ************** boron **

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
** b-nat **									
	5000.01c	bmccs	endl-73	510	cont	0	yes h*	5058	
	5000.01d	d9	endl-73	510	disc	0	yes h*	4151	
** b-10 **									
	5010.01c	xmccs	endl-73	511	cont	0	yes h*	5359	
	5010.02c	amccs	endf/b-iii	1155	cont	0	yes h*	2589	
	5010.03c	bmccs	endf/b-iv	1273	cont	0	yes p	9241	
	5010.03d	d9	endf/b-iv	1273	disc	0	yes p	5720	
	5010.31c	endl79	endl-79		cont	0	no	5180	
	5010.31d	drl79	endl-79		disc	0	no	3902	
——>	5010.50c	rmccs	endf/b-v	1305	cont	300	yes p(e)	20261	
	5010.50d	drmccs	endf/b-v	1305	disc	300	yes p(e)	12383	

ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
5010.51c	endf5t	endf/b-v	1305	cont	300	yes p(e)	18886	
** b-11 **								
5011.01c	xmccs	endl-73	565	cont	0	no	1516	
5011.02c	bmccs	endf/b-iii	1160	cont	0	no	5134	
5011.02d	d9	endf/b-iii	1160	disc	0	no	2522	
5011.31c	endl79	endl-79		cont	0	no	3024	
5011.31d	drl79	endl-79		disc	0	no	3946	
5011.50c	endf5p	endf/b-v	1160	cont	300	no	4385	
5011.50d	dre5	endf/b-v	1160	disc	300	no	2853	
5011.51c	endf5t	endf/b-v	1160	cont	300	no	4193	
---> 5011.55c	rmccsa	group t-2	5011	cont	300	yes p(e)	12315	
5011.55d	drmccs	group t-2	5011	disc	300	yes p(e)	7167	

z = 6 ************** carbon ***

ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
** c-nat **								
---> 6000.50c	rmccs	endf/b-v	1306	cont	300	yes p(e)	23387	
6000.50d	drmccs	endf/b-v	1306	disc	300	yes p(e)	16905	
6000.51c	endf5t	endf/b-v	1306	cont	300	yes p(e)	23067	
** c-12 **								
6012.01c	xmccs	landf	3006	cont	0	yes h*	3098	
6012.02c	xmccs	endf/b-iii	1165	cont	0	yes h*	6942	
6012.03c	xmccs	endf/b-iv	1274	cont	0	yes p	7567	
6012.04c	xmccs	webster	6965	cont	0	no	2430	
6012.10c	bmccs	lasl-sub	102	cont	0	yes p(e)	12102	
6012.10d	d9	lasl-sub	102	disc	0	yes p(e)	9832	
6012.30c	xmccs	endl-76	7112	cont	0	yes p	2815	
6012.31c	endl79	endl-79		cont	0	no	3610	
6012.31d	drl79	endl-79		disc	0	no	3583	
---> 6012.50c	rmccs	endf/b-v	1306	cont	300	yes p(e)	23387	
6012.50d	drmccs	endf/b-v	1306	disc	300	yes p(e)	16905	

z = 7 ************** nitrogen **

ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
** n-14 **								
7014.01c	xmccs	endl-73	513	cont	0	yes h*	9586	
7014.02c	xmccs	landf	5513	cont	0	yes h*	9211	
7014.04c	bmccs	endf/b-iv	1275	cont	0	yes p	21553	
7014.04d	d9	endf/b-iv	1275	disc	0	yes p	9874	
7014.30c	amccs	endl-76	7113	cont	0	yes p	5148	
---> 7014.50c	rmccs	endf/b-v	1275	cont	300	yes p(e)	45518	

ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
7014.50d	drmccs	endf/b-v	1275	disc	300	yes p(e)	26854	
7014.51c	endf5t	endf/b-v	1275	cont	300	yes p(e)	45397	
** n-15 **								
---> 7015.55c	rmccsa	group t-2	9993	cont	300	yes p(e)	20981	
7015.55d	drmccs	group t-2	9993	disc	300	yes p(e)	15334	
z = 8 ************** oxygen **								
** o-16 **								
8016.01c	xmccs	endl-73	514	cont	0	yes h*	5540	
8016.02c	xmccs	landf	5514	cont	0	yes h*	4791	
8016.04c	bmccs	endf/b-iv	1276	cont	0	yes p	21823	
8016.04d	d9	endf/b-iv	1276	disc	0	yes p	10486	
8016.30c	amccs	endl-76	7114	cont	0	yes p	4565	
8016.30d	d9	endl-76	7114	disc	0	yes p	4739	
---> 8016.50c	rmccs	endf/b-v	1276	cont	300	yes p(e)	38003	
8016.50d	drmccs	endf/b-v	1276	disc	300	yes p(e)	20516	
8016.51c	endf5t	endf/b-v	1276	cont	300	yes p(e)	37996	
z = 9 ************** fluorine **								
** f-19 **								
9019.01c	amccs	endl-73	515	cont	0	yes h*	3101	
9019.01d	d9	endl-73	515	disc	0	yes h*	4216	
9019.02c	xmccs	endf/b-iv	1277	cont	0	yes h	26334	
9019.03c	bmccs	endf/b-iv	1277	cont	0	yes p	24464	
9019.03d	d9	endf/b-iv	1277	disc	0	yes p	8926	
9019.31c	endl79	endl-79		cont	0	no	23368	
9019.31d	drl79	endl-79		disc	0	no	9764	
---> 9019.50c	endf5p	endf/b-v	1309	cont	300	yes p(e)	44191	
9019.50d	dre5	endf/b-v	1309	disc	300	yes p(e)	23217	
9019.51c	rmccs	endf/b-v	1309	cont	300	yes p(e)	41503	
9019.51d	drmccs	endf/b-v	1309	disc	300	yes p(e)	23217	
z = 11 ************** sodium **								
** na-23 **								
11023.01c	bmccs	endl-73	516	cont	0	yes h*	6816	
11023.01d	d9	endl-73	516	disc	0	yes h*	5094	
11023.31c	endl79	endl-79		cont	0	no	14656	

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
	11023.31d	dr179	endl-79		disc	0	no	4299	
--->	11023.50c	endf5p	endf/b-v	1311	cont	300	yes p(e)	52313	
	11023.50d	dre5	endf/b-v	1311	disc	300	yes p(e)	41726	
	11023.51c	mmccs	endf/b-v	1311	cont	300	yes p(e)	48924	
	11023.51d	dmmccs	endf/b-v	1311	disc	300	yes p(e)	41726	

z = 12 ************** magnesium ***

** mg-nat **

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
	12000.01c	xmccs	endl-73	517	cont	0	no	5137	
	12000.02c	bmccs	endf/b-iii	1014	cont	0	no	3771	
	12000.02d	d9	endf/b-iii	1014	disc	0	no	2874	
	12000.31c	endl79	endl-79		cont	0	no	6199	
	12000.31d	dr179	endl-79		disc	0	no	3626	
--->	12000.50c	endf5u	endf/b-v	1312	cont	300	yes p(e)	56395	
	12000.50d	dre5	endf/b-v	1312	disc	300	yes p(e)	14131	
	12000.51c	mmccs	endf/b-v	1312	cont	300	yes p(e)	48978	
	12000.51d	dmmccs	endf/b-v	1312	disc	300	yes p(e)	14131	

z = 13 ************** aluminum ***

** al-27 **

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
	13027.01c	amccs	endl-73	518	cont	0	yes h*	3915	
	13027.02c	xmccs	landf	2035	cont	0	yes h*	5532	
	13027.03c	xmccs	endf/b-iii	1135	cont	0	yes h*	18038	
	13027.04c	bmccs	endf/b-iv	1193	cont	0	yes p	32517	
	13027.04d	d9	endf/b-iv	1193	disc	0	yes p	9700	
	13027.31c	endl79	endl-79		cont	0	no	27295	
	13027.31d	dr179	endl-79		disc	0	no	9871	
--->	13027.50c	mmccs	endf/b-v	1313	cont	300	yes p(e)	54223	
	13027.50d	dmmccs	endf/b-v	1313	disc	300	yes p(e)	42008	
	13027.51c	endf5t	endf/b-v	1313	cont	300	yes p(e)	53438	

z = 14 ************** silicon ***

** si-nat **

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
	14000.01c	amccs	endl-73	519	cont	0	yes h*	12371	
	14000.01d	d9	endl-73	519	disc	0	yes h*	4631	
	14000.02c	bmccs	endf/b-iii	1151	cont	0	yes h*	21632	
	14000.02d	d9	endf/b-iii	1151	disc	0	yes h*	6973	

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
	14000.31c	endl79	endl-79		cont	0	no	11924	
	14000.31d	drl79	endl-79		disc	0	no	3955	
--->	14000.50c	endf5p	endf/b-v	1314	cont	300	yes p(e)	98670	
	14000.50d	dre5	endf/b-v	1314	disc	300	yes p(e)	69559	
	14000.51c	rmccs	endf/b-v	1314	cont	300	yes p(e)	88190	
	14000.51d	drmccs	endf/b-v	1314	disc	300	yes p(e)	69559	

z = 15 ************** phosphorus ***

** p-31 **

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
	15031.01c	bmccs	endl-73	520	cont	0	no	2842	
	15031.01d	d9	endl-73	520	disc	0	no	3370	
	15031.31c	endl79	endl-79		cont	0	no	3637	
	15031.31d	drl79	endl-79		disc	0	no	3423	
--->	15031.50c	endf5u	endf/b-v	1315	cont	300	yes p(e)	5794	
	15031.50d	dre5	endf/b-v	1315	disc	300	yes p(e)	5822	
	15031.51c	rmccs	endf/b-v	1315	cont	300	yes p(e)	5793	
	15031.51d	drmccs	endf/b-v	1315	disc	300	yes p(e)	5822	

z = 16 ************** sulfur ***

** s-32 **

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
	16032.01c	bmccs	endl-73	521	cont	0	no	3252	
	16032.01d	d9	endl-73	521	disc	0	no	3211	
	16032.31c	endl79	endl-79		cont	0	no	4071	
	16032.31d	drl79	endl-79		disc	0	no	3341	
--->	16032.50c	endf5u	endf/b-v	1316	cont	300	yes p(e)	6850	
	16032.50d	dre5	endf/b-v	1316	disc	300	yes p(e)	6363	
	16032.51c	rmccs	endf/b-v	1316	cont	300	yes p(e)	6841	
	16032.51d	drmccs	endf/b-v	1316	disc	300	yes p(e)	6363	

z = 17 ************** chlorine ***

** cl-nat **

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
	17000.01c	amccs	endl-73	522	cont	0	yes h*	9854	
	17000.02c	bmccs	endf/b-iii	1149	cont	0	yes h	38371	
	17000.02d	d9	endf/b-iii	1149	disc	0	yes h	8237	
	17000.31c	endl79	endl-79		cont	0	no	8664	
	17000.31d	drl79	endl-79		disc	0	no	3430	
--->	17000.50c	endf5p	endf/b-v	1149	cont	300	yes p(e)	23374	

