ornl

ORNL-4972/R1

ORNL--4972/R1

DE83 009578

**OAK
RIDGE
NATIONAL
LABORATORY**

**UNION
CARBIDE**

**The Morse Monte Carlo
Radiation Transport
Code System**

M. B. Emmett

**OPERATED BY
UNION CARBIDE CORPORATION
FOR THE UNITED STATES
DEPARTMENT OF ENERGY**

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A05; Microfiche A01

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Contract No. W-7405-eng-26

Engineering Physics Division

Revisions and Additions for ORNL-4972

THE MORSE MONTE CARLO RADIATION TRANSPORT CODE SYSTEM

M. B. Emmett*

*Computer Sciences Division, UCC Nuclear Division

Date Published: FEBRUARY 1983

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
DEPARTMENT OF ENERGY

MASTER

EBB

Preface

This report is an addendum to the MORSE report, ORNL-4972, originally published in 1975. This addendum contains descriptions of several modifications to the MORSE Monte Carlo Code, replacement pages containing corrections, Part II of the report which was previously unpublished, and a new Table of Contents. All pages have been three-hole punched so that users having loose-leaf copies of the original report can insert or replace pages in their binders; others can treat this as a second volume of ORNL-4972.

The modifications include a Klein Nishina estimator for gamma rays which is described on pages 4.6-41 to 4.6-44. Use of such an estimator required changing the cross section routines to process pair production and Compton scattering cross sections directly from ENDF tapes and writing a new version of subroutine RELCOL.

Another modification is the use of free form input for the SAMBO analysis data described on page 4.6-24.1. This required changing subroutines SCORIN and adding new subroutine RFRE.

References on page 4.12-2 were updated.

Errors have been corrected in the following pages from the original report:

Title page	4.4-49	4.8-5	4.10-11
4.3-7	4.5-14	4.8-8	4.10-17
4.3-11	4.6-23	4.10-5	4.10-18
4.3-12	4.7-5	4.10-6	4.10-20
4.3-13	4.7-11	4.10-8	4.10-23
4.4-38	4.7-18	4.10-9	4.10-25
4.4-39	4.7-30	4.10-10	4.10-26
4.4-48			4.10-27

Part II of the MORSE report is a guide on using MORSE. It attempts to help the user solve some of the more common problems incurred in running the code. It is by no means a complete list of troubles, rather, it is an attempt to help the user do his own troubleshooting.

MORSE-CG is now available for the CRAY computer. It has been tested on the Lawrence Livermore Laboratories MFE CRAY and is now available from the Radiation Shielding Information Center (RSIC) at ORNL.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL-4972/R1

Contract No. W-7405-eng-26

Engineering Physics Division

THE MORSE MONTE CARLO RADIATION TRANSPORT CODE SYSTEM

M. B. Emmett*

NOTE:

This Work Supported by
DEFENSE NUCLEAR AGENCY
Under Subtask PE 074

*Computer Sciences, UCC Nuclear Division

Originally Published February 1975

REVISED FEBRUARY 1983

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
DEPARTMENT OF ENERGY

TABLE OF CONTENTS

	Page
PART I - General Introduction to the MORSE Code	
1.1 Introduction	1.1-1
1.2 References	1.2-1
PART II - A Guide on How to Use the Various Codes	2.0-1
2.1 Introduction	2.1-1
2.2 Is Core Storage Space a Problem?	2.2-1
2.3 Is Amount of Computer Time Too Large?	2.3-1
2.4 Are There Too Many Abnormal Terminations	2.4-1
2.5 Is There Too Much Output?	2.5-1
2.6 Special Cases	2.6-1
2.7 Are the Standard Deviations Too Large?	2.7-1
2.8 References	2.8-1
PART III - A Sample Problem Notebook	
3.1 Introduction	3.1-1
3.2 MORSE Sample Problem 1	3.2-1
3.3 PICTURE Sample Problem	3.3-1
PART IV - MORSE	
4.1 Abstract	4.1-1
4.2 Introduction	4.2-1
4.3 Input Instructions	4.3-1
4.4 Random Walk Module	4.4-1
4.5 MORSEC Module	4.5-1
4.6 SAMBO - the Analysis Module	4.6-1
4.7 CG - The Combinatorial Geometry Module	4.7-1
4.8 The Diagnostic Module	4.8-1
4.9 Hardware and Software Requirements	4.9-1
4.10 The Many Integral Forms of the Boltzmann Transport Equation and its Adjoint	4.10-1
4.11 Generalized Guassian Quadrature	4.11-1
4.12 References	4.12-1

TABLE OF CONTENTS (cont'd)

	Page
PART V - Program PICTURE	
5.1 Abstract	5.1-1
5.2 Introduction	5.2-1
5.3 Routines	5.3-1
5.4 Input Data	5.4-1
5.5 Options	5.5-1

2.0-1

Part II

A Guide on How to Use the Various Codes

TABLE OF CONTENTS

	<u>Page</u>
2.1 Introduction	2.1-1
2.2 Is Core Storage Space a Problem?	2.2-1
2.3 Is Amount of Computer Time Too Large?	2.3-1
2.4 Are There Too Many Abnormal Terminations?	2.4-1
2.5 Is There Too Much Output?	2.5-1
2.6 Special Cases	2.6-1
2.7 Are the Standard Deviations Too Large?	2.7-1
2.8 References	2.8-1

2.1 Introduction

With a code system as large as MORSE, it requires a great deal of experience in using it before its potential is really appreciated. One might say that the user must acquire a "feel" for running various types of problems.

It is the intent of this part of the report to help the user acquire this "feeling" by pointing out certain situations which may arise in his application of the system to his particular problem. Knowing how to handle these situations will save him a lot of time and/or telephone calls to the author.

This report makes no pretense of solving all potential problem areas but attempts to alleviate some of the more common ones.

2.2 Is Core Storage Space a Problem?

Depending on the complexity of a problem the amount of storage required can be prohibitive. Cross sections require the largest amount of storage except when using DOMINO with a large quadrature set.

The very first solution that presents itself should be the use of program XCHEKR to pre-mix the cross sections. Input instructions for running XCHEKR appear in Section 4.3. It requires only one input card in addition to the standard MORSEC input and using it will cut your cross section storage requirements by approximately half in the typical case. Once XCHEKR has mixed the data and put it either on disk or tape, the user will know exactly how much permanent cross section storage will be required when running MORSE. Storing data only for the groups you will use not necessarily for all the groups also saves storage. Unless the user is using a next flight estimator, he should not save Legendre coefficients; i.e., he should set ISTAT = 0. Another way of saving some storage is to use dummy subroutines for the mixing routines ANGLES, GTSCT, READSG, JINPUT, BADMOM, GETMUS, FFREAD, FIND, RESTOR, STORE and GTNSDK when using premixed cross sections.

In running a DOT-DOMINO-MORSE problem, the user may find the amount of DOMINO data quite large. The MORSE routines that handle the DOMINO data in the IBM version have the option of using direct access devices to store this data. Direct access storage is chosen whenever the user's dimension in blank common is insufficient to hold all the data. It does require some user foresight though, because the user must specify the amount of space needed on the disk both on a DD card and in a DEFINE FILE statement. (See the DOMINO reports ORNL-4853 and ORNL/TM-7771).

If the user selects a set of analysis parameters which require a large amount of storage, he could run the analysis as a separate program by using the option of writing a collision tape.

Another way of saving storage is to allow fewer particles per batch (a smaller NSTRT and NMOST) since a particle bank containing 12* NMOST locations is required on the IBM machines and 14* NMOST on CDC and UNIVAC common. Cut it down to the size you need after you make the first run (which will tell you how much was required).

2.3 Is Amount of Computer Time Too Large?

There are any number of reasons why the computer time required may be quite large. The materials used may be such that the number of scatterings is tremendous. The complexity of the geometry is also a factor because of the number of boundary crossings involved. The use of a next-flight estimator requiring tracking through the geometry from the collision point to the detector point can also be quite time-consuming especially if there are many detectors. Following are some suggestions on cutting down running time.

If the geometry of the problem is such that large portions of it can be treated in two-dimensions, the user can use a DOT-DOMINO-MORSE set-up in which DOT handles the two-dimensional geometry and MORSE is coupled to it for the portion of the geometry requiring three-dimensions. This can save a significant amount of time.

Another feature which is of particular interest in reactor calculations is the "fictitious scattering"^{3,4} method of choosing path lengths. Using this method, pseudo-collision sites are chosen using the maximum total cross section in the system. A rejection technique based on the ratio of the actual cross section at the pseudo-collision to the maximum cross section is utilized to determine if this is a real collision. If the test fails, a new pseudo-collision site is chosen. This method avoids much of the geometry tracking and thereby saves time in the estimation process. Routines to handle this are available.

Sometimes tracking thermal neutrons involves a large number of collisions and therefore a significant amount of computer time. In many problems, the thermal neutrons do not make a significant contribution to the answer so the user should cut-off the particles before they reach thermal (use an energy cut-off by setting NGPQTN to exclude the thermal group).

Another technique that is useful in a case where the geometry is symmetrical is to use this symmetry to advantage by tracking through only part of the system and then properly normalizing the results.