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
	17000.50d	dre5	endf/b-v	1149	disc	300	yes p(e)	18270	
	17000.51c	rmccs	endf/b-v	1149	cont	300	yes p(e)	21145	
	17000.51d	drmccs	endf/b-v	1149	disc	300	yes p(e)	18270	
z = 18 ************** argon **********									
** ar-nat **									
	18000.01c	bmccs	endl-73	523	cont	0	no	2029	
	18000.01d	d9	endl-73	523	disc	0	no	2875	
	18000.31c	endl79	endl-79		cont	0	no	2857	
	18000.31d	drl79	endl-79		disc	0	no	2939	
——>	18000.35c	rmccsa	endl-85		cont	0	yes p(e)	5646	
	18000.35d	drmccs	endl-85		disc	0	yes p(e)	14764	
z = 19 ************** potassium **********									
** k-nat **									
	19000.01c	bmccs	endl-73	524	cont	0	yes h*	7436	
	19000.01d	d9	endl-73	524	disc	0	yes h*	4130	
	19000.31c	endl79	endl-79		cont	0	no	6645	
	19000.31d	drl79	endl-79		disc	0	no	3293	
——>	19000.50c	endf5u	endf/b-v	1150	cont	300	yes p(e)	22112	
	19000.50d	dre5	endf/b-v	1150	disc	300	yes p(e)	23198	
	19000.51c	rmccs	endf/b-v	1150	cont	300	yes p(e)	18859	
	19000.51d	drmccs	endf/b-v	1150	disc	300	yes p(e)	23198	
z = 20 ************** calcium **********									
** ca-nat **									
	20000.01c	amccs	endl-73	525	cont	0	yes h*	9623	
	20000.01d	d9	endl-73	525	disc	0	yes h*	4457	
	20000.10c	bmccs	endf/b-iv	1195	cont	0	yes p	24085	
	20000.10d	d9	endf/b-iv	1195	disc	0	yes p	9198	
	20000.31c	endl79	endl-79		cont	0	no	8959	
	20000.31d	drl79	endl-79		disc	0	no	3637	
——>	20000.50c	endf5u	endf/b-v	1320	cont	300	yes p(e)	62685	
	20000.50d	dre5	endf/b-v	1320	disc	300	yes p(e)	29094	
	20000.51c	rmccs	endf/b-v	1320	cont	300	yes p(e)	53433	
	20000.51d	drmccs	endf/b-v	1320	disc	300	yes p(e)	29094	

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD		LENGTH	NUBAR
z = 22 ************** titanium ***										
** ti-nat **										
	22000.01c	amccs	endl-73	526	cont	0	no		9259	
	22000.01d	d9	endl-73	526	disc	0	no		3058	
	22000.02c	xmccs	lamdf	2190	cont	0	no		3884	
	22000.11c	bmccs	endf/b-iv	1286	cont	300	yes	p	10644	
	22000.11d	d9	endf/b-iv	1286	disc	300	yes	p	3897	
	22000.31c	endl79	endl-79		cont	0	no		9626	
	22000.31d	drl79	endl-79		disc	0	no		3205	
--->	22000.50c	endf5u	endf/b-v	1322	cont	300	yes	p(e)	54862	
	22000.50d	dre5	endf/b-v	1322	disc	300	yes	p(e)	10514	
	22000.51c	rmccs	endf/b-v	1322	cont	300	yes	p(e)	31893	
	22000.51d	drmccs	endf/b-v	1322	disc	300	yes	p(e)	10514	
z = 23 ************** vanadium ***										
** v-nat **										
	23000.30c	bmccs	endf/b-iv	1196	cont	0	yes	p	6456	
	23000.30d	d9	endf/b-iv	1196	disc	0	yes	p	4603	
--->	23000.50c	endf5u	endf/b-v	1323	cont	300	yes	p(e)	38373	
	23000.50d	dre5	endf/b-v	1323	disc	300	yes	p(e)	8929	
	23000.51c	rmccs	endf/b-v	1323	cont	300	yes	p(e)	34171	
	23000.51d	drmccs	endf/b-v	1323	disc	300	yes	p(e)	8929	
** v-51 **										
	23051.31c	endl79	endl-79		cont	0	no		21394	
	23051.31d	drl79	endl-79		disc	0	no		5505	
z = 24 ************** chromium ***										
** cr-nat **										
	24000.01c	amccs	lamdf	2045	cont	0	no		3667	
	24000.01d	d9	lamdf	2045	disc	0	no		3554	
	24000.11c	bmccs	endf/b-iv	1191	cont	300	yes	p	38240	
	24000.11d	d9	endf/b-iv	1191	disc	300	yes	p	11767	
	24000.12c	xmccs	endf/b-iv	1191	cont	900	yes	p	51663	
	24000.31c	endl79	endl-79		cont	0	no		5827	
	24000.31d	drl79	endl-79		disc	0	no		5260	
--->	24000.50c	rmccs	endf/b-v	1324	cont	300	yes	p(e)	134515	

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
	24000.50d	drmccs	endf/b-v	1324	disc	300	yes p(e)	30775	
	24000.51c	endf5t	endf/b-v	1324	cont	300	yes p(e)	55677	
z = 25	************** manganese **								
	** mn-55 **								
	25055.01c	bmccs	endl-73	527	cont	0	no	3586	
	25055.01d	d9	endl-73	527	disc	0	no	3099	
	25055.31c	endl79	endl-79		cont	0	no	4149	
	25055.31d	drl79	endl-79		disc	0	no	3186	
——>	25055.50c	endf5u	endf/b-v	1325	cont	300	yes p(e)	105154	
	25055.50d	dre5	endf/b-v	1325	disc	300	yes p(e)	9742	
	25055.51c	rmccs	endf/b-v	1325	cont	300	yes p(e)	25788	
	25055.51d	drmccs	endf/b-v	1325	disc	300	yes p(e)	9742	
z = 26	************** iron ***								
	** fe-nat **								
	26000.01c	xmccs	endl-73	528	cont	0	yes h*	4102	
	26000.03c	xmccs	dna	4180	cont	0	yes h*	62886	
	26000.11c	bmccs	endf/b-iv	1192	cont	300	yes p	54104	
	26000.11d	d9	endf/b-iv	1192	disc	300	yes p	8852	
	26000.12c	xmccs	endf/b-iv	1192	cont	900	yes p	57638	
	26000.30c	amccs	endl-76	7132	cont	0	yes p	23179	
	26000.30d	d9	endl-76	7132	disc	0	yes p	4079	
	26000.31c	endl79	endl-79		cont	0	no	23960	
	26000.31d	drl79	endl-79		disc	0	no	3339	
	26000.50c	endf5p	endf/b-v	1326	cont	300	yes p(e)	115508	
	26000.50d	dre5	endf/b-v	1326	disc	300	yes p(e)	33957	
	26000.51c	endf5t	endf/b-v	1326	cont	300	yes p(e)	78377	
——>	26000.55c	rmccs	group t-2	260	cont	300	yes p(e)	178453	
	26000.55d	drmccs	group t-2	260	disc	300	yes p(e)	72693	
z = 27	************** cobalt ***								
	** co-59 **								
	27059.31c	endl79	endl-79		cont	0	no	31707	
	27059.31d	drl79	endl-79		disc	0	no	6391	
——>	27059.50c	endf5u	endf/b-v	1327	cont	300	yes p(e)	117136	
	27059.50d	dre5	endf/b-v	1327	disc	300	yes p(e)	11830	

ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
27059.51c	rmccs	endf/b-v	1327	cont	300	yes p(e)	28416	
27059.51d	drmccs	endf/b-v	1327	disc	300	yes p(e)	11830	
z = 28 ************** nickel **								
** ni-nat **								
28000.01c	amccs	lamdf	2046	cont	0	no	5714	
28000.01d	d9	lamdf	2046	disc	0	no	3376	
28000.11c	bmccs	endf/b-iv	1190	cont	300	yes p	35192	
28000.11d	d9	endf/b-iv	1190	disc	300	yes p	5658	
28000.12c	xmccs	endf/b-iv	1190	cont	900	yes p	40842	
28000.31c	endl79	endl-79		cont	0	no	32964	
28000.31d	drl79	endl-79		disc	0	no	8096	
---> 28000.50c	rmccs	endf/b-v	1328	cont	300	yes p(e)	139974	
28000.50d	drmccs	endf/b-v	1328	disc	300	yes p(e)	22059	
28000.51c	endf5t	endf/b-v	1328	cont	300	yes p(e)	93636	
** ni-58 **								
28058.01c	bmccs	endl-73	529	cont	0	no	5347	
28058.01d	d9	endl-73	529	disc	0	no	3529	
z = 29 ************** copper **								
** cu-nat **								
29000.01c	amccs	endl-73	530	cont	0	yes h*	3629	
29000.01d	d9	endl-73	530	disc	0	yes h*	4427	
29000.02c	xmccs	lamdf	2249	cont	0	yes h*	6476	
29000.10c	bmccs	endf/b-iv	1295	cont	0	yes p	14703	
29000.10d	d9	endf/b-iv	1295	disc	0	yes p	8610	
29000.31c	endl79	endl-79		cont	0	no	4157	
29000.31d	drl79	endl-79		disc	0	no	4066	
---> 29000.50c	rmccs	endf/b-v	1329	cont	300	yes p(e)	51911	
29000.50d	drmccs	endf/b-v	1329	disc	300	yes p(e)	12838	
29000.51c	endf5t	endf/b-v	1329	cont	300	yes p(e)	51375	
z = 31 ************** gallium **								
** ga-nat **								
31000.01c	bmccs	endl-73	531	cont	0	no	3730	
31000.01d	d9	endl-73	531	disc	0	no	2907	
31000.31c	endl79	endl-79		cont	0	no	4225	