Two other minor improvements can be made as follows. The user can increase the value of NAZ in the geometry input. This increases the size of the geometry array that saves the tracking information and thus allows more table-lookup and less actual calculation. Also since the advent of COMBGEOM or CG, there is no compelling reason for continuing to use GTMED to establish a correspondence between geometry and cross section media numbers. It is quite simple to change the one input card involved to make the correspondence one-to-one. The user could thus eliminate all calls to GTMED plus the GTMED routine and set up his geometry accordingly. This will save a number of subroutine calls. At some future date, this may be incorporated in the standard MORSE.

2.4 Are There Too Many Abnormal Terminations?

There are several aspects of any MORSE problem that can be checked for errors before actually running it on the computer. This can save time in the long run.

Whenever starting on a geometry for the first time, or after modification of it, the user should make several pictures of his mock-up using program PICTURE (see part five). He should take "cuts" through all major portions in each plane. This will reveal most errors in the geometry.

Pre-mixing cross sections by using XCHEKR¹ can also eliminate many possible trouble spots such as incorrect ID numbers. It also allows the user to look at the mixed cross sections to see if they appear to be correct prior to making a MORSE run.

Double checking the input cards is always a good practice. Are the cards in the proper order? Are any cards missing? Is the number of media the same in your input as it is on the pre-mixed data set? What about your tape or disk specifications? Have you defined your logical units and scratch disks? Did you allocate enough disk space?

If the user requires a direct access device for his DOMINO data, is his DEFINE FILE statement in agreement with the DD card he specified for it? Even if it is in agreement, is it correct?

Another major pitfall is failure to allocate enough space in blank common. Do not guess: Calculate! If XCHEKR was run, the amount of cross section storage is known. The particle bank is 12*NMOST on the IBM-360. Space for random walk input can be calculated based on Fig. 4.2. As a result of having run PICTURE, the user has enough information to make an intelligent estimate of the storage required for geometry data. Take the "Length of FPD array" *2+ "Length of integer array" + 10* number of code zones as the estimate. This will always equal or exceed what is needed. The space required for analysis data can be calculated based on Fig. 4.6. Needless to say, after one MORSE run you will know the exact amount of blank common required from your output.

2.5 Is There Too Much Output?

If the user feels that the volume of output data is excessive, he has several alternatives.

One such alternative involves pre-mixing cross sections when the same ones will be used in several production runs. This will eliminate all but one page of output from the cross sections (he will already have a complete set of output from the XCHEKR run). Even if he does not pre-mix, he can turn off all the switches in the cross section module that print data as read, moments, angles, etc.

Another precaution is to be sure that $IDBG = 0$ in the geometry input except when debugging. The amount of output produced when $IDBG \neq 0$ is voluminous.

In production runs where several hundred batches are being run, batch output can be suppressed by using a special subroutine OUTPT which is available from the author.

2.6 Special Cases

There are a number of special cases which can arise in running MORSE, DOMINO, etc. An attempt will be made to describe some of them and the action required.

Please note that when running a DOT adjoint to use in determining the important radial areas for a MORSE source or to determine importance for source energy biasing, the DOT source should be isotropic. Otherwise, improper angular distributions for an adjoint source will result.

If the individual who makes the library tape for the cross sections uses ID numbers with more than five digits, the user must alter the format in subroutine XSEC from 14I5 to something else.

Another event which has occurred is that a user may have the desire to use premixed cross sections on cards. This is accomplished by running XCHEKR, and later MORSE, using a special subroutine XSTAPE designed to read or punch cards. This routine is available from the author.

Anyone desiring to do a k-calculation only should use a dummy subroutine BANKR which will preclude doing any analysis.

Many times there is a need for some special normalization. This can be handled in several ways. One "built-in" method for detector dependent normalization is by use of the "FACT" array (see Table 4.21). This array can be set-up either in INSCOR or STRUN. The author prefers the STRUN way because INSCOR is frequently used for other things. Another way of changing the normalization is to alter WTSTRT in the input data; in other words, start the particles with a different weight.

Another very special case which requires some comment is the type in which the user is "sandwiching" MORSE between two DOT runs. That is to say, he runs a forward DOT to some boundary which is then the source for a forward MORSE run to some other boundary which will be the scoring surface based on a DOT adjoint run. It should be obvious that this type of run has lots of room for error and also that it can not all be handled with a MORSE run with in-core analysis. In this case, the user runs MORSE with the DOMINO tape from the forward DOT as his source and writes a collision tape of particles crossing the boundary where the DOT adjoint will be coupled to it. Then he runs a SAMBO analysis which reads the collision tape and uses the DOMINO tape from the adjoint DOT run as a scoring function. It would ordinarily be impossible to handle two DOMINO tapes in one run because of the number of I/O units (disks and tapes) and the amount of storage required. This necessitates a two-job set-up.

2.7 Are the Standard Deviations Too Large?

One of the most difficult things to determine is how long a computer run is needed to get acceptable statistics. After completion of a production run, the user may decide that he must have better (or lower) standard deviations. His course of action should be as follows.

The output from the run will contain the next random number in the sequence. This number should be used as the starting random number in a new run. Upon completion of this second run, he can obtain the average of the runs and also the standard deviations. This is done as follows:

Assume a_1 and a_2 are the answers from runs one and two respectively; f_1 and f_2 are the corresponding f.s.d.'s (or fractional standard deviations) as output; $\sigma_i = a_i f_i$; and c_1 and c_2 are the number of batches per run. The average answer is then $\bar{x} = \frac{a_1 * c_1 + a_2 * c_2}{c_1 + c_2}$ and the approxi-

$$\text{mate corresponding f.s.d.} = \frac{\sqrt{\left(\frac{c_1}{c_1+c_2}\right)^2 \sigma_1^2 + \left(\frac{c_2}{c_1+c_2}\right)^2 \sigma_2^2}}{\bar{x}}.$$

Any number of runs may be averaged together by using an expanded form of the above equations: i.e.,

$$\bar{x} = \frac{\sum_{i=1}^N a_i c_i}{\sum_{i=1}^N c_i} \quad \text{and}$$

$$\text{f.s.d.} = \frac{\sqrt{\sigma^2}}{\bar{x}} = \sqrt{\frac{\sum_{i=1}^N c_i^2 \sigma_i^2}{\left(\sum_{i=1}^N c_i\right)^2}} / \bar{x},$$

where N is the number of runs.

2.8 References

1. M. B. Emmett, C. E. Burgart and T. J. Hoffman, DOMINO, A General Purpose Code for Compiling Discrete Ordinates and Monte Carlo Radiation Transport Calculations, ORNL-4853 (1973).
2. M. B. Emmett, DOMINO-II, A General Purose Code for Coupling DOT-IV Discrete Ordinates and Monte Carlo Radiation Transport Calculations, ORNL/TM-7771 (1981).
3. C. E. Burgart, Capabilities of the MORSE Multigroup Monte Carlo Code in Solving Reactor Eigenvalue Problems, ORNL/TM-3662 (1971).
4. S. N. Cramer, Application of the Fictitious Scattering Radiation Transport Model for Deep-Penetration Monte Carlo Calculations, Nucl. Sci. & Eng., 65, 237-253 (1978).

TABLE OF CONTENTS

	Page
4.1. Abstract	4.1-1
4.2. Introduction	4.2-1
4.3. Input Instructions	4.3-1
4.3.1. Random Walk Input Instructions	4.3-1 †
4.3.2. Combinatorial Geometry Input Instructions.	4.3-7
4.3.3. MØRSEC - Cross Section Module Input Instructions	4.3-11 †
4.3.4. SAMBØ Analysis Input Instructions	4.3-14
4.4. Random Walk Module	4.4-1
4.4.1. Introduction	4.4-1
4.4.2. Main Program	4.4-19
4.4.3. Subroutines	4.4-21
Subroutine MØRSE	4.4-21
Subroutine DATE	4.4-23
Function DIREC	4.4-25
Subroutine EUCLID	4.4-26
Subroutine FBANK	4.4-28
Random Number Package	4.4-31
Subroutine FPRØB	4.4-33
Subroutine FSØUR	4.4-34
Subroutine GETETA	4.4-35
Subroutine GETNT	4.4-38 †
Subroutine GØMST	4.4-39 †
Subroutine GPRØB	4.4-40
Subroutine GSTØRE	4.4-41
Subroutine INPUT	4.4-42
Subroutine INPUT1	4.4-43
Subroutine INPUT2	4.4-46
Subroutine IWEEK	4.4-47
Subroutine MSØUR	4.4-48
Subroutine NXTCØL	4.4-50
Subroutine ØUTPT	4.4-52
Subroutine ØUTPT2	4.4-53

†Revised.

	Page
Subroutine SØRIN	4.4-54
Subroutine SØURCE	4.4-56
Subroutine TESTW	4.4-57
Subroutine TIMER	4.4-58
4.4.4. Energy Indexing in MØRSE	4.4-60
4.5. MØRSEC - The Cross-Section Module	4.5-1
4.5.1. Introduction	4.5-1
4.5.2. Blank Common Cross Section Storage	4.5-5
4.5.3. Subroutines	4.5-14
Subroutine ALBDØ	4.5-14 †
Subroutine ANGLES	4.5-15
Subroutine BADMØM	4.5-17
Subroutine CØLISN	4.5-18
Subroutine FFREAD	4.5-20
Subroutine FIND	4.5-21
Subroutine FISGEN	4.5-23
Subroutine GAMGEN	4.5-24
Subroutine GETMUS	4.5-25
Subroutine GTIØUT	4.5-27
Subroutine GTMED	4.5-28
Subroutine GTNSDK	4.5-29
Subroutine GTSCT	4.5-30
Subroutine JNPUT	4.5-32
Subroutine LEGEND	4.5-33
Subroutine MAMENT	4.5-35
Subroutine NSIGTA	4.5-37
Subroutine PTHETA	4.5-38
Subroutine Q	4.5-40
Subroutine READSG	4.5-41
Subroutine RESTØR	4.5-43
Subroutine STØRE	4.5-44
Subroutine XSEC	4.5-45
Subroutine XSTAPE	4.5-47
4.6. SAMBØ - The Analysis Module	4.6-1
4.6.1. Introduction	4.6-1

†Revised.

	Page
4.6.2. Bookkeeping Routines	4.6-3
Subroutine ENDRUN	4.6-13
Subroutine ENRGYS	4.6-14
Subroutine FLUXST	4.6-15
Subroutine INSCØR	4.6-17
Subroutine NBATCH	4.6-18
Subroutine NRUN	4.6-20
Subroutine SCØRIN	4.6-22 †
Subroutine STBTCH	4.6-25
Subroutine STRUN	4.6-26
Subroutine VAR2	4.6-27
Subroutine VAR3	4.6-29
4.6.3. Interface Routine	4.6-31
Subroutine BANKR	4.6-31
4.6.4. Surface Crossing Estimators	4.6-33
General Description	4.6-33
Subroutine BDRYX	4.6-36
Subroutine SDATA	4.6-37
4.6.5. Point Detector Estimators	4.6-38
General Description	4.6-38
Subroutine RELCØL	4.6-39
Subroutine SDATA	4.6-40
4.7. CG - The Combinatorial Geometry Module	4.7-1
4.7.1. Body Types	4.7-1
4.7.2. Introduction	4.7-1
4.7.3. Descriptions of Body Types	4.7-5 †
RPP	4.7-5
SPH	4.7-6
RCC	4.7-6
REC	4.7-7
TRC	4.7-7
ELL	4.7-8
WED or RAW	4.7-8
BØX	4.7-9
ARB	4.7-9

	Page	
4.7.4. Subroutines	4.7-11	
Subroutine G1	4.7-11	+
Subroutine ALBERT	4.7-13	
Subroutine GENI	4.7-14	
Subroutine GG	4.7-15	
Subroutine GTVLIN	4.7-16	
Subroutine JØMIN	4.7-17	
Subroutine LØØKZ	4.7-18	+
Subroutine NØRML	4.7-19	
Subroutine PR	4.7-20	
4.8. The Diagnostic Module	4.8-1	
4.8.1. Introduction	4.8-1	
4.8.2. Subroutines	4.8-10	
Subroutine BNKHL P	4.8-10	
Subroutine HELP	4.8-11	
Subroutine HELPER	4.8-13	
Subroutine SUBRT	4.8-14	
Subroutine XSCHLP	4.8-15	
4.9. Hardware and Software Requirements	4.9-1	
4.9.1. Hardware Requirements	4.9-1	
4.9.2. Library Routines and Functions	4.9-2	
General Comments	4.9-2	
Function ICLØCK	4.9-4	
Function ICØMPA	4.9-5	
Subroutine IDAY	4.9-7	
Function LØC	4.9-8	
Function MØDEL	4.9-8	
Masking Functions	4.9-9	
Character Manipulation Routines	4.9-12	
Subroutine ERTRAN	4.9-15	
Function TICKER	4.9-15	
Subroutine SECØND	4.9-15	
4.9.3. Overlay Structure	4.9-16	
4.10. The Many Integral Forms of the Boltzmann Transport Equation and its Adjoint	4.10-1	+
4.11. Generalized Gaussian Quadrature	4.11-1	
4.12. References	4.12-1	+

LIST OF TABLES

	Page
Table 4.1. Sample Group Input Numbers for Some Representative Problems	4.3-2
Table 4.2. Variables That May be Written on Collision Tape.	4.3-5
Table 4.3. Input Required on CGB Cards for Each Body Type	4.3-9
Table 4.4. Definition of Variables in Common APØLLØ	4.4-4
Table 4.5. Definition of Variables in Common NUTRØN	4.4-10
Table 4.6. List of Routines Altering Variables in NUTRØN Common	4.4-11
Table 4.7. Subroutines in Which Variables in NUTRØN Common are Changed	4.4-12
Table 4.8. Definition of Variables in Random Walk Blank Common	4.4-13
Table 4.9. Indexing of Random Walk Blank Common Arrays	4.4-16
Table 4.10. Layout of Particle Bank in Blank Common	4.4-18
Table 4.11. Layout of Fission Bank in Blank Common	4.4-29
Table 4.12. Definitions of Variables in Common FISBNK	4.4-30
Table 4.13. Numerical Examples for Various Options with Corresponding Energies for Each Case	4.4-61
Table 4.14. Values of NGPQT1, NGPQT2, NGPQT3 for Several Cases	4.4-62
Table 4.15. Definitions of Variables in Common LØCSIG	4.5-7
Table 4.16. Location of Permanent Cross Sections in Blank Common	4.5-11
Table 4.17. Definitions of Variables in Common GTSC1	4.5-13
Table 4.18. Definitions of Variables in Common USER	4.6-7
Table 4.19. Definitions of Problem-Dependent Energy Group Limits	4.6-8
Table 4.20. Definitions of Variables in Common PDET	4.6-9
Table 4.21. Indexing of Analysis Arrays in Blank Common	4.6-11
Table 4.22. BANKR Arguments	4.6-30
Table 4.23. Definitions of Variables in Common GØMLØC	4.7-26
Table 4.24. Definitions of Variables in Common PAREM as Found in Combinatorial Geometry	4.7-28
Table 4.25. Definitions of Variables in Common ØRGI	4.7-30 †
Table 4.26. Diagnostic Messages from INPUT Module in MØRSE	4.8-2 †
Table 4.27. Diagnostic Messages from Other Modules of MØRSE.	4.8-6 †

†Revised.

CARDS M (7E10.4) (Omit if MFISTP on Card L $\leq$ 0)

(FWLØ(I), I = 1, MXREG) values of the weight to be assigned to fission neutrons.

CARDS N (7E10.4) (Omit if MFISTP on Card L $\leq$ 0)

(FSE(IG,IMED), IG = 1, NMGP), IMED = 1, MEDIA) the fraction of fission-induced source particles in group IG and medium IMED.

NOTE: Input for each medium must start on a new card.

CARDS O (7E10.5) (Omit if NGPQTN = 0 or NGPQTG = 0, i.e., include if coupled neutron-gamma-ray problem)

((CWLØ(IG,NREG) IG = 1, NMGP or NMTG - NMGP), NREG = 1, MXREG) - values of the probability of generating a gamma ray. NMGP groups are read for each region in a forward problem and NMTG-NMGP for an adjoint. Input for each region must start on a new card.

4.3.2. Combinatorial Geometry Input Instructions

The combinatorial geometry input data is read by the JØMIN subroutine, except for the region volumes VNØR(I), which are read by the GTVLIN subroutine whenever IVØPT = 3. For clarity of terminology, the terms "regions" and "media" have essentially the same meaning as in the Ø5R Geometry Package, but are constructed in a different manner. The term "zone" is the same as the "region" as defined in the original combinatorial geometry package. The term "body" has the same meaning as in the original combinatorial geometry package.

CARD CGA (2I5,10X,10A6)

IVØPT - option which defines the method by which region volumes are determined; if

IVØPT = 0, volumes set equal to 1,

IVØPT = 1, concentric sphere volumes are calculated,

IVØPT = 2, slab volumes (1-dim.) are calculated,*

IVØPT = 3, volumes are input by card.

IDBG - if IDBG > 0, subroutine PR is called to print results of combinatorial geometry calculations during execution. Use only for debugging.

JTY - alphanumeric title for geometry input (columns 21-80).

*Not operational.

CARDS CGB (2X,A3,1X,I4,6D10.3)

One set of CGB cards is required for each body and for the END card (see Table 4.3). Leave columns 1-6 blank on all continuation cards.

- ITYPE - specifies body type or END to terminate reading of body data (for example BØX, RPP, ARB, etc.). Leave blank for continuation cards.
- IALP - body number assigned by user (all input body numbers must form a sequence set beginning at 1). If left blank, numbers are assigned sequentially. Either assign all or none of the numbers. Leave blank for continuation cards.
- FPD(I) - real data required for the given body as shown in Table 4.3. This data must be in cm.

CARDS CGC (2X,A3,I5,9(A2,I5))

Input zone specification cards. One set of cards required for each input zone, with input zone numbers being assigned sequentially.

- IALP - IALP must be a nonblank for the first card of each set of cards defining an input zone. If IALP is blank, this card is treated as a continuation of the previous zone card.

IALP - END denotes the end of zone description.

- NAZ - total number of zones that can be entered upon leaving any of the bodies defined for this input region (some zones may be counted more than once). Leave blank for continuation cards for a given zone. (If $NAZ \leq 0$ on the first card of the zone card set, then it is set to 5). This is used to allocate blank common.

Alternate IIBIAS(I) and JTY(I) for all bodies defining this input zone.

- IIBIAS(I) - specify the "ØR" operator if required for the JTY(I) body.

- JTY(I) - body number with the (+) or (-) sign as required for the zone description.