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
	31000.31d	dr179	endl-79		disc	0	no	3027	
--->	31000.50c	rmccs	endf/b-v	1358	cont	300	yes p(e)	7989	
	31000.50d	drmccs	endf/b-v	1358	disc	300	yes p(e)	6272	
	31000.51c	endf5t	endf/b-v	1358	cont	300	yes p(e)	7989	

z = 33 ************** arsenic ***

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
**	as-75 **								
--->	33075.35c	rmccsa	endl-85		cont	0	yes p(e)	50992	
	33075.35d	drmccs	endl-85		disc	0	yes p(e)	8541	

z = 36 ************** krypton ***

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
**	kr-78 **								
--->	36078.50c	rmccsa	endf/b-v	1330	cont	300	no	9098	
	36078.50d	drmccs	endf/b-v	1330	disc	300	no	4399	
**	kr-80 **								
--->	36080.50c	rmccsa	endf/b-v	1331	cont	300	no	10206	
	36080.50d	drmccs	endf/b-v	1331	disc	300	no	4317	
**	kr-82 **								
--->	36082.50c	rmccsa	endf/b-v	1332	cont	300	no	7261	
	36082.50d	drmccs	endf/b-v	1332	disc	300	no	4307	
**	kr-83 **								
--->	36083.50c	rmccsa	endf/b-v	1333	cont	300	no	8119	
	36083.50d	drmccs	endf/b-v	1333	disc	300	no	4400	
**	kr-84 **								
--->	36084.50c	rmccsa	endf/b-v	1334	cont	300	no	9405	
	36084.50d	drmccs	endf/b-v	1334	disc	300	no	4504	
**	kr-86 **								
--->	36086.50c	rmccsa	endf/b-v	1336	cont	300	no	10457	
	36086.50d	drmccs	endf/b-v	1336	disc	300	no	4342	

z = 39 ************** yttrium ***

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
**	y-89 **								
	39089.35c	miscxs	endl-85		cont	0	yes p(e)	49946	
--->	39089.50c	endf5u	endf/b-v	9202	cont	300	no	18672	
	39089.50d	dre5	endf/b-v	9202	disc	300	no	2352	

APPENDIX G

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
z = 40 ************** zirconium **									
** zr-nat **									
	40000.01c	amccs	landf	2009	cont	0	no	3837	
	40000.01d	d9	landf	2009	disc	0	no	2009	
	40000.02c	bmccs	endl-75	7141	cont	0	yes h	10312	
	40000.02d	d9	endl-75	7141	disc	0	yes h	4311	
	40000.31c	endl79	endl-79		cont	0	no	10585	
	40000.31d	drl79	endl-79		disc	0	no	4749	
--->	40000.50c	endf5p	endf/b-v	1340	cont	300	no	52105	
	40000.50d	dre5	endf/b-v	1340	disc	300	no	5441	
	40000.51c	mmccs	endf/b-v	1340	cont	300	no	16857	
	40000.51d	drmccs	endf/b-v	1340	disc	300	no	5441	
z = 41 ************** niobium **									
** nb-93 **									
	41093.01c	amccs	endl-73	532	cont	0	no	5880	
	41093.01d	d9	endl-73	532	disc	0	no	3270	
	41093.30c	bmccs	endl-76	7143	cont	0	yes p	29725	
	41093.30d	d9	endl-76	7143	disc	0	yes p	5905	
	41093.31c	endl79	endl-79		cont	0	no	41316	
	41093.31d	drl79	endl-79		disc	0	no	6609	
--->	41093.50c	endf5p	endf/b-v	1189	cont	300	yes p(e)	129021	
	41093.50d	dre5	endf/b-v	1189	disc	300	yes p(e)	10393	
	41093.51c	mmccs	endf/b-v	1189	cont	300	yes p(e)	14736	
	41093.51d	drmccs	endf/b-v	1189	disc	300	yes p(e)	10393	
z = 42 ************** molybdenum ***									
** mo-nat **									
	42000.01c	bmccs	endl-73	533	cont	0	yes h*	5714	
	42000.01d	d9	endl-73	533	disc	0	yes h*	3950	
	42000.31c	endl79	endl-79		cont	0	no	4949	
	42000.31d	drl79	endl-79		disc	0	no	3112	
--->	42000.50c	endf5u	endf/b-v	1321	cont	300	yes p(e)	35695	
	42000.50d	dre5	endf/b-v	1321	disc	300	yes p(e)	7815	
	42000.51c	mmccs	endf/b-v	1321	cont	300	yes p(e)	10200	
	42000.51d	drmccs	endf/b-v	1321	disc	300	yes p(e)	7815	

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
z = 45	************** rhodium **************								
** rh-103 **									
--->	45103.50c	rmccsa	endf/b-v	1310	cont	300	no	18911	
	45103.50d	drmccs	endf/b-v	1310	disc	300	no	4704	
z = 45	********* average fission product from uranium-235 *********								
** u-235 fp **									
--->	45117.90c	rmccs	foster	998	cont	300	yes p(e)	10375	
	45117.90d	drmccs	foster	998	disc	300	yes p(e)	9568	
z = 46	********* average fission product from plutonium-239 *********								
** pu-239 fp **									
--->	46119.90c	rmccs	foster	999	cont	300	yes p(e)	10505	
	46119.90d	drmccs	foster	999	disc	300	yes p(e)	9603	
z = 47	************** silver **************								
** ag-nat **									
--->	47000.55c	rmccsa	group t-2	47	cont	300	yes p(e)	29153	
	47000.55d	drmccs	group t-2	47	disc	300	yes p(e)	12470	
** ag-107 **									
--->	47107.50c	rmccsa	endf/b-v	1371	cont	300	no	12152	
	47107.50d	drmccs	endf/b-v	1371	disc	300	no	4124	
** ag-109 **									
--->	47109.50c	rmccsa	endf/b-v	1373	cont	300	no	14626	
	47109.50d	drmccs	endf/b-v	1373	disc	300	no	3864	
z = 48	************** cadmium **************								
** cd-nat **									
	48000.01c	bmccs	endl-73	534	cont	0	no	7690	
	48000.01d	d9	endl-73	534	disc	0	no	2971	
	48000.31c	endl79	endl-79		cont	0	no	8131	
	48000.31d	drl79	endl-79		disc	0	no	3031	
--->	48000.50c	endf5u	endf/b-v	1281	cont	300	no	19755	
	48000.50d	dre5	endf/b-v	1281	disc	300	no	3067	

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
	48000.51c	mmccs	endf/b-v	1281	cont	300	no	6775	
	48000.51d	drmccs	endf/b-v	1281	disc	300	no	3067	
z = 50	************** tin	**							
	** sn-nat **								
--->	50000.01c	bmccs	endl-73	535	cont	0	no	2332	
	50000.01d	d9	endl-73	535	disc	0	no	3125	
	50000.31c	endl79	endl-79		cont	0	no	2876	
	50000.31d	drl79	endl-79		disc	0	no	3256	
z = 50	********* fission products	**							
	** ave fp **								
--->	50120.35c	mmccs	endl-85		cont	0	yes p(e)	8427	
	50120.35d	drmccs	endl-85		disc	0	yes p(e)	9024	
	50999.02c	bmccs	endl-73	558	cont	0	no	1647	
	50999.02d	d9	endl-73	558	disc	0	no	2840	
z = 56	************** barium	**							
	** ba-138 **								
	56138.01c	bmccs	endl-73	536	cont	0	no	2606	
	56138.01d	d9	endl-73	536	disc	0	no	3201	
	56138.31c	endl79	endl-79		cont	0	no	3059	
	56138.31d	drl79	endl-79		disc	0	no	3218	
--->	56138.50c	mmccs	endf/b-v	1353	cont	300	yes p(e)	6079	
	56138.50d	drmccs	endf/b-v	1353	disc	300	yes p(e)	6381	
	56138.51c	endf5t	endf/b-v	1353	cont	300	yes p(e)	6064	
z = 62	************** samarium	**							
	** sm-149 **								
--->	62149.50c	endf5u	endf/b-v	1319	cont	300	no	15703	
	62149.50d	dre5	endf/b-v	1319	disc	300	no	4470	

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
z = 63 ************** europium **									
** eu-nat **									
	63000.01c	bmccs	endl-73	537	cont	0	no	3133	
	63000.01d	d9	endl-73	537	disc	0	no	3017	
	63000.31c	endl79	endl-79		cont	0	no	3532	
	63000.31d	drl79	endl-79		disc	0	no	3039	
--->	63000.35c	mccsa	endl-85		cont	0	yes p(e)	6987	
	63000.35d	dmccs	endl-85		disc	0	yes p(e)	6715	
** eu-151 **									
--->	63151.50c	mccs	endf/b-v	1357	cont	300	yes p(e)	68118	
	63151.50d	dmccs	endf/b-v	1357	disc	300	yes p(e)	10074	
	63151.51c	endf5t	endf/b-v	1357	cont	300	yes p(e)	16796	
** eu-152 **									
--->	63152.50c	endf5u	endf/b-v	1292	cont	300	no	49354	
	63152.50d	dre5	endf/b-v	1292	disc	300	no	5696	
	63152.51c	endf5t	endf/b-v	1292	cont	300	no	10893	
** eu-153 **									
--->	63153.50c	mccs	endf/b-v	1359	cont	300	yes p(e)	55292	
	63153.50d	dmccs	endf/b-v	1359	disc	300	yes p(e)	11305	
	63153.51c	endf5t	endf/b-v	1359	cont	300	yes p(e)	15463	
** eu-154 **									
--->	63154.50c	endf5u	endf/b-v	1293	cont	300	no	37049	
	63154.50d	dre5	endf/b-v	1293	disc	300	no	5499	
	63154.51c	endf5t	endf/b-v	1293	cont	300	no	10407	
z = 64 ************** gadolinium ***									
** gd-nat **									
	64000.01c	bmccs	endl-73	538	cont	0	no	3206	
	64000.01d	d9	endl-73	538	disc	0	no	3078	
	64000.31c	endl79	endl-79		cont	0	no	4192	
	64000.31d	drl79	endl-79		disc	0	no	3092	
--->	64000.35c	mccsa	endl-85		cont	0	yes p(e)	7939	
	64000.35d	dmccs	endl-85		disc	0	yes p(e)	6894	
** gd-152 **									
--->	64152.50c	endf5u	endf/b-v	1362	cont	300	no	26292	
	64152.50d	dre5	endf/b-v	1362	disc	300	no	5940	
	64152.51c	endf5t	endf/b-v	1362	cont	300	no	10970	
** gd-154 **									
--->	64154.50c	endf5u	endf/b-v	1364	cont	300	no	49613	
	64154.50d	dre5	endf/b-v	1364	disc	300	no	5971	