CARDS CGD (14I5)

MRIZ(I) - MRIZ(I) is the region number in which the "Ith" input zone is contained (I = 1, to the number of input zones). Region numbers must be sequentially defined from 1.

CARDS CGE (14I5)

MMIZ(I) - MMIZ(I) is the medium number in which the "Ith" input zone is contained (I = 1, to the number of input zones). Medium numbers must be sequentially defined from 1.

CARDS CGF (7D10.5) (Omit if IVØPT ≠ 3)

VNØR(I) - volume of the "Ith" region (I = 1 to MXREG, the number of regions).

4.3.3. MØRSEC - Cross-Section Module Input Instructions

CARD XA (20A4)

Title card for cross sections. This title is also written on tape if a processed tape is written; therefore, it is suggested that the title be definitive.

CARD XB (14I5)

NGP* - the number of primary groups for which there are cross sections to be stored. Should be same as NMGP input in MØRSE.

NDS - number of primary downscatters for NGP (usually NGP).

NGG* - number of secondary groups for which there are cross sections to be stored.

NDSG - number of secondary downscatters for NGG (usually NGG).

INGP* - total number of groups for which cross sections are to be input.

ITBL - table length, i.e., the number of cross sections for each group (usually equal to number of downscatters + number of upscatters + IHT).

ISGG - location of within-group scattering cross sections (usually equal to number of upscatters plus IHT+1).

NMED - number of media for which cross sections are to be stored - should be same as MEDIA input in MØRSE.

*See Table 4.1 for sample input.

NELEM	- number of elements for which cross sections are to be read.
NMIX	- number of mixing operations (elements times density operations) to be performed (must be ≥ 1).
NCØEF	- number of coefficients for each element, including P_0 .
NSCT	- number of discrete angles (usually $NCØEF/2_{\text{integral}}$).
ISTAT	- flag to store Legendre coefficients if greater than zero.
IHT	- location of total cross section in the table.
CARD XC (11I5)	
IRDSG†	- switch to print the cross sections as they are read if > 0 .
ISTR†	- switch to print cross sections as they are stored if > 0 .
IFMU†	- switch to print intermediate results of μ 's calculation if > 0 .
IMØM†	- switch to print moments of angular distribution if > 0 .
IPRIN†	- switch to print angles and probabilities if > 0 .
IPUN†	- switch to print results of bad Legendre coefficients if > 0 .
IDTF†	- switch to signal that input format is DTF-IV format if > 0 ; otherwise, ANISN format is assumed.
IXTAPE	- logical tape unit if binary cross section tape, set equal to 0 if cross sections are from cards. If negative, then the processed cross sections and other necessary data from a previous run will be read; in this case ($IXTAPE < 0$) no cross sections from cards and no mixing cards may be input. The absolute value of IXTAPE is the logical tape unit.
JXTAPE	- logical tape unit of a processed cross-section tape to be written. This processed tape will contain the title card, the variables from common LØCSIG and the pertinent cross sections from blank common

†Switches are ignored if IXTAPE < 0 .

IØ6RT - logical tape unit of a point cross-section tape in 06R format.

IGQPT - last group (MØRSE multigroup structure) for which the 06R point cross sections are to be used ($\leq$ NMGP).

CARD XD (14I5) (Omit if IXTAPE $\leq$ 0)

Element identifiers for cross-section tape. If element identifiers are in same order as elements on tape, the efficiency of the code is increased due to fewer tape rewinds.

CARDS XE (Omit if IXTAPE $\neq$ 0)

If cross sections are in free-form, a card with ** in columns 2 and 3 must precede the actual data.

ANISN format if IDTF $\leq$ 0; otherwise, DTF-IV format. Cross sections for INGP groups with a table length ITBL for NELEM elements each with NCØEF coefficients.

CARDS XF (2I5,E10.5) (Omit if IXTAPE $<$ 0)

NMIX (see Card XB) cards are required.

KM - medium number.

KE - element number occurring in medium KM (negative value indicates last mixing operation for that medium).
Failure to have a negative value causes code not to generate angular probabilities for that media (LEGEND and ANGLE not called).

RHØ - density of element KE in medium KM in units of atoms/(barn cm).

CARDS XG (I5) (Omit if IØ6RT $\leq$ 0)

NXPM - number of point cross-section sets per medium found on an 06R^{7,8} tape.

= 1, total cross section only,

= 2, total + scattering cross section,

= 3, total, scattering, and v *fission cross section.

NOTE: Cross sections and cross-section input data may be checked independently of MØRSE utilizing XCHEKR.⁹ The input to XCHEKR consists of the cross-section cards XA through XG preceded by a card as follows:

Format (4I5)

IADJM - set greater than zero for an adjoint problem.

- MEDIA - number of cross-section media; should equal NMED on Card XB.
- NMGP - number of primary particle energy groups for which cross sections are to be stored; should equal NGP on Card XB.
- NMTG - total number of energy groups for which cross sections are to be stored. Should be equal to INGP on Card XB.
-

4.3.4. SAMBØ Analysis Input Instructions

The following data are read from cards by SCØRIN:

CARD AA (20A4)

Title information - will be immediately output.

CARD BB (8I5)

- ND - number of detectors (set = 1 if ≤ 0).
- NNE - number of primary particle (neutron) energy bins to be used (must be $\leq$ NE).
- NE - total number of energy bins (set = 0 if ≤ 1).
- NT - number of time bins for each detector (may be negative, in which case $|NT|$ values are to be read and used for every detector) (set = 0 if $|NT| \leq 1$).
- NA - number of angle bins (set = 0 if ≤ 1).
- NRESP - number of energy-dependent response functions to be used (set = 1 if ≤ 0).
- NEX - number of extra arrays of size NMTG to be set aside (useful, for example, as a place to store an array of group-to-group transfer probabilities for estimator routines).
- NEXND - number of extra arrays of size ND to be set aside (useful, for example, as a place to store detector-dependent counters).

CARDS CC (3E10.4) (ND cards will be read)

- X,Y,Z - detector location. (If other than point detectors are desired, the point locations must still be input and can be combined with additional data built in to user routines to fully define each detector.)

Subroutine FPRØB

FPRØB calculates the expected weight of fission neutrons at a collision point and then splits or plays Russian roulette so as to produce the correct average number of fissions, all of weight FWLØ (specified in problem input for each region). FBANK is called for each neutron produced, to be stored for processing in the next generation. FPRØB calls FISGEN for forward fission problems. For adjoint problems, $\overline{\nu}\Sigma_f/\Sigma_t$ is selected from the FISH array in blank common.

Called from: MØRSE.

Subroutines called: FISGEN, GTMED, BANKR(3), FBANK, FLTRNF, HELP, ERRØR, SIGN (library).

Commons required: Blank, NUTRØN, APØLLØ, FISBNK.

Variables required:

NMED, WATE, NREG, IG (from NUTRØN common, see page 4.4-10)
 IADJM - =1 if adjoint problem;
 =0 if forward problem.
 LØCFNS - location in blank common of cell zero of array of
 fission cross sections.
 LØCNCS - location in blank common of cell zero of scattering
 counter arrays.
 IMED - cross-section medium of collision point.
 MXREG - maximum region number.
 NMTG - total number of energy groups.
 FTØTL - total of fission weights from all collisions.
 LØCFWL - location in blank common of cell zero of array FWLØW.
 NPSCL(3) - fission counter.

Variables changed:

WATEF - fission weight transferred to FBANK.
 NFIZ - actual number of fissions per group and region.
 WFIZ - weight equivalent to NFIZ.
 FTØTL
 NPSCL(3)

Significant internal variables:

FWL - current value from array FWLØ.
 ISCT - location in blank common of (IG,NREG) cell of scattering
 counter array NFIZ (and later WFIZ).

Subroutine FSØUR

This routine is called by the source executive routine, MSØUR, when the source for the present batch is to be taken from the previous batch fissions. Its function is to transfer the neutron parameters from the fission bank to the neutron bank. If there were no fissions in the previous batch, it sets a flag, prints a message, and returns.

Called from: MSØUR

Subroutines called:

STØRNT(N) - loads parameters in common NUTRØN into the Nth location in the neutron bank.

Commons required: Blank, NUTRØN, FISBNK, APØLLØ.

Variables required:

NFISH - number of fissions produced in the previous batch.
 NMEM - set equal to NFISH.
 NITS - number of batches requested for the run.
 ITERS - batch counter.
 NFISBN - location in blank common of cell zero of the fission bank.

Variables changed:

NITS - set to number of batches completed if NFISH = 0.
 ITERS - set to zero if NFISH = 0.

NAME	}	Set to zero (in NUTRØN common, see page 4.4-10).
NMED		
NREG		
U, V, W		
BLZNT		
X, Y, Z	}	Set to values found in fission bank (in NUTRØN common, see page 4.4-10).
WATE		
AGE		
IG		
NAMEX		

NOTE: IG is group index of neutron causing fission.

Function used: DIREC. EXPRNF, AMØD(library), EXP(library)

Commons required: Blank, NUTRØN, APØLLØ.

Variables required:

IG, X, Y, Z, U, V, W, WATE, NREG (from NUTRØN common, see page 4.4-10).

MAXGP - number of energy groups for weight standards and/or
path-length stretching parameters PATH.

NØLEAK - an index for nonleakage biasing.

RAD - the largest overall dimension in the system.

PATH - path-length stretching parameters (in blank common).

Variables changed:

ETA - the number of mean-free paths to the next collision.

WATE - the particle's weight corrected for the biasing
employed during the present flight selection.

Significant internal variables:

ARG - the distance in mean-free paths from the last collision
site to an external boundary along the present flight
direction.

Subroutine GETNT(N)

Three entry points are used in this routine.[†] Entry SETNT saves the address (in words) of the first cell available for the neutron bank in blank common and returns the address of the last cell it will use. Entry STØRNT(N) stores values from common NUTRØN into the Nth set of locations in the neutron bank and entry GETNT(N) does the reverse; it picks up variables from the bank and puts them in common NUTRØN.

Called from: INPUT2 (SETNT), MØRSE (GETNT), MSØUR (STØRNT), FSØUR(STØRNT), ØUTPT (GETNT), TESTW(STØRNT).

Commons required: Blank, NUTRØN. The area of blank common used for the neutron bank is shown in Table 4.10. Notice that IG, NAME, NAMEX, NMED, and NREG are stored in 2-byte words (and are therefore limited to ≤ 65535) on the IBM-360, but are full words on the UNIVAC-1108 and CDC-6600.

Variables required:

SETNT: NLAST

NMØST

GETNT: N

STØRNT: N, NAME, NAMEX, NMED, NREG, IG, U, V, W, X, Y, Z, WATE, BLZNT, AGE (from NUTRØN common, see page 4.4-10).

Variables changed:

SETNT: NLAST

GETNT: variables in common NUTRØN required by SETNT above.

STØRNT: 12 locations in blank common.

Significant internal variables:

NNØ - location in blank common of start of neutron bank.

[†]On the CDC-6600 3 separate subroutines exist.

Subroutine GØMST (TSIG, MARK)

Any boundary crossings between the present and next collision sites are determined by calling the combinatorial geometry routine G1. Before the G1 call, combinatorial geometry variables in common PAREM are initialized; and after the call, NUTRØN variables are updated. If an albedo medium is encountered, the flag NALB is set to the albedo medium number, and then NØRML is called to determine the normal to the surface encountered. MARK is set to -1 for an external void.

Called from: NXTCØL.

Subroutines called: G1, NØRML, PR.

Commons required: APØLLØ, NUTRØN, GØMLØC, ØRGI, PAREM.

Variables required:

X, Y, Z, NMED, U, V, W, NREG} from NUTRØN common, see page 4.4-10.
 IBLZN - value of IR from last track or from LOOKZ.
 ETATH - distance to be traveled (in cm) if the flight remains in the starting medium.
 MARK - initial value of flag indicating type of trajectory.
 TSIG - total cross section of starting point medium.
 DIST - present distance from XB(3).

Variables changed:

X, Y, Z - end point of flight.
 NALB - albedo flag (= MEDALB or 0).
 MARK - flag indicating type of termination of flight,
 0 - normal boundary crossing,
 1 - flight within one medium,
 -1 - particle escaped,
 -2 - particle entered internal void.
 NMED - medium of end point.
 NREG - region of end point.
 ETAUSD - actual flight distance (in m.f.p.).
 BLZNT - combinatorial geometry region of end point.
 ETATH - actual flight distance (in cm).
 DISTO - distance from XB(3) to next collision site.
 IR }
 XB } - set to values from NUTRON Common.
 WB }

Subroutine GPRØB

This subroutine is the executive routine for the generation and storage of secondary gamma rays (or neutrons for an adjoint, coupled problem). The version of GPRØB which is found to be most useful uses GWL as the probability of generating a gamma ray. Thus, a random number is compared with GWL, and if greater, no gamma ray is generated; if less than or equal, then a gamma ray with weight = $WATE * PGEN / GWL$ is stored. This procedure produces gamma rays of varying weights, but the number of gamma rays may be controlled easily. This version is the one distributed to users.

Another version of GPRØB which has been implemented in some cases uses GWL as a desired gamma weight. The probability of generating a gamma ray is determined and the resulting gamma-ray weight, WATEG, is compared with input values of the desired gamma-ray weight, GWL. Russian roulette and splitting are used to produce gamma rays of weight GWL. That is, if the gamma-ray weight is less than the input values, then the gamma ray is killed with probability $(GWL - |WATEG|) / GWL$ and stored with probability $(|WATEG|) / GWL$. If the gamma-ray weight is greater than the input value, then there are $J = WATEG / GWL$ gamma rays stored with weight GWL with Russian roulette played with the remaining gamma ray of weight $WATEG - J * GWL$. Called from: MØRSE.

Subroutines called: GAMGEN, GSTØRE, FLTRNF.

Commons required: Blank, NUTRØN, APØLLØ.

Variables required:

IG	- primary particle energy group.
NMED	- geometry medium.
WATE	- primary particle weight.
GWL	- input weight values for gamma rays.
NREG	- geometry region.
NMTG	- total number of particle groups.
MXREG	- number of regions for which there are weight standards.

Significant internal variables:

WATEG	- gamma-ray weight.
PGEN	- gamma-ray generation probability.

Function I WEEK (MØNTH, IDAT, IYEAR) - IBM-360 Version*

This routine will look up the date for you if you don't know it and fill in integer values for MØNTH, IDAT, and IYEAR (requested with MØNTH ≤ 0). It also returns, as the function value, an integer from 1 to 7 representing the day of the week. If it is given a positive value of MØNTH, it assumes you have given it a month, day of month, and year and will not disturb these but will simply determine the day of the week. If you stump it (by specifying a year before 1901 or after 2099) I WEEK is returned as zero.

Called by: DATE

Routines called:

IDAY - library routine at ORNL; the output is two 4-byte words containing 8 EBCDIC characters representing the number of the month, a hyphen, the day of the month, a hyphen, and the last two digits of the year. That is, on May 30, 1970, the argument for IDAY will return containing the EBCDIC representation of 05-30-70.

Variables required:

MØNTH ≤ 0 - flag to calculate MØNTH, IDAT, and IYEAR.
 > 0 - flag to leave arguments alone.

Variables modified:

MØNTH - integer representing month.
 IDAT - integer representing day of month.
 IYEAR - integer representing year.
 I WEEK - integer representing day of week.

*This routine is a dummy routine in the CDC-6600 and UNIVAC-1108 versions.

Subroutine MSØUR

MSØUR is the executive routine for the generation and storage of the source parameters at the starting of each batch. The source parameters may be read into INPUT on cards, generated by subroutine SØURCE or obtained from the fission bank for a multiplying system. For either type of problem the calculations by subroutine SØURCE override the fission bank input or the values read from cards. If the direction cosines are all input as zero, an isotropic source direction is generated. The group number obtained from the fission bank is the group causing fission and may be used in the selection of the source group for the fission neutrons. FSE in blank common contains the group distribution for each medium.

Called from: MØRSE

Subroutines called: FSØUR, GETNT, SØURCE, GTISØ, STØRNT, BANKR(1), LØØKZ

Commons required: NUTRØN, FISBNK, APØLLØ, GØMLØC, ØRGI

Variables required:

- ITSTR - an index which determines if the source should be obtained from the previous batch fissions (ITSTR $\neq$ 0) or generated by SØURCE or from input data (ITSTR = 0).
- ISØUR - an index which determines the options for the energy distribution of the source. If ISØUR > 0 the source energies are all generated in energy group ISØUR. If ISØUR < 0, or if ISØUR = 0 and NGPFS $\neq$ 0, subroutine INPUT1 calls SØRIN and the energy is selected by SØURCE.
- NMEM - the number of particles to be generated for the batch, = NSTRT for non-fissioning systems and NFISH for multiplying systems.
- XSTRT, YSTRT, ZSTRT } starting parameters input from cards,
WTSTRT, AGSTRT } from common APØLLØ, see page 4.4-4 .
UINP, VINP, ZINP }

Variables changed:

- UØLD, VØLD, WØLD, ETATH, XØLD, YØLD, ZØLD, IBLZØ, ETA, IGØ,
MEDØLD, ØLDAGE - previous collision parameters are zeroed for the source.
- ØLDWT - previous collision weight set equal to WTSTRT.

X, Y, Z, WATE, AGE, NAMEX,	}	parameters set for each particle
IBLZN, NREG, NMED, NAME,		generated, put in NUTRON common,
U, V, W, IG		see page 4.4-10.

NPSCL(1) - counter for number of sources.

NEWNM - set to name of last particle generated.

FTOTL	}	zeroed for the next batch.
FWATE		
NFISH		

NMED	}	defined based on NMEDG in ORGI common.
NREG		

Subroutine NXTCOL

This subroutine is called by the main program to determine the spatial coordinates, the block and zone number, particle's age, and non-absorption probability at the next collision site and at every boundary crossing encountered along the way. The total number of boundary crossings is recorded as is the number of escapes. If a particle escapes, its weight is set equal to zero and the history will be terminated by the main program.

Called from: MØRSE

Subroutines called: GETETA, NSIGTA, GØMST, BANKR(7), BANKR(8)

Commons required: Blank, NUTRON, APØLLØ

Variables required:

AGE - chronological age of the particle at the previous collision site.

IBLZN - an integer specifying the zone number at the previous collision site.

NMED - the medium number at the previous collision site.

XØLD, YØLD, ZØLD - spatial coordinates at the previous collision site.

UØLD, VØLD, WØLD - the particle's precollision direction cosines.

TSIG - total cross section.