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
	64154.51c	endf5t	endf/b-v	1364	cont	300	no	11520	
**	gd-155 **								
--->	64155.50c	endf5u	endf/b-v	1365	cont	300	no	45006	
	64155.50d	dre5	endf/b-v	1365	disc	300	no	6569	
	64155.51c	endf5t	endf/b-v	1365	cont	300	no	11919	
**	gd-156 **								
--->	64156.50c	endf5u	endf/b-v	1366	cont	300	no	37412	
	64156.50d	dre5	endf/b-v	1366	disc	300	no	6216	
	64156.51c	endf5t	endf/b-v	1366	cont	300	no	11443	
**	gd-157 **								
--->	64157.50c	endf5u	endf/b-v	1367	cont	300	no	39016	
	64157.50d	dre5	endf/b-v	1367	disc	300	no	6387	
	64157.51c	endf5t	endf/b-v	1367	cont	300	no	11365	
**	gd-158 **								
--->	64158.50c	endf5u	endf/b-v	1368	cont	300	no	95917	
	64158.50d	dre5	endf/b-v	1368	disc	300	no	5852	
	64158.51c	endf5t	endf/b-v	1368	cont	300	no	11975	
**	gd-160 **								
--->	64160.50c	endf5u	endf/b-v	1370	cont	300	no	54029	
	64160.50d	dre5	endf/b-v	1370	disc	300	no	5071	
	64160.51c	endf5t	endf/b-v	1370	cont	300	no	10021	

z = 67 ************** holmium **

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
**	ho-165 **								
	67165.01c	bmccs	endl-73	539	cont	0	no	3626	
	67165.01d	d9	endl-73	539	disc	0	no	3289	
	67165.31c	endl79	endl-79		cont	0	no	44092	
	67165.31d	drl79	endl-79		disc	0	no	3311	
--->	67165.35c	rmccsa	endl-85		cont	0	yes p(e)	54340	
	67165.35d	drmccs	endl-85		disc	0	yes p(e)	7080	

z = 73 ************** tantalum **

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
**	ta-181 **								
	73181.01c	amccs	landf	8731	cont	0	no	2431	
	73181.02c	bmccs	endl-73	540	cont	0	no	18114	
	73181.02d	d9	endl-73	540	disc	0	no	4372	
	73181.31c	endl79	endl-79		cont	0	no	20876	
	73181.31d	drl79	endl-79		disc	0	no	5871	
--->	73181.50c	endf5u	endf/b-v	1285	cont	300	yes p(e)	60801	

ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
73181.50d	dre5	endf/b-v	1285	disc	300	yes p(e)	16422	
73181.51c	rmccs	endf/b-v	1285	cont	300	yes p(e)	21588	
73181.51d	drmccs	endf/b-v	1285	disc	300	yes p(e)	16422	

z = 74 ************** tungsten **

ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
** w-nat **								
74000.01c	bmccs	endl-73	541	cont	0	yes h*	3237	
74000.01d	d9	endl-73	541	disc	0	yes h*	4050	
74000.31c	endl79	endl-79		cont	0	no	20541	
74000.31d	drl79	endl-79		disc	0	no	3212	
---> 74000.55c	rmccs	group t-2	7400	cont	300	yes p(e)	50700	
74000.55d	drmccs	group t-2	7400	disc	300	yes p(e)	34333	
** w-182 **								
74182.10c	bmccs	endf/b-iv	1128	cont	0	yes p	33247	
74182.10d	d9	endf/b-iv	1128	disc	0	yes p	5920	
74182.50c	endf5p	endf/b-v	1128	cont	300	yes p(e)	94428	
74182.50d	dre5	endf/b-v	1128	disc	300	yes p(e)	17790	
74182.51c	endf5t	endf/b-v	1128	cont	300	yes p(e)	23259	
---> 74182.55c	rmccsa	group t-2	182	cont	300	yes p(e)	122351	
74182.55d	drmccs	group t-2	182	disc	300	yes p(e)	26448	
** w-183 **								
74183.10c	bmccs	endf/b-iv	1129	cont	0	yes p	27816	
74183.10d	d9	endf/b-iv	1129	disc	0	yes p	7125	
74183.50c	endf5p	endf/b-v	1129	cont	300	yes p(e)	58860	
74183.50d	dre5	endf/b-v	1129	disc	300	yes p(e)	19504	
74183.51c	endf5t	endf/b-v	1129	cont	300	yes p(e)	22707	
---> 74183.55c	rmccsa	group t-2	183	cont	300	yes p(e)	79595	
74183.55d	drmccs	group t-2	183	disc	300	yes p(e)	26381	
** w-184 **								
74184.10c	bmccs	endf/b-iv	1130	cont	0	yes p	27996	
74184.10d	d9	endf/b-iv	1130	disc	0	yes p	6139	
74184.50c	endf5p	endf/b-v	1130	cont	300	yes p(e)	58931	
74184.50d	dre5	endf/b-v	1130	disc	300	yes p(e)	17093	
74184.51c	endf5t	endf/b-v	1130	cont	300	yes p(e)	20631	
---> 74184.55c	rmccsa	group t-2	184	cont	300	yes p(e)	80067	
74184.55d	drmccs	group t-2	184	disc	300	yes p(e)	26171	
** w-186 **								
74186.10c	bmccs	endf/b-iv	1131	cont	0	yes p	30916	
74186.10d	d9	endf/b-iv	1131	disc	0	yes p	6208	
74186.50c	endf5p	endf/b-v	1131	cont	300	yes p(e)	63762	
74186.50d	dre5	endf/b-v	1131	disc	300	yes p(e)	17079	

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
	74186.51c	endf5t	endf/b-v	1131	cont	300	yes p(e)	21487	
--->	74186.55c	rmccsa	group t-2	186	cont	300	yes p(e)	83679	
	74186.55d	drmccs	group t-2	186	disc	300	yes p(e)	26342	
z = 75 ************** rhenium *********									
** re-185 **									
	75185.32c	miscxs	endl-80		cont	0	yes p	13691	
--->	75185.50c	rmccsa	endf/b-v	1083	cont	300	no	9231	
	75185.50d	drmccs	endf/b-v	1083	disc	300	no	4293	
** re-187 **									
	75187.32c	miscxs	endl-80		cont	0	yes p	12359	
--->	75187.50c	rmccsa	endf/b-v	1084	cont	300	no	8303	
	75187.50d	drmccs	endf/b-v	1084	disc	300	no	4716	
z = 78 ************** platinum *********									
** pt-nat **									
	78000.01c	bmccs	endl-73	566	cont	0	no	10313	
	78000.01d	d9	endl-73	566	disc	0	no	3181	
	78000.31c	endl79	endl-79		cont	0	no	10657	
	78000.31d	drl79	endl-79		disc	0	no	3209	
--->	78000.35c	rmccsa	endl-85		cont	0	yes p(e)	15432	
	78000.35d	drmccs	endl-85		disc	0	yes p(e)	6994	
z = 79 ************** gold *********									
** au-197 **									
	79197.01c	xmccs	endl-73	542	cont	0	no	3561	
	79197.02c	xmccs	lamdf	2222	cont	0	no	2660	
	79197.10c	bmccs	endl-75	7163	cont	0	yes p	19844	
	79197.10d	d9	endl-75	7163	disc	0	yes p	4673	
	79197.31c	endl79	endl-79		cont	0	no	25040	
	79197.31d	drl79	endl-79		disc	0	no	3965	
	79197.50c	endf5p	endf/b-v	1379	cont	300	no	139466	
	79197.50d	dre5	endf/b-v	1379	disc	300	no	4923	
	79197.51c	endf5t	endf/b-v	1379	cont	300	no	12283	
--->	79197.55c	rmccsa	group t-2	1379	cont	300	yes p(e)	134386	
	79197.55d	drmccs	group t-2	1379	disc	300	yes p(e)	7944	

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
z = 82	************** lead **								
	** pb-nat **								
	82000.01c	amccs	endl-73	543	cont	0	yes h*	3148	
	82000.01d	d9	endl-73	543	disc	0	yes h*	4056	
	82000.02c	xmccs	lamdf	2026	cont	0	yes h*	2192	
	82000.10c	bmccs	endf/b-iv	1288	cont	0	yes p	21052	
	82000.10d	d9	endf/b-iv	1288	disc	0	yes p	11526	
	82000.31c	endl79	endl-79		cont	0	no	3964	
	82000.31d	drl79	endl-79		disc	0	no	3383	
--->	82000.50c	rmccs	endf/b-v	1382	cont	300	yes p(e)	37694	
	82000.50d	drmccs	endf/b-v	1382	disc	300	yes p(e)	20710	
	82000.51c	endf5t	endf/b-v	1382	cont	300	yes p(e)	37694	
z = 83	************** bismuth ***								
	** bi-209 **								
--->	83209.50c	endf5u	endf/b-v	1375	cont	300	yes p(e)	15000	
	83209.50d	dre5	endf/b-v	1375	disc	300	yes p(e)	7577	
	83209.51c	rmccs	endf/b-v	1375	cont	300	yes p(e)	13782	
	83209.51d	drmccs	endf/b-v	1375	disc	300	yes p(e)	7577	
z = 90	************** thorium ***								
	** th-232 **								
	90232.01c	xmccs	endl-73	544	cont	0	no	4413	prompt
	90232.02c	xmccs	lamdf	2022	cont	0	no	2633	prompt
	90232.10c	bmccs	endl-75	7165	cont	0	yes p	31409	prompt
	90232.10d	d9	endl-75	7165	disc	0	yes p	4565	prompt
	90232.31c	endl79	endl-79		cont	0	no	40220	prompt
	90232.31d	drl79	endl-79		disc	0	no	4045	prompt
--->	90232.50c	endf5u	endf/b-v	1390	cont	300	yes p(e)	152843	both
	90232.50d	dre5	endf/b-v	1390	disc	300	yes p(e)	11998	both
	90232.51c	rmccs	endf/b-v	1390	cont	300	yes p(e)	17986	both
	90232.51d	drmccs	endf/b-v	1390	disc	300	yes p(e)	11998	both
z = 91	************** protactinium **								
	** pa-233 **								
--->	91233.50c	endf5u	endf/b-v	1391	cont	300	no	19560	total

ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
91233.50d	dre5	endf/b-v	1391	disc	300	no	3741	total
91233.51c	rmccs	endf/b-v	1391	cont	300	no	5682	total
91233.51d	drmccs	endf/b-v	1391	disc	300	no	3741	total

z = 92 ************** uranium ***

ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
** u-233 **								
92233.01c	xmccs	endl-73	545	cont	0	no	4779	prompt
92233.10c	bmccs	endl-75	7166	cont	0	yes p	8133	prompt
92233.10d	d9	endl-75	7166	disc	0	yes p	4757	prompt
92233.31c	endl79	endl-79		cont	0	no	22575	prompt
92233.31d	drl79	endl-79		disc	0	no	4029	prompt
——> 92233.50c	rmccs	endf/b-v	1393	cont	300	no	18856	both
92233.50d	drmccs	endf/b-v	1393	disc	300	no	4213	both
92233.51c	endf5t	endf/b-v	1393	cont	300	no	7754	both
** u-234 **								
92234.01c	xmccs	endl-73	546	cont	0	no	2926	prompt
92234.10c	bmccs	endl-75	7167	cont	0	yes p	3184	prompt
92234.10d	d9	endl-75	7167	disc	0	yes p	4911	prompt
92234.31c	endl79	endl-79		cont	0	no	4033	prompt
92234.31d	drl79	endl-79		disc	0	no	4311	prompt
——> 92234.50c	endf5p	endf/b-v	1394	cont	300	no	89474	total
92234.50d	dre5	endf/b-v	1394	disc	300	no	4874	total
92234.51c	rmccs	endf/b-v	1394	cont	300	no	6467	total
92234.51d	drmccs	endf/b-v	1394	disc	300	no	4874	total
** u-235 **								
92235.01c	amccs	endl-73	547	cont	0	yes h*	12958	prompt
92235.01d	d9	endl-73	547	disc	0	yes h*	5349	prompt
92235.04c	xmccs	endf/b-iv	1261	cont	3000	yes h	29516	both
92235.05c	xmccs	endf/b-iv	1261	cont	30000	yes h	18573	both
92235.06c	xmccs	endf/b-iv	1261	cont	6.e+05	yes h	:2560	both
92235.07c	xmccs	endf/b-iv	1261	cont	1.2e+07	yes h	11268	both
92235.08c	xmccs	endf/b-iv	1261	cont	0	yes h	42923	both
92235.09c	xmccs	endf/b-iv	1261	cont	300	yes h	41638	both
92235.10c	amccs	endf/b-iv	1261	cont	0	yes p	42716	both
92235.10d	d9	endf/b-iv	1261	disc	0	yes p	7412	both
92235.11c	amccs	endf/b-iv	1261	cont	300	yes p	41332	both
92235.11d	d9	endf/b-iv	1261	disc	300	yes p	7412	both
92235.15c	rmccsb	endf/b-iv	1261	cont	1.2e+07	yes p(e)	15173	both
92235.15d	d9	endf/b-iv	1261	disc	1.2e+07	yes p(e)	11406	both
92235.18c	amccs	endf/b-iv	1261	cont	3000	yes p	29254	both
92235.18d	d9	endf/b-iv	1261	disc	3000	yes p	7412	both