Variables changed:

AGE - chronological age at new collision site.

IBLZN - an integer containing the zone number at the new collision site.

NMED - end-of-flight medium.

NPSCL(7) - total number of boundary crossings.

NPSCL(8) - number of escapes.

X, Y, Z - end-of-flight spatial coordinates.

WATE - weight of particle undergoing flight to the new collision site.

Significant internal variables:

MARK - an index which identifies the type of event at (X,Y,Z);
 MARK = 0, normal boundary crossing; MARK = 1, flight ended within the medium; MARK = -1, particle escaped; MARK = -2, particle entered an interior void.

Table 4.17. Definitions of Variables in Common GTSC1

Variable	Definition
NPT(16)	Number of points per supergroup for each medium. (Only 16 media are allowed.) Redefined at end of GTSC1 as the total storage required for each cross section for each medium.
NXSECT(17)	Starting location of point cross sections for each medium.
NDSGP	Total storage required per supergroup for all media.
I1, I0	Input and output logical unit numbers.
IØ6RT	Logical tape number of Ø6R point cross-section tape.
IGQPT	Last MØRSE multigroup for which point cross sections will be used ($\leq$ NMGP)
NEGPS	Number of supergroup boundaries.
ESPD(100)	Energy boundaries (in eV) of the supergroups (only 100 supergroups are allowed).
NNIC	Starting location of the point cross sections.
NXPM	Number of cross sections per medium = 1, if total only = 2, if total + scattering = 3, if total, scattering, and ν *fission
NINC	Index for locating point cross sections. Used as flag in FISGEN to determine if transport process has begun.

4.5.3 SubroutinesSUBROUTINE ALBDO(IG,U,V,W,WATE,NMED,NREG)

This routine is called upon encountering an albedo scattering surface and provides the outgoing neutron parameters for the albedo collision.

The sample routine performs specular reflection at the albedo scattering surface. The requirements of specular reflection may be written as

$$I \cdot N = -R \cdot N,$$

and

$$I \times N = R \times N,$$

where I is the incoming neutron direction vector,

R is the reflected neutron direction vector, and

N is the outward normal to the surface ($I \cdot N < 0$).

Manipulation of the above two equations results in

$$R = I - 2(I \cdot N)N.$$

Called from: MØRSE

Commons required: NUTRØN, NØRMAL

Variables required: U, V, W (from common NUTRØN, see page 4.4-10)

UNØRM, VNØRM, WNØRM - components of unit vector normal to boundary.

Variables changed: U, V, W

CARDS CC (3E10.4) (ND cards will be read)

X,Y,Z - detector location. (If other than point detectors are desired, the point locations must still be input and can be combined with additional data built in to user routines to fully define each detector.)

Note that the distance between the above points and the XSTRT, YSTRT, ZSTRT values and the initial age, AGSTRT (see common USER writeup, page 4.6-7) will be used to define the lower limit of the first time bin.

CARD DD (20A4)

Units for SUD and SSD array printouts. Will be used in columns 54 through 133 of the title for the print of these arrays.

CARD EE (20A4)

Identification for each response function. Will be used as title in printout.

CARDS FF (7E10.4)

Response function values. NMTG values will be read in each set of FF cards. Input order is from energy group 1 to NMTG (order of decreasing energy).

Note: Cards EE and FF are read in the following order: EE,FF1,...FFN, EE,FF1,...FFN, etc.

CARD GG (20A4) (omit if $NE \leq 1$)

Units for title of SQE array printout.

CARDS HH (14I5) (omit if $NE \leq 1$)

Energy group numbers defining lower limit of energy bins (in order of increasing group number). The NNE (if > 0) entry must equal NGPQTN; the NE entry must be set to NMGP + NGPQTG for a combined problem; or else NGPQTG or NGPQTN.

CARD II (20A4) (omit if $NT \leq 1$)

Units for title of SQT array printout.

CARD JJ (20A4) (omit if $NT \leq 1$ or $NE \leq 1$)

Units for title of SQTE array printout.

CARDS KK (7E10.4) (omit if $|NT| \leq 1$)

NT values of upper limits of time bins for each detector (in order of increasing time and detector number). The values for each detector must start on a new card. $|NT|$ values only are read if NT is negative. They are then used for every detector.

CARD LL (20A4) (omit if $NA \leq 1$)

Units for title of SQAE array printout.

CARD MM (7E10.4) (omit if $NA \leq 1$)

NA values of upper limits of angle bins (actually cosine bins; the
N_A value must equal one).

Subroutine SCORIN with Free-Form SAMBO Analysis Input

Subroutine SCORIN has been modified to allow reading the analysis input data, other than title cards, in free-form using subroutine FFREAD which was already being used for cross section input. Since more options are allowed in the analysis free-form data, a new routine named RFRE was written which calls FFREAD to read each array to determine whether it is integer or floating point and to properly store the data. RFRE is described below.

This new feature can be very useful when a response function is all one's or when using a large number of energy bins.

Old SAMBO input decks are compatible only if

1. there are no spaces within each number,
2. blanks are not used for zeroes (blanks are ignored by FFREAD), and
3. user places a card containing either a \$\$ or an ** in Column Two and Three in front of the array to specify integer or floating point.

Subroutine RFRE(N5,N6,ARRAY,IARR,NV)

This routine reads in the arrays previously contained in Cards BB, CC, FF, HH, KK and MM. The data is still read in the same order as described on pages 4.3-14 to 4.3-16 of the MORSE report, ORNL-4972. It determines whether each array is integer or floating point by checking for \$\$ or ** in front of each array. RFRE accepts the options I, Z, and R described in Ref. 13 of the MORSE report and processes them accordingly.

Called from: SCORIN

Subroutines called: FFREAD, FLOAT

Variables required:

N5, N6 - Logical unit numbers of I/O tapes.

NV - The number of values to be read.

Variables changed:

ARRAY - An array of floating point data which is used only if the the input numbers are floating point.

IARR - An array of integer data used only if the input numbers are integers.

4.6.6. Klein Nishina Option in MORSE

The MORSE cross section package (MORSEC) has been modified to allow for reading, storing and processing gamma ray pair production and Compton scattering cross sections. This is necessary in order to use the Klein Nishina formula for making estimates in RELCOL.

Previously, use of this estimator in MORSE was accomplished by means of the OGRE cross section package being called to get the pair production and Compton scattering data. It was decided that a more automatic process based on the same ENDF libraries used in MORSE neutron calculations would be a better solution, making such calculations less user-dependent.

The MORSEC routines that required modification were XSEC, JNPUT, STORE, and NSIGTA. Both temporary and permanent cross section storage had to be allocated for the two new cross sections-pair production and Compton scattering. MORSEC had two areas of storage in blank common that were not allocated for any data. The pointers to these areas were the variables INGN and INGNP. In order to avoid adding new commons or new variables to old commons, these pointers were used - INGNP for temporary storage location and INGN for permanent. This allocation is made in XSEC only if the user indicates the presence of this data on his cross section tape by setting NCOEF = -N where N is the number of coefficients. The value of IHT, the position in the table of the total cross section is now an input number in XSEC, having a default value of three which was the original position. IHT should appear in columns 66-70 (right-adjusted) of Card B of the cross section input. This change has been in use for sometime at Oak Ridge but has not previously been announced to the public. This also necessitated changing some of the indexing on output tables to make them correspond to a variable IHT rather than a fixed IHT=3. The pair production, Σ_{pp} , data should be in position IHT-4; and the Compton scattering data, Σ_{cs} , in IHT-3 of the cross section table (IHT is the position of the total cross section). Temporary storage is handled by subroutine STORE. In JNPUT the pair production and Compton scattering data must be mixed just like the other cross sections and placed in permanent storage. For a forward problem, JNPUT also checks to be sure that

$$2*\Sigma_{pp} + \Sigma_{cs} = \Sigma_s$$

where Σ_s is the scattering cross section; if it doesn't, an error message is printed before continuing. Subroutine NSIGTA was modified by the addition of a new entry point, KLEINA, which looks up the proper Σ_{pp} and Σ_{cs} for each gamma group. The calling sequence is

CALL KLEINA(IG,NMED,PAIRR,COMR)

where PAIRR is Σ_{pp} and COMR is Σ_{cs} . This routine must be called by the user from his estimator routine and only for gamma groups.

A sample user routine, RELCOL, has also been written; but some of the burden of making sure it will work for the user's problem falls on the user. For example, users who run combined neutron-gamma problems as all primary particles must specify internal to RELCOL just how many groups are actually neutrons. This is necessary in order to insure that KLEINA is called only for gamma rays and not for neutrons. Estimates for neutrons do not involve the Klein Nishina formula but are done as the usual point-detector next-flight estimator. RELCOL is somewhat like two separate routines because it uses two entirely different methods of making estimates for neutrons and gammas. For gamma groups, a choice is made between pair production and Compton estimates. A complete description of this RELCOL follows.

Subroutine RELCOL

The fluence estimate for neutron groups, from the normal collisions in RELCOL, is given by

$$\frac{WATE \cdot e^{ARG} \cdot P(THETA, IGO)}{R^2}$$

where WATE is the statistical weight of the particle leaving the collision, ARG is the negative of the number of mean free paths from collision to detector, R is the distance from collision to detector, P(THETA, IGO) is the probability, per steradian, for a particle with energy in the incoming group IGO scattering to a lower energy group through angle $\cos^{-1}(THETA)$ and THETA is the cosine of the angle between the incoming direction and the line from the collision site to the point detector. PTHETA generates the P array.

The fluence estimate for gamma groups, from the normal collisions in RELCOL, has several possible forms. For a forward problem, a choice is made between Compton and Pair Production estimates by selecting a random number, R1, and using a Compton scattering estimate if

$$R1 < \frac{\Sigma_c}{2\Sigma_p + \Sigma_c} ;$$

otherwise Pair Production is used. The Compton scattering estimate is

$$\frac{WATE \cdot e^{ARG} \cdot PMU}{R^2}$$

where WATE, ARG and R are as described above and PMU is probability of scattering calculated by the usual Klein Nishina formulas.¹⁷

The Pair Production estimate is $\frac{WATE \cdot e^{ARG}}{4\pi R^2}$ and it is made to group IZ, the one containing .511 MeV.

Called by: User at real collisions (in Subroutine BANKR when NBNKID=5).

Routines called:

NSIGTA - Looks up total cross section and non-absorption probability.

KLEINA - For gamma groups, looks up pair production and Compton scattering cross sections.

SQRT, EXP - Library functions.

PTHETA - Calculates probability P of scattering.

EUCLID - Determines ARG.

FLUXST - Stores estimates in appropriate bins.

FLTRNF - Selects random number R where $0 \leq R \leq 1$.

Commons required: USER, PDET, NUTRON, APOLLO, Blank

Variables required:

X,Y,Z - Collision site.

NMED - Medium number where X, Y, Z is located.

IGO - Incoming energy index.

LOCXD - Location in blank common of cell zero of detector position.

LOCRSP - Location in blank common of cell zero of response function position.

NMTG - Number of energy groups.

LOCRSP - Location in blank common of cell zero of response function position.

NMTG - Number of energy groups.

NRESP - Number of response functions.

ND - Number of detectors.

IADJM - Signal for adjoint problem (= 1 if adjoint; = 0, otherwise).

NGPQTN - Number of primary energy groups being followed.

MEDIA - Number of media in problem.

NOPQT1)

NOPQT2) - See p. 4.4-6 of ORNL-4972.

NOPQT3)

WATE - Statistical weight leaving X, Y, Z.

AGE - Chronological age at X, Y, Z.

NEX = 1 for combined forward - used for probabilities stored by PTHETA.
= 0 for gamma only forward.

NEXND = 1 used for counter for number of times RELCOL gets called per detector.

The geometry must be specified by establishing two tables. The first table describes the type and location of the set of bodies used in the geometrical description. The second table identifies the physical zones in terms of these bodies. The input routine processes these tables to put the data in the form required for ray tracing. Because the ray tracing routines cannot track across the outermost body, all of the zones must be within a surrounding external void so that all escaping particles are absorbed. Also no point may be in more than one zone. This geometry package is in double precision on the IBM-360 system but remains in single precision on UNIVAC-1108 and CDC-6600 versions. Because of the change in precision, variables in commons PAREM and ØRGI had to be reordered in the IBM-360 version. Media number 1000 is an internal void, and media number 0 is an external void.

4.7.3. Description of Body Types

The information required to specify each type of body is as follows:

a. Rectangular Parallelepiped (RPP)

Specify the minimum and maximum values of the x, y, and z coordinates which bound the parallelepiped.

Fig. 4.9. Rectangular Parallelepiped (RPP).

b. Sphere (SPH)

Specify the vertex $\underline{V}$ at the center and the scalar, R , denoting the radius.

ORNL-DWG 74-6763

Fig. 4.10. Sphere (SPH).

c. Right Circular Cylinder (RCC)

Specify the vertex $\underline{V}$ at the center of one base, a height vector, $\underline{H}$, expressed in terms of its x , y , and z components, and a scalar, R , denoting the radius.

ORNL-DWG 74-6764

Fig. 4.11. Right Circular Cylinder (RCC).

4.7.4. SubroutinesSubroutine G1 (S, MA, FPD, LØCREG, NUMBØD, IRØR, IR1, IR2)

G1 is the control routine for the combinatorial geometry. On one call, it calculates the distance traveled in the present zone, and the number IR of the next zone to be entered. Essentially, GG is called for each body adjacent to the present zone, calculating RIN and RØUT, the distances to entry and exit of the body along the trajectory. The next zone to be entered is determined by again calling GG to calculate RIN and RØUT for each body adjacent to the next possible zone. These next possible zones are determined by examining a list of all the previous zones entered on crossing this body. RIN and RØUT are checked against the input zone descriptions to determine the correct zone. If it is not found in the list of previous zones, all other zones are examined in a similar fashion, and when the correct zone is found, it is added to the list of previous zones for that body, if the storage allocated is not yet exhausted. If no more storage is available, future calls will search the list as it was adding no more to it but examining all other zones if it doesn't find the correct zone in the existing list. If the new zone is different from the old, G1 returns; otherwise G1 continues tracking until a different zone is encountered. One change added to the MØRSE version of G1 is that if the distance to the next boundary is greater than the distance to scattering, G1 returns without determining the next zone past the boundary, setting the flag MARKG in common ØRGI. Called from: GØMST, EUCLID, MESH.

Subroutines called: GG, PR.

Commons required: PAREM, GØMLØC, DBG, ØRGI, TAPE

Variables required:

XB(3)	- starting coordinates of present trajectory.
WB(3)	- direction cosines of trajectory.
IR	- present zone.
DIST	- present distance from XB(3).
DISTO	- distance from XB(3) to next scattering point.
NASC	- less than zero if this is a new trajectory.
KLØØP	- trajectory index.
PINF	- machine infinity.

MA, FPD, LØCREG,
 NUMBØD, IRØR, - locations in blank common used for variable
 IR1, IR2 dimensioning.

Variables changed:

KLØØP - trajectory index incremented if this is a new trajectory.

NASC - next body intersected by trajectory.

LSURF - surface of body NASC crossed at next intersection (negative if leaving and positive if entering NASC).

DIST - distance from XB(3) to next intersection or collision site.

MARK - set to 1 if distance to collision (DIST0) is less than distance to next intersection (otherwise 0).

S - distance traveled on this call to G1.

IRPRIM - zone to be entered on boundary crossing.

MA(INEXT) - new zone added to the list of next possible zones for body NBØ.

MA(INEX) - location in MA of next item in the next possible zone list for body NBØ (these lists leap-frog through the end of the MA array).

Significant internal variables:

NBØ - absolute value is body being considered while a negative or positive sign indicates that zone IR or IRP is outside or inside the body respectively.

RØUT - distance to exit of body NBØ calculated by GG.

RIN - distance to entry of body NBØ calculated by GG.

LRI - surface of body NBØ entered by trajectory.

LRØ - surface of body NBØ trajectory exits.

IRP - zone being considered as next zone.

Errors:

If message is "exit being called . . . next region not found" - NASC will be next body the ray will intersect (NBØ, N, NUM, ITYPE, SMIN, IRP, LØCAT all apply to maximum body number and have no significance).

Subroutine JØMIN (NADD, I1, IO)

The input of the geometry data is controlled by the JØMIN subroutine, which performs the following tasks:

- Reads all geometry input data except the region volumes.
- Writes the body and zone data on a mass storage unit (IØUT=16).
- Determines the length of all geometry arrays.
- Calculates the beginning location in blank common of geometry arrays.
- Initializes geometry arrays.
- Calls the GTVLIN subroutine which returns region volumes.

Since combinatorial geometry input data is dynamically allocated to conserve storage area, it is stored temporarily on a mass storage device. This allows the core storage requirements to be determined. Hence, much of the coding in JØMIN is similar to GENI, which reads the data on the mass storage device and puts it into blank common.

Called from: INPUT1

Subroutines called: GENI, GTVLIN

Commons required: Blank, GØMLØC, PAREM, TAPE

Variables required:

All variables in GØMLØC, I1, IO.

Variables changed:

All variables in GØMLØC, IVØPT, IDBG, MRIZ(I), MMIZ(I).

Important internal variables:

IØUT - mass storage unit.

NAZT - total number of adjacent zones summed over all zones.

Subroutine LØØKZ (X, Y, Z, MA, FPD, LØCREG, NUMBØD, IRØR, NSØR)

The purpose of this routine is to return the combinatorial geometry zone of point (X, Y, Z) so that tracking can be initialized. The coding has been borrowed from the second half of subroutine G1 and adapted to determine the zone of a source particle. For efficiency LØØKZ builds a list of possible source zones to search on future calls. If the region is not found on this list, all other zones are examined and upon determining the new source zone, it too is added to the list. Notice that the starting direction cosines (.8, .6, 0.0) are assumed in LØØKZ, but may be changed elsewhere.

Routines called: GG

Commons required: ØRGI, PAREM, GØMLØC, and DBG.

Variables required:

X, Y, Z - coordinates for which a zone is desired.
 MA, FPD, LØCREG, - locations in blank common used for variable
 NUMBØD, IRØR, NSØR dimensioning.

Variables changed:

KLØØP - trajectory index is incremented.
 NMED - common ØRGI variable set to correct zone number.
 NSØR(INEXT)- new zone added to list of possible source zones.
 XB - initialized to x, y, z from NUTRON common.
 IRPRIM - input zone number.
 IR - code zone number.
 DIST - initialized to 0.
 NBLZ - packed word containing IRPRIM and IR.
 NMEDG - input zone number in ORGI common.