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
	92235.19c	mmccsb	endf/b-iv	1261	cont	30000	yes p(e)	22298	both
	92235.19d	d9	endf/b-iv	1261	disc	30000	yes p(e)	11406	both
	92235.20c	amccs	endf/b-iv	1261	cont	6.e+05	yes p	12497	both
	92235.20d	d9	endf/b-iv	1261	disc	6.e+05	yes p	7412	both
	92235.30c	amccs	endl-76	7168	cont	0	yes p	18122	total
	92235.30d	d9	endl-76	7168	disc	0	yes p	5289	total
	92235.31c	endl79	endl-79		cont	0	no	19132	prompt
	92235.31d	drl79	endl-79		disc	0	no	4638	prompt
--->	92235.50c	rmccs	endf/b-v	1395	cont	300	yes p(e)	60550	both
	92235.50d	drmccs	endf/b-v	1395	disc	300	yes p(e)	11849	both
	92235.51c	endf5t	endf/b-v	1395	cont	300	yes p(e)	25862	both
**	u-236 **								
	92236.01c	bmccs	endl-73	548	cont	0	no	3327	prompt
	92236.01d	d9	endl-73	548	disc	0	no	4236	prompt
	92236.31c	endl79	endl-79		cont	0	no	4023	prompt
	92236.31d	drl79	endl-79		disc	0	no	4350	prompt
--->	92236.50c	endf5p	endf/b-v	1396	cont	300	no	138756	total
	92236.50d	dre5	endf/b-v	1396	disc	300	no	4879	total
	92236.51c	rmccs	endf/b-v	1396	cont	300	no	7343	total
	92236.51d	drmccs	endf/b-v	1396	disc	300	no	4879	total
**	u-237 **								
	92237.01c	bmccs	endl-73	549	cont	0	no	2676	prompt
	92237.01d	d9	endl-73	549	disc	0	no	3956	prompt
	92237.31c	endl79	endl-79		cont	0	no	4549	prompt
	92237.31d	drl79	endl-79		disc	0	no	3986	prompt
--->	92237.50c	endf5p	endf/b-v	8237	cont	300	yes p(e)	32506	total
	92237.50d	dre5	endf/b-v	8237	disc	300	yes p(e)	8912	total
	92237.51c	rmccs	endf/b-v	8237	cont	300	yes p(e)	10378	total
	92237.51d	drmccs	endf/b-v	8237	disc	300	yes p(e)	8912	total
**	u-238 **								
	92238.01c	amccs	endl-73	550	cont	0	yes h*	4853	prompt
	92238.01d	d9	endl-73	550	disc	0	yes h*	5419	prompt
	92238.04c	xmccs	endf/b-iv	1262	cont	30000	yes h	32918	both
	92238.05c	xmccs	endf/b-iv	1262	cont	6.e+05	yes h	18803	both
	92238.06c	xmccs	endf/b-iv	1262	cont	1.2e+07	yes h	10399	both
	92238.12c	umccs	endf/b-iv	1262	cont	300	yes p	50412	both
	92238.13c	mmccsb	endf/b-iv	1262	cont	30000	yes p(e)	38686	both
	92238.13d	d9	endf/b-iv	1262	disc	30000	yes p(e)	12905	both
	92238.15c	mmccsb	endf/b-iv	1262	cont	1.2e+07	yes p(e)	16230	both
	92238.15d	d9	endf/b-iv	1262	disc	1.2e+07	yes p(e)	13045	both
	92238.20c	amccs	endf/b-iv	1262	cont	6.e+05	yes p	18721	both
	92238.20d	d9	endf/b-iv	1262	disc	6.e+05	yes p	7034	both
	92238.30c	amccs	endl-76	7171	cont	0	yes p	15254	total

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
	92238.30d	d9	endl-76	7171	disc	0	yes p	6937	total
	92238.31c	endl79	endl-79		cont	0	no	18324	prompt
	92238.31d	drl79	endl-79		disc	0	no	8985	prompt
--->	92238.50c	rmccs	endf/b-v	1398	cont	300	yes p(e)	89059	both
	92238.50d	drmccs	endf/b-v	1398	disc	300	yes p(e)	16876	both
	92238.51c	endf5t	endf/b-v	1398	cont	300	yes p(e)	23921	both
**	u-239 **								
	92239.01c	bmccs	endl-73	551	cont	0	no	3393	prompt
	92239.01d	d9	endl-73	551	disc	0	no	4087	prompt
--->	92239.35c	rmccsa	endl-85		cont	0	yes p(e)	9870	prompt
	92239.35d	drmccs	endl-85		disc	0	yes p(e)	9347	prompt
**	u-240 **								
--->	92240.01c	bmccs	endl-73	552	cont	0	no	2951	prompt
	92240.01d	d9	endl-73	552	disc	0	no	4139	prompt
	92240.31c	endl79	endl-79		cont	0	no	3804	prompt
	92240.31d	drl79	endl-79		disc	0	no	4238	prompt

z = 93 ************** neptunium ***

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
**	np-237 **								
	93237.31c	endl79	endl-79		cont	0	no	14149	total
	93237.31d	drl79	endl-79		disc	0	no	4032	total
	93237.50c	endf5p	endf/b-v	1337	cont	300	no	63264	total
	93237.50d	dre5	endf/b-v	1337	disc	300	no	5308	total
	93237.51c	endf5t	endf/b-v	1337	cont	300	no	9787	total
--->	93237.55c	rmccsa	group t-2	1337	cont	300	no	32599	both
	93237.55d	drmccs	group t-2	1337	disc	300	no	20525	both

z = 94 ************** plutonium ***

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
**	pu-238 **								
	94238.01c	bmccs	endl-73	553	cont	0	no	2784	prompt
	94238.01d	d9	endl-73	553	disc	0	no	3845	prompt
	94238.31c	endl79	endl-79		cont	0	no	8624	prompt
	94238.31d	drl79	endl-79		disc	0	no	3901	prompt
--->	94238.50c	endf5p	endf/b-v	1338	cont	300	no	18804	total
	94238.50d	dre5	endf/b-v	1338	disc	300	no	5445	total
	94238.51c	rmccs	endf/b-v	1338	cont	300	no	6108	total
	94238.51d	drmccs	endf/b-v	1338	disc	300	no	5445	total
**	pu-239 **								
	94239.01c	amccs	endl-73	554	cont	0	yes h*	6253	prompt

APPENDIX G

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
	94239.01d	d9	endl-73	554	disc	0	yes h*	6124	prompt
	94239.02c	xmccs	endf/b-iv	1264	cont	3000	yes h	40464	both
	94239.03c	xmccs	endf/b-iv	1264	cont	30000	yes h	25460	both
	94239.04c	xmccs	endf/b-iv	1264	cont	6.e+05	yes h	13633	both
	94239.05c	xmccs	endf/b-iv	1264	cont	1.2e+07	yes h	11349	both
	94239.06c	xmccs	endf/b-iv	1264	cont	0	yes h	41167	both
	94239.07c	xmccs	endf/b-iv	1264	cont	300	yes h	34659	both
	94239.15c	rmccsb	endf/b-iv	1264	cont	1.2e+07	yes p(e)	14205	both
	94239.15d	d9	endf/b-iv	1264	disc	1.2e+07	yes p(e)	11541	both
	94239.16c	amccs	endf/b-iv	1264	cont	0	yes p	41153	both
	94239.16d	d9	endf/b-iv	1264	disc	0	yes p	8056	both
	94239.17c	amccs	endf/b-iv	1264	cont	300	yes p	34631	both
	94239.17d	d9	endf/b-iv	1264	disc	300	yes p	8056	both
	94239.18c	amccs	endf/b-iv	1264	cont	3000	yes p	40421	both
	94239.18d	d9	endf/b-iv	1264	disc	3000	yes p	8056	both
	94239.19c	rmccsb	endf/b-iv	1264	cont	30000	yes p(e)	28311	both
	94239.19d	d9	endf/b-iv	1264	disc	30000	yes p(e)	11541	both
	94239.20c	amccs	endf/b-iv	1264	cont	6.e+05	yes p	13590	both
	94239.20d	d9	endf/b-iv	1264	disc	6.e+05	yes p	8056	both
	94239.31c	endl79	endl-79		cont	0	no	20976	prompt
	94239.31d	drl79	endl-79		disc	0	no	5926	prompt
	94239.50c	endf5p	endf/b-v	1399	cont	300	yes p(e)	74110	both
	94239.50d	dre5	endf/b-v	1399	disc	300	yes p(e)	12692	both
	94239.51c	endf5t	endf/b-v	1399	cont	300	yes p(e)	18898	both
——>	94239.55c	rmccs	group t-2	1399	cont	300	yes p(e)	102160	both
	94239.55d	drmccs	group t-2	1399	disc	300	yes p(e)	20788	both
	94239.99c	xmccs	landf	942	cont	0	yes h*	4031	prompt
**	pu-240 **								
	94240.01c	amccs	endl-73	555	cont	0	yes h*	3829	prompt
	94240.01d	d9	endl-73	555	disc	0	yes h*	4992	prompt
	94240.12c	bmccs	endf/b-iv	1265	cont	900	yes p	41821	both
	94240.12d	d9	endf/b-iv	1265	disc	900	yes p	6087	both
	94240.31c	endl79	endl-79		cont	0	no	33109	prompt
	94240.31d	drl79	endl-79		disc	0	no	5339	prompt
——>	94240.50c	rmccs	endf/b-v	1380	cont	300	yes p(e)	58978	both
	94240.50d	drmccs	endf/b-v	1380	disc	300	yes p(e)	9630	both
	94240.51c	endf5t	endf/b-v	1380	cont	300	yes p(e)	15195	both
**	pu-241 **								
	94241.01c	bmccs	endl-73	556	cont	0	no	3615	prompt
	94241.01d	d9	endl-73	556	disc	0	no	4281	prompt
	94241.31c	endl79	endl-79		cont	0	no	4139	prompt
	94241.31d	drl79	endl-79		disc	0	no	4303	prompt
——>	94241.50c	endf5p	endf/b-v	1381	cont	300	yes p(e)	38662	both

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
	94241.50d	dre5	endf/b-v	1381	disc	300	yes p(e)	11636	both
	94241.51c	rmccs	endf/b-v	1381	cont	300	yes p(e)	13464	both
	94241.51d	drmccs	endf/b-v	1381	disc	300	yes p(e)	11636	both
**	pu-242 **								
--->	94242.50c	endf5p	endf/b-v	1342	cont	300	yes p(e)	71490	both
	94242.50d	dre5	endf/b-v	1342	disc	300	yes p(e)	12524	both
	94242.51c	rmccs	endf/b-v	1342	cont	300	yes p(e)	15763	both
	94242.51d	drmccs	endf/b-v	1342	disc	300	yes p(e)	12524	both
**	pu-243 **								
--->	94243.31c	bmccs	endl-79		cont	0	no	7087	prompt
	94243.31d	d9	endl-79		disc	0	no	5547	prompt

z = 95 ************** americium ***

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
**	am-241 **								
	95241.31c	endl79	endl-79		cont	0	no	19896	prompt
	95241.31d	drl79	endl-79		disc	0	no	4378	prompt
--->	95241.50c	endf5u	endf/b-v	1361	cont	300	yes p(e)	42145	total
	95241.50d	dre5	endf/b-v	1361	disc	300	yes p(e)	10032	total
	95241.51c	rmccs	endf/b-v	1361	cont	300	yes p(e)	12435	total
	95241.51d	drmccs	endf/b-v	1361	disc	300	yes p(e)	10032	total
**	am-242m **								
	95242.01c	bmccs	endl-73	557	cont	0	no	5947	prompt
	95242.01d	d9	endl-73	557	disc	0	no	3965	prompt
	95242.31c	endl79	endl-79		cont	0	no	6575	prompt
	95242.31d	drl79	endl-79		disc	0	no	4005	prompt
--->	95242.50c	endf5u	endf/b-v	1369	cont	300	yes p(e)	8654	total
	95242.50d	dre5	endf/b-v	1369	disc	300	yes p(e)	9109	total
	95242.51c	rmccs	endf/b-v	1369	cont	300	yes p(e)	8563	total
	95242.51d	drmccs	endf/b-v	1369	disc	300	yes p(e)	9109	total
**	am-243 **								
	95243.31c	endl79	endl-79		cont	0	no	31050	total
	95243.31d	drl79	endl-79		disc	0	no	4255	total
--->	95243.50c	endf5u	endf/b-v	1363	cont	300	yes p(e)	92076	total
	95243.50d	dre5	endf/b-v	1363	disc	300	yes p(e)	11803	total
	95243.51c	rmccs	endf/b-v	1363	cont	300	yes p(e)	13745	total
	95243.51d	drmccs	endf/b-v	1363	disc	300	yes p(e)	11803	total

	ZAID	FILE	SOURCE	MAT	TYPE	TEMP(°K)	GPD	LENGTH	NUBAR
z = 96	**************	curium	******	******	******	******	******	******	******
** cm-242 **									
	96242.31c	endl79	endl-79		cont	0	no	15537	total
	96242.31d	drl79	endl-79		disc	0	no	4197	total
——>	96242.50c	endf5u	endf/b-v	8642	cont	300	yes p(e)	30958	total
	96242.50d	dre5	endf/b-v	8642	disc	300	yes p(e)	8964	total
	96242.51c	rmccs	endf/b-v	8642	cont	300	yes p(e)	9828	total
	96242.51d	drmccs	endf/b-v	8642	disc	300	yes p(e)	8964	total
** cm-266 **									
	96244.31c	endl79	endl-79		cont	0	no	15126	total
	96244.31d	drl79	endl-79		disc	0	no	4290	total
——>	96244.50c	endf5u	endf/b-v	1344	cont	300	yes p(e)	46052	total
	96244.50d	dre5	endf/b-v	1344	disc	300	yes p(e)	9570	total
	96244.51c	rmccs	endf/b-v	1344	cont	300	yes p(e)	10908	total
	96244.51d	drmccs	endf/b-v	1344	disc	300	yes p(e)	9570	total

Appendix H

FISSION SPECTRA CONSTANTS AND FLUX-TO-DOSE FACTORS

This Appendix is divided into two sections: fission spectra constants to be used with the SP input card and ANSI standard flux-to-dose conversion factors to be used with the DE and DF input cards.

I. CONSTANTS FOR FISSION SPECTRA

The following is a list of recommended parameters for use with the MCNP source fission spectra and the SP input card described in Chapter 3. The constants for neutron-induced fission are taken directly from the ENDF/B-V library. For each fissionable isotope, constants are given for either the Maxwell spectrum or the Watt spectrum, but not both. The Watt fission spectrum is preferred to the Maxwell fission spectrum. The constants for spontaneously fissioning isotopes are supplied by Madland of Group T-2. If you desire constants for isotopes other than those listed below, contact X-6. Note that both the Watt and Maxwell fission spectra are approximations. A more accurate representation has been developed by Madland in T-2. If you are interested in this spectrum, contact X-6.

A. Constants for the Maxwell fission spectrum (neutron-induced)

$$f(E) = CE^{1/2}\exp(-E/a)$$

	Incident Neutron Energy (MeV)	a (MeV)
$n + {}^{233}Pa$	Thermal	1.3294
	1	1.3294
	14	1.3294
$n + {}^{234}U$	Thermal	1.2955
	1	1.3086
	14	1.4792
$n + {}^{236}U$	Thermal	1.2955
	1	1.3086
	14	1.4792

n + ^{237}U	Thermal	1.2996
	1	1.3162
	14	1.5063
n + ^{237}Np	Thermal	1.315
	1	1.315
	14	1.315
n + ^{238}Pu	Thermal	1.330
	1	1.330
	14	1.330
n + ^{240}Pu	Thermal	1.346
	1	1.3615
	14	1.547
n + ^{241}Pu	Thermal	1.3597
	1	1.3752
	14	1.5323
n + ^{242}Pu	Thermal	1.337
	1	1.354
	14	1.552
n + ^{241}Am	Thermal	1.330
	1	1.330
	14	1.330
n + ^{242m}Am	Thermal	1.330
	1	1.330
	14	1.330
n + ^{243}Am	Thermal	1.330
	1	1.330
	14	1.330
n + ^{242}Cm	Thermal	1.330
	1	1.330
	14	1.330
n + ^{244}Cm	Thermal	1.330
	1	1.330
	14	1.330

APPENDIX H
Fission Spectra

n + ^{245}Cm	Thermal	1.4501
	1	1.4687
	14	1.6844
n + ^{246}Cm	Thermal	1.3624
	1	1.4075
	14	1.6412

B. *Constants for the Watt Fission Spectrum*

$$f(E) = C \exp(-E/a) \sinh(bE)^{1/2}$$

1. Neutron-Induced Fission

	Incident Neutron Energy (MeV)	a (MeV)	b (MeV^{-1})
n + ^{232}Th	Thermal	1.0888	1.6871
	1	1.1096	1.6316
	14	1.1700	1.4610
n + ^{233}U	Thermal	0.977	2.546
	1	0.977	2.546
	14	1.0036	2.6377
n + ^{235}U	Thermal	0.988	2.249
	1	0.988	2.249
	14	1.028	2.084
n + ^{238}U	Thermal	0.88111	3.4005
	1	0.89506	3.2953
	14	0.96534	2.8330
n + ^{239}Pu	Thermal	0.966	2.842
	1	0.966	2.842
	14	1.055	2.383

2. Spontaneous Fission

	a (MeV)	b (MeV^{-1})
^{240}Pu	0.799	4.903
^{242}Cm	0.891	4.046
^{244}Cm	0.906	3.848
^{252}Cf	1.025	2.926

II. Flux-to-Dose Conversion Factors

This section presents several flux-to-dose rate conversion factor sets for use on the DE and DF tally cards to convert from calculated particle flux to human biological dose equivalent rate. Note that these sets of conversion factors are not the only ones in existence, nor are they recommended by this publication. Rather, they are presented for convenience should you decide that one is appropriate for your use. The original publication cited or other sources should be consulted to determine if they are appropriate for your application.

Although the various conversion factor sets differ from one another, it seems to be the consensus of the health physics community that they do not differ significantly from most health physics applications where accuracies of ±20% are generally acceptable. Some of the differences in the various sets are attributable to different assumptions about source directionality, phantom geometry, and depth of penetration. The neutron quality factors, derived primarily from animal experiments, are also somewhat different.

Be aware that conversion factor sets are subject to change based on the actions of various national and international organizations such as the National Council on Radiation Protection and Measurements (NCRP), the International Commission on Radiological Protection (ICRP), the International Commission on Radiation Units and Measurements (ICRU), the American National Standards Institute (ANSI), and the American Nuclear Society (ANS). Changes may be based on the re-evaluation of existing data and calculations or on the availability of new information. Currently, a revision of the 1977 ANSI/ANS[54] conversion factors is under way and the ICRP and NCRP are considering an increase in the neutron quality factors by a factor of 2 to 2.5.

In addition to biological dose factors, a reference is given for silicon displacement kerma factors for potential use in radiation effects

assessment of electronic semiconductor devices. The use of these factors is subject to the same caveats stated above for biological dose rates.

A. *Biological Dose Equivalent Rate Factors*

In the following discussions, dose rate will be used interchangeably with biological dose equivalent rate. In all cases the conversion factors will contain the quality factors used to convert the absorbed dose in rads to rem. The neutron quality factors implicit in the conversion factors are also tabulated for information. For consistency, all conversion factors are given in units of rem/h per unit flux (particles/cm^2-s) rather than in the units given by the original publication. The interpolation mode chosen should correspond to that recommended by the reference. For example, the ANSI/ANS publication recommends log-log interpolation; significant differences at interpolated energies can result if a different interpolation scheme is used.

1. Neutrons

The NCRP-38[Ref. 55] and ICRP-21[Ref. 56] neutron flux-to-dose rate conversion factors and quality factors are listed in Table H.1. Note that the 1977 ANSI/ANS factors referred to earlier were taken from NCRP-38 and therefore are not listed separately.

2. Photons

The 1977 ANSI/ANS[54] and the ICRP-21[Ref. 56] photon flux-to-dose rate conversion factors are given the Table H.2. No tabulated set of photon conversion factors have been provided by the NCRP as far as can be determined. Note that the 1977 ANSI/ANS and the ICRP-21 conversion factor sets differ significantly (>20%) below approximately 0.7 MeV. The maximum disagreement occurs at approximately 0.06 MeV, where the ANSI/ANS value is about 2.3 times larger than the ICRP value.

B. *Silicon Displacement Kerma Factors*

Radiation damage to or effects on electronic components are often of interest in radiation fields. Of particular interest are the absorbed dose in rads and silicon displacement kerma factors. The absorbed dose may be calculated for a specific material by using the FM tally card discussed in Chapter 3 with an appropriate constant C to convert from the MCNP default units to rads. The silicon displacement kermas, however, are given as a function of energy, similar to the biological conversion factors. Therefore, they may be implemented on the DE and DF cards. One source of

these kerma factors and a discussion of their significance and use can be found in Reference 57.

Table H.1
Neutron Flux-to-Dose Rate Conversion Factors and Quality Factors

Neutron Energy, E (MeV)	NCRP-38, ANSI/ANS-6.1.1-1977* DF(E) (rem/hr)/(n/cm^2·s)	NCRP-38, ANSI/ANS-6.1.1-1977* Quality Factor	ICRP-21 DF(E) (rem/hr)/(n/cm^2·s)	ICRP-21 Quality Factor
2.5E-08	3.67E-06	2	3.85E-06	2.3
1.0E-07	3.67E-06	2	4.17E-06	2
1.0E-06	4.46E-06	2	4.55E-06	2
1.0E-05	4.54E-06	2	4.35E-06	2
1.0E-04	4.18E-06	2	4.17E-06	2
1.0E-03	3.76E-06	2	3.70E-06	2
1.0E-02	3.56E-06	2.5	3.57E-06	2
1.0E-01	2.17E-05	7.5	2.08E-05	7.4
5.0E-01	9.26E-05	11	7.14E-05	11
1.0	1.32E-04	11	1.18E-04	10.6
2.0			1.43E-04	9.3
2.5	1.25E-04	9		
5.0	1.56E-04	8	1.47E-04	7.8
7.0	1.47E-04	7		
10.0	1.47E-04	6.5	1.47E-04	6.8
14.0	2.08E-04	7.5		
20.0	2.27E-04	8	1.54E-04	6

*Extracted from American National Standard ANSI/ANS-6.1.1-1977 with permission of the publisher, the American Nuclear Society.

Table H.2
Photon Flux-to-Dose Rate Conversion Factors

ANSI/ANS-6.1.1-1977		ICRP-21	
Energy,E (MeV)	DF(E) (rem/hr)/(photons/cm^2·s)	Energy,E (MeV)	DF(E) (rem/hr)/(photons/cm^2·s)
0.01	3.96E-06	0.01	2.78E-06
0.03	5.82E-07	0.015	1.11E-06
0.05	2.90E-07	0.02	5.88E-07
0.07	2.58E-07	0.03	2.56E-07
0.1	2.83E-07	0.04	1.56E-07
0.15	3.79E-07	0.05	1.20E-07
0.2	5.01E-07	0.06	1.11E-07
0.25	6.31E-07	0.08	1.20E-07
0.3	7.59E-07	0.1	1.47E-07
0.35	8.78E-07	0.15	2.38E-07
0.4	9.85E-07	0.2	3.45E-07
0.45	1.08E-06	0.3	5.56E-07
0.5	1.17E-06	0.4	7.69E-07
0.55	1.27E-06	0.5	9.09E-07
0.6	1.36E-06	0.6	1.14E-06
0.65	1.44E-06	0.8	1.47E-06
0.7	1.52E-06	1.	1.79E-06
0.8	1.68E-06	1.5	2.44E-06
1.0	1.98E-06	2.	3.03E-06
1.4	2.51E-06	3.	4.00E-06
1.8	2.99E-06	4.	4.76E-06
2.2	3.42E-06	5.	5.56E-06
2.6	3.82E-06	6.	6.25E-06
2.8	4.01E-06	8.	7.69E-06
3.25	4.41E-06	10.	9.09E-06
3.75	4.83E-06		
4.25	5.23E-06		
4.75	5.60E-06		
5.0	5.80E-06		
5.25	6.01E-06		
5.75	6.37E-06		
6.25	6.74E-06		
6.75	7.11E-06		
7.5	7.66E-06		
9.0	8.77E-06		
11.0	1.03E-05		
13.0	1.18E-05		
15.0	1.33E-05		

Appendix I

GLOSSARY

1. Monte Carlo – a numerical analysis technique that uses random sampling to construct the solution of a physical or mathematical problem.

2. Random Numbers – an infinite set of numbers that are uniformly distributed from 0 to 1 and are uncorrelated. We actually use pseudorandom numbers, a deterministic reproducible sequence of random numbers generated by a computer that satisfy statistical tests for randomness.

3. Probability Density Function (PDF) – a non–negative mathematical function of unit integral that describes the frequency of occurence of various events.

4. Cumulative Distribution Function (CDF) – the integral of the PDF that represents the probability of selecting an event. (The total integral is unity.)

5. Random Sampling – using pseudorandom numbers to select events from CDFs or PDFs.

6. Rejection Technique – a PDF $f(x)$ can be sampled from a different PDF, $g(x)/\int g(y)dy$, for which $g(x) \geq f(x)$, and accepting x_i with probability $f(x_i)/g(x_i)$. Otherwise x_i is rejected and a new x_i is sampled.

7. Physical Phase Space – $\vec{r}$, E, $\vec{\Omega}$, t, which are all of the variables required to describe a physical particle.

8. Monte Carlo Weight – the number of physical particles W that a Monte Carlo particle represents.

9. Monte Carlo Phase Space – W, $\vec{r}$, E, $\vec{\Omega}$, t, which are all the variables required to describe a Monte Carlo particle.

10. Random Walk – the random selection of events for a particle history.

11. History – the complete random walk of a Monte Carlo particle from its birth in the source to its death, including all progeny.

12. Monte Carlo Track – a branch, or subset, of a history that can be obtained by physical events (for example, fissions) or by variance reduction techniques (for example, geometry splitting).

13. Analog Monte Carlo – the sampling procedure that directly mimics the transport process.

13. Nonanalog Monte Carlo – any sampling procedure that does not directly mimic the transport process. (See variance reduction techniques.)

15. Score – contribution from a track to a tally.

16. History Score – sum of all scores from one source particle's tracks.

17. Tally – used interchangeably with score. Also, the quantity we want to estimate (average score), obtained by summing all scores from all histories.

18. Central Limit Theorem – for large sample sizes N and independent and identically distributed random variables x_i with finite means and variances, the distribution of the sample means approaches a normal distribution. This theorem is the basis for error estimation in MCNP.

19. Sample Variance – the spread of the x_i's compared to the sample mean X_N, estimated by

$$S^2 = \frac{1}{N-1}\sum_{i=1}^{N} (x_i - X_N)^2 \sim \overline{X_N^2} - X_N^2$$

20. Standard Deviation of the Mean – estimated by $S_N = S/\sqrt{N}$, which determines confidence intervals for X_N.

21. Relative Error – the standard deviation of the mean of a tally divided by the mean. Printed with each tally is an associated relative (and not percent) error R representing one standard deviation of the tally. That is, as $N \infty$ the tally is correct to within $\pm 100 \cdot R\%$ about 68% of the time or to within $\pm 100 \cdot 2R\%$ about 95% of the time. The error refers to the precision of the tally, not to its accuracy.

22. Precision – the uncertainty in X_N caused by the statistical behavior of x_i's for the portion of physical phase space sampled by the Monte Carlo process.

23. Accuracy – the degree to which the estimated mean X_N represents the true physical mean, the deviation being a function of the nuclear data, the Monte Carlo model, and the adequacy of the Monte Carlo sampling, reflected by the precision.

24. Figure of Merit (FOM) – a quantity defined for a tally as $1/(R^2T)$ where R is the relative error and T is computer time. For a well-sampled problem, the FOM will approach a constant as N ∞. It is a relative indication of the computational efficiency of an MCNP calculation – the larger the better.

25. Variance Reduction Technique – also referred to as biasing. Any analog technique applied to increase computational efficiency. (See FOM.) For example, a particle of weight w can be replaced by N particles of weight w/N.

26. Importance – the expected score per unit weight of a track at phase-space point ($\vec{r}$, E, $\vec{\Omega}$, t).

27. Flux Density – the product of particle density and particle speed. The flux density is often referred to simply as flux and is mathematically and physically equivalent to the sum of the lengths of all Monte Carlo tracks per unit volume per unit time. Flux density is in units of particles/cm^2/shake.

28. Fluence – integral of flux over time. If the MCNP source is in units of particles, the flux tallies are really fluence tallies. If the MCNP source is in units of particles per unit time, the flux tallies are truly fluxes. Fluence has units of particles/cm^2.

29. Current – the number of particles crossing a surface in a given time interval and in a given direction interval.

REFERENCES

1. Video reels 1-25 taped at Magnetic Fusion Energy Conference on MCNP, October 1983, available from RSIC, Oak Ridge National Laboratory, Oak Ridge, TN 37830.

2. T. E. Booth, "A Sample Problem for Variance Reduction in MCNP," Los Alamos National Laboratory report LA-10363-MS (June 1985).

3. R. Kinsey, "Data Formats and Procedures for the Evaluated Nuclear Data File, ENDF," Brookhaven National Laboratory report BNL-NCS-50496 (ENDF 102) 2nd Edition (ENDF/B-V) (October 1979).

4. R. J. Howerton, D. E. Cullen, R. C. Haight, M. H. MacGregor, S. T. Perkins, and E. F. Plechaty, "The LLL Evaluated Nuclear Data Library (ENDL): Evaluation Techniques, Reaction Index, and Descriptions of Individual Reactions," Lawrence Livermore National Laboratory report UCRL-50400, Vol. 15, Part A (September 1975).

5. M. A. Gardner and R. J. Howerton, "ACTL: Evaluated Neutron Activation Cross-Section Library - Evaluation Techniques and Reaction Index," Lawrence Livermore National Laboratory report UCRL-50400, Vol. 18 (October 1978).

6. E. D. Arthur and P. G. Young, "Evaluated Neutron-Induced Cross Sections for ^{54,56}Fe to 40 MeV," Los Alamos Scientific Laboratory report LA-8626-MS (ENDF-304) (December 1980).

7. D. G. Foster, Jr. and E. D. Arthur, "Average Neutronic Properties of "Prompt" Fission Products," Los Alamos National Laboratory report LA-9168-MS (February 1982).

8. E. D. Arthur, P. G. Young, A. B. Smith, and C. A. Philis, "New Tungsten Isotope Evaluations for Neutron Energies Between 0.1 and 20 MeV," *Trans. Am. Nucl. Soc.* 39, 793 (1981).

9. R. E. MacFarlane, D. W. Muir, and R. M. Boicourt, "The NJOY Nuclear Data Processing System, Volume I: User's Manual," Los Alamos National Laboratory report LA-9303-M, Vol. I (ENDF-324) (May 1982).
   R. E. MacFarlane, D. W. Muir, and R. M. Boicourt, "The NJOY Nuclear Data Processing System, Volume II: The NJOY, RECONR, BROADR, HEATR,

and THERMR Modules," Los Alamos National Laboratory report LA-9303-M, Vol. II (ENDF-324) (May 1982).

10. L. L. Carter and E. D. Cashwell, "Particle-Transport Simulation with the Monte Carlo Method," *ERDA Critical Review Series*, TID-26607 (1975).

11. R. Alcouffe, R. Dautray, A. Forster, G. Ledanois, and B. Mercier, Eds. Monte Carlo Methods and Applications in Neutronics, Photonics and Statistical Physics, Proceedings, 1985. Vol. 240 in Lecture Notes in Physics, H. Araki, J. Ehlers, K. Hepp, R. Kippenhahn, H. A. Weidenmuller, and J. Zittartz, Eds. (Springer-Verlag, Berlin, 1985).

12. W. L. Thompson, E. D. Cashwell, T. N. K. Godfrey, R. G. Schrandt, O. L. Deutsch, and T. E. Booth, "The Status of Monte Carlo at Los Alamos," Los Alamos Scientific Laboratory report LA-8353-MS (May 1980).

13. C. J. Everett and E. D. Cashwell, "A Third Monte Carlo Sampler," Los Alamos National Laboratory report LA-9721-MS (March 1983).

14. "Fermi Invention Rediscovered at LASL," *The Atom*, Los Alamos Scientific Laboratory (October 1966).

15. E. D. Cashwell and C. J. Everett, *A Practical Manual on the Monte Carlo Method for Random Walk Problems*, Pergamon Press, Inc., New York (1959).

16. Robert R. Johnston, "A General Monte Carlo Neutronics Code," Los Alamos Scientific Laboratory report LAMS-2856 (March 1963).

17. E. D. Cashwell, J. R. Neergaard, W. M. Taylor, and G. D. Turner, "MCN: A Neutron Monte Carlo Code," Los Alamos Scientific Laboratory report LA-4751 (January 1972).

18. E. D. Cashwell, J. R. Neergaard, C. J. Everett, R. G. Schrandt, W. M. Taylor, and G. D. Turner, "Monte Carlo Photon Codes: MCG and MCP," Los Alamos Scientific Laboratory report LA-5157-MS (March 1973).

19. K. C. Chandler and T. W. Armstrong, "Operating Instructions for the High-Energy Nucleon-Meson Transport Code HETC," Oak Ridge National Laboratory report ORNL-4744 (January 1972).

20. E. D. Cashwell and C. J. Everett, "Intersection of a Ray with a Surface of Third or Fourth Degree," Los Alamos Scientific Laboratory report LA-4299 (December 1969).

REFERENCES

21. M. W. Asprey, R. B. Lazarus, and R. E. Seamon, "EVXS: A Code to Generate Multigroup Cross Sections from the Los Alamos Master Data File," Los Alamos Scientific Laboratory report LA-4855 (June 1974).

22. R. J. Howerton, R. E. Dye, P. C. Giles, J. R. Kimlinger, S. T. Perkins and E. F. Plechaty, "Omega Documentation," Lawrence Livermore National Laboratory report UCRL-50400, Vol. 25 (August 1983).

23. J. H. Hubbell, W. J. Veigele, E. A. Briggs, R. T. Brown, D. T. Cromer and R. J. Howerton, "Atomic Form Factors, Incoherent Scattering Functions, and Photon Scattering Cross Sections," *J. Phys. Chem. Ref. Data* 4, 471 (1975).

24. E. Storm and H. I. Israel, "Photon Cross Sections from 0.001 to 100 Mev for Elements 1 through 100," Los Alamos Scientific Laboratory report LA-3753 (November 1967).

25. C. J. Everett and E. D. Cashwell, "MCP Code Fluorescence-Routine Revision," Los Alamos Scientific Laboratory report LA-5240-MS (May 1973).

26. J. U. Koppel and D. H. Houston, "Reference Manual for ENDF Thermal Neutron Scattering Data," General Atomics report GA-8744, Revised (ENDF-269) (July 1978).

27. R. C. Little and R. E. Seamon, "Neutron-Induced Photon Production in MCNP," *Proceedings of the Sixth International Conference on Radiation Shielding*, Vol. I, 151, (May 1983).

28. H. Kahn, "Applications of Monte Carlo," AEC-3259 The Rand Corporation (April 1956).

29. L. Koblinger, "Direct Sampling from the Klein-Nishina Distribution for Photon Energies Above 1.4 MeV," *Nucl. Sci. Eng.*, 56, 218 (1975).

30. R. N. Blomquist and E. M. Gelbard, "An Assessment of Existing Klein-Nishina Monte Carlo Sampling Methods," *Nucl. Sci. Eng.*, 83, 380 (1983).

31. J. E. Stewart, "A General Point-on-a-Ring Detector," *Transactions of the American Nuclear Society*, 28, 643 (1978).

32. R. A. Forster, "Ring Detector and Angle Biasing," Los Alamos Scientific Laboratory technical memorandum TD-6-8-79 (July 1979).

33. H. J. Kalli and E. D. Cashwell, "Evaluation of Three Monte Carlo Estimation Schemes for Flux at a Point," Los Alamos Scientific Laboratory report LA-6865-MS (September 1977).

34. A. Dubi, "On the Analysis of the Variance in Monte Carlo Calculations," *Nucl. Sci. Eng.*, 72, 108 (1979). See also I. Lux, "On Efficient Estimation of Variances," *Nucl. Sci. Eng.*, 92, 607 (1986).

35. G. P. Estes and E. D. Cashwell, "MCNP1B Variance Error Estimator," Los Alamos Scientific Laboratory memorandum TD-6-27-78 (August 1978).

36. J. S. Hendricks, "Construction of Equiprobable Bins for Monte Carlo Calculation," *Trans. Am. Nucl. Soc.*, 35, 247 (1980).

37. T. E. Booth and J. S. Hendricks, "Importance Estimation in Forward Monte Carlo Calculations," *Nucl. Tech./Fusion*, 5 (1984).

38. F. H. Clark, "The Exponential Transform as an Importance-Sampling Device, A Review," ORNL-RSIC-14 (January 1966).

39. P. K. Sarkar and M. A. Prasad, "Prediction of Statistical Error and Optimization of Biased Monte Carlo Transport Calculations," *Nucl. Sci. Eng.*, 70, 243-261, (1979).

40. G. Bell and S. Glasstone, Nuclear Reactor Theory, Litton Educational Publishing, Inc., 1970.

41. M. Halperin, "Almost Linearly-Optimum Combination of Unbiased Estimates," *Amer. Stat. Ass. J.*, 56, 36-43 (1961).

42. E. M. Gelbard and R. E. Prael, "Monte Carlo Work at Argonne National Laboratory," Proceedings of the NEACRP Meeting of a Monte Carlo Study Group, ANL-75-2 (July 1974).

43. J. S. Hendricks, "Calculation of Cell Volumes and Surface Areas in MCNP," Los Alamos Scientific Laboratory report LA-8113-MS (January 1980).

44. B. Spain, Analytical Conics, Pergamon, 1957.

45. D. H. Lehmer, "Mathematical Methods in Large-Scale Computing Units," *Am. Comput. Lab., Harvard Univ.*, 26, 141-146 (1951).

REFERENCES

46. D. Garber (editor), "ENDF/B-V," Report BNL-17541 (ENDF-201), National Nuclear Data Center, Brookhaven National Laboratory, Upton, N. Y. (October 1975).

47. K. Parker, "The Format and Conventions of the U.K.A.E.A. Nuclear Data Library," Atomic Weapons Research Establishment report AWRE O-70/63, Aldermasten, England (1963).

48. P. G. Young and L. Stewart, "Evaluated Data for n+^{9}Be Reactions," Los Alamos Scientific Laboratory report LA-7932-MS (ENDF-283) (July 1979).

49. UPDATE Reference Manual, Control Data Corporation Publication 60449900 (1975).

50. The HISTORIAN User's Manual, OpCode, Inc. (April 1983).

51. American National Standards Institute, Inc., American National Standard Programming Language FORTRAN, ANSI X3.9-1978., (New York, 1978).

52. P. D. Soran and R. E. Seamon, "Graphs of the Cross Sections in the Recommended Monte Carlo Cross-Section Library at the Los Alamos Scientific Laboratory," Los Alamos Scientific Laboratory report LA-8374-MS (May 1980).

53. R. E. Seamon and P. D. Soran, "Graphs of the Cross Sections in the Alternate Cross-Section Library at the Los Alamos Scientific Laboratory," Los Alamos Scientific Laboratory report LA-8394-MS (June 1980).

54. ANS-6.1.1 Working Group, M. E. Battat (Chairman), "American National Standard Neutron and Gamma-Ray Flux-to-Dose Rate Factors," ANSI/ANS-6.1.1-1977 (N666), American Nuclear Society, LaGrange Park, Illinois (1977).

55. NCRP Scientific Committee 4 on Heavy Particles, H. H. Rossi, chairman, "Protection Against Neutron Radiation," NCRP-38, National Council on Radiation Protection and Measurements (January 1971).

56. ICRP Committee 3 Task Group, P. Grande and M. C. O'Riordan, chairmen, "Data for Protection Against Ionizing Radiation from External Sources: Supplement to ICRP Publication 15," ICRP-21, International Commission on Radiological Protection, Pergamon Press (April 1971).

57. ASTM Committee E-10 on Nuclear Technology and Applications, "Characterizing Neutron Energy Fluence Spectra in Terms of an Equivalent Monoenergetic Neutron Fluence for Radiation-Hardness Testing of Electronics," American Society for Testing and Materials Standard E722-80, Annual Book of ASTM Standards (1980).

Printed in the United States of America
Available from
National Technical Information Service
US Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

Microfiche (A01)

Page Range	NTIS Price Code	Page Range	NTIS Price Code	Page Range	NTIS Price Code	Page Range	NTIS Price Code
001-025	A02	151-175	A08	301-325	A14	451-475	A20
026-050	A03	176-200	A09	326-350	A15	476-500	A21
051-075	A04	201-225	A10	351-375	A16	501-525	A22
076-100	A05	226-250	A11	376-400	A17	526-550	A23
101-125	A06	251-275	A12	401-425	A18	551-575	A24
126-150	A07	276-300	A13	426-450	A19	576-600	A25
						601-up*	A99

*Contact NTIS for a price quote.

☆U.S. GOVERNMENT PRINTING OFFICE:1986—676-034/40034