Significant internal variables:

WB(3) - set to .8, .6, 0.0 so that LØØKZ need not be called
 with a direction.

Table 4.24 (Cont'd.)

Variable	Definition
LRØ	Exit surface calculated in GG.
KLØØP	Trajectory index of present path incremented in G1.
LØØP	Index of last trajectory calculated for body NBØ. If LØØP is equal to KLØØP, GG returns immediately with old values in RIN, RØUT, LRI, and LRØ.
ITYPE	Body type of body NBØ (indicates BØX, SPH, etc.).
NØA	Not used.

Table 4.25. Definitions of Variables in Common ØRGI*

Variable	Definition
DISTO	Distance from point XB(3) to next scattering point. Used in G1 to avoid calculating the next zone if a scattering event occurs before the intersection.
MARK	Set 1 in G1 if trajectory end point is reached before next intersection. Otherwise set to 0.
NMEDG	Zone number IR from a LØØKZ call.
NBLZ	Packed word containing both input zone and code zone numbers. Stored in BLZNT by MSOUR.
BLZOLD	Packed word containing code and input zone numbers for previous collision.

* Note: Variable names are not the same in all routines. Also, on non-IBM-360 machines, the order of the variables is MARK, DISTO, NMEDG. Reordering resulted from the conversion of the IBM-360 version to double precision.

Table 4.26 (Cont'd.)

Subroutine Printing Message	Message	Meaning
	'** ERROR IN NNE, NE OR THE GROUP NUMBERS -- PRINT IS OF MODIFIED VALUES'	Either IB(NNE) does not equal NGPQTN or IB(NE) does not equal the maximum number of groups. The routine modifies the user's data to correct this. User should change his IB array in his input.
JØMIN	'ITYPE' = ____ DOES NOT EQUAL ANY OF THE FOLLOWING ____ '	The body type given does not exist in the code.
GENI	'ITYPE = ____ DOES NOT EQUAL ANY OF THE FOLLOWING ____ '	The body type given does not exist in the code.
ALBERT	'ERROR IN SIDE DE- SCRIPTION ____ '	There are less than 3 points that describe this side - i.e., the user has defined a point or a line rather than a surface.
	'ERROR IN FACE DE- SCRIPTION ____ '	The points describing this face are not coplanar.
GTVLIN	'***** ERROR IN VOLUME CALCULATION ***** JF = ____ NIR = ____ '	The number of regions for which volumes were calculated does not equal the number of regions in the geometry.

Table 4.27. Diagnostic Messages From Other Modules of MØRSE

Subroutine Printing Message	Message	Meaning
FBANK	'WARNING***NO ROOM IN BANK FOR SECONDARIES***'	Maximum number of particles have been generated and the bank is full. No more fission particles will be generated.
GSTØRE	'WARNING***NO ROOM IN BANK FOR SECONDARIES***'	Maximum number of particles have been generated and the bank is full. No more gamma rays will be generated until bank decreases.
FSØUR	'NO FISSIONS GENERATED IN LAST BATCH, ____ BATCHES COMPLETED'	Problem continues for a fixed source problem. For a criticality problem the neutron population has died away.
MØRSE	'NREG = ____ MXREG = ____ MXREG ON CARD I MUST BE GE TO THE NUMBER OF REGIONS DESCRIBED IN GEOMETRY INPUT'	Self explanatory - change your input data.
CØLISN	'IN GRP = ____, OUT GRP = ____ J = ____ PROB = ____ RAND = ____'	All the scattering angle probabilities are 0 or negative for this group. An index is wrong or the data on the angle of scattering has been destroyed.
PTHETA	'ERROR in PTHETA ____ ISTAT = ____ NCOEF = ____'	The user has called PTHETA without saving Legendre coefficients - ISTAT must be non zero.

Table 4.27 (Cont'd.)

Subroutine Printing Message	Message	Meaning
GOMST	'NAME = ____ NMED = ____ NREG = ____ X = ____ Y = ____ Z = ____'	An error has been encountered in the geometry tracking for this particle. There should be messages before and after this giving more information. G1 will print a message and PR will be called here. Particle will be treated as an escape and calculation will continue.
EUCLID	'IRPRIM = ____ IN EUCLID'	An error has occurred in calculating the number of mean free paths to the detector. There will be an error message preceding this which will come from G1. Program allows <u>5</u> such errors before termination. Particle will be treated as an escape and will make no contribution to estimate.
LOOKZ	'IR = ____ XB = ____ ____ WB = ____ ____, DIST = ____ ____ 'MA ARRAY' Entire MA array printed 'FPD ARRAY' Entire FPD array printed '**** EXIT BEING CALLED FROM LOOKZ'	Zone has not been found. Check your zone specifications.

Table 4.27 (Cont'd.)

Subroutine Printing Message	Message	Meaning
NØRML	'INVALID REGION OR BODY IN NORMAL. IR = ____ NASC = ____'	NASC isn't in region IR Check to see if IR or NASC has been overwritten.
GG	'IN GG ITYPE = ____ IR = ____ NBO = ____'	ITYPE is not one of the body types (1 - 9) allowed. User has usually overstored on MA array. Occurs if body number out of range is used in zone description (usu- ally a mispunched item).
G1	'NO VALID DISTANCE IN G1, ____' '***** GEOMETRY SEARCH ARRAY FULL *****' 'IR = ____ XB = ____ WB = ____ DIST = ____' 'MA ARRAY' Entire MA array printed 'FPD ARRAY' Entire FPD array printed	G1 could not determine the next body that the ray will intersect. There is a probable error in user's geometry specifications, or he may have written over his geometry data. In order to save some computer time, user may want to increase the values of NAZ on his zone specification cards. Only harm done is increase in computer time which is often insignificant. G1 could not find the next region that the particle would enter. It checked all regions and is saying that the particle won't enter any of them.

The total derivative of the angular flux with respect to R is given by

$$\frac{d}{dR} \phi(\bar{r}', E, \bar{\Omega}, t') = \frac{\partial x}{\partial R} \frac{\partial \phi}{\partial x} + \frac{\partial y}{\partial R} \frac{\partial \phi}{\partial y} + \frac{\partial z}{\partial R} \frac{\partial \phi}{\partial z} + \frac{\partial t}{\partial R} \frac{\partial \phi}{\partial t}$$

which, according to Fig. A.1 and noting that the particle's speed (v) is equal to $(-dR/dt)$ can be rewritten as

$$\begin{aligned} \frac{d}{dR} \phi(\bar{r}', E, \bar{\Omega}, t') &= -\Omega_x \frac{\partial \phi}{\partial x} - \Omega_y \frac{\partial \phi}{\partial y} - \Omega_z \frac{\partial \phi}{\partial z} - \frac{1}{v} \frac{\partial \phi}{\partial t} \\ &= -\bar{\Omega} \cdot \nabla \phi(\bar{r}', E, \bar{\Omega}, t') - \frac{1}{v} \frac{\partial \phi}{\partial t} . \end{aligned} \quad (10)$$

Equation (10) can be expressed in group notation as

$$-\frac{d}{dR} \phi_g(\bar{r}', \bar{\Omega}, t') = \frac{1}{v_g} \frac{\partial}{\partial t} \phi_g(\bar{r}', \bar{\Omega}, t') + \bar{\Omega} \cdot \nabla \phi_g(\bar{r}', \bar{\Omega}, t') . \quad (11)$$

Substitution of Eq. (11) into Eq. (9) with $\bar{r} \equiv \bar{r}'$ and $t \equiv t'$ yields

$$\begin{aligned} -\frac{d}{dR} \phi_g(\bar{r}', \bar{\Omega}, t') + \Sigma_t^g(\bar{r}') \phi_g(\bar{r}', \bar{\Omega}, t') &= S_g(\bar{r}', \bar{\Omega}, t') \\ &+ \sum_{g'=g}^1 \int_{4\pi} d\bar{\Omega}' \Sigma_s^{g' \rightarrow g}(\bar{r}', \bar{\Omega}' \rightarrow \bar{\Omega}) \phi_{g'}(\bar{r}', \bar{\Omega}', t') . \end{aligned} \quad (12)$$

The integrating factor

$$e^{-\int_0^R \Sigma_t^g(\bar{r} - R'\bar{\Omega}) dR'}$$

is introduced in the following manner:

$$\begin{aligned} \frac{d}{dR} \left[\phi_g(\bar{r}', \bar{\Omega}, t') e^{-\int_0^R \Sigma_t^g(\bar{r} - R'\bar{\Omega}) dR'} \right] &= -e^{-\int_0^R \Sigma_t^g(\bar{r} - R'\bar{\Omega}) dR'} \\ &\times \left[-\frac{d\phi_g}{dR} + \Sigma_t^g(\bar{r}') \phi_g(\bar{r}', \bar{\Omega}, t') \right] . \end{aligned} \quad (13)$$

Using Eq. (13), Eq. (12) can be rewritten as

$$\begin{aligned}
 & - \frac{d}{dR} \left\{ \phi_g(\bar{r}', \bar{\Omega}, t') e^{-\int_0^R \Sigma_t^g(\bar{r} - R'\bar{\Omega}) dR'} - \int_0^R \Sigma_t^g(\bar{r} - R'\bar{\Omega}) dR' \right. \\
 & \left. \times \left[S_g(\bar{r}', \bar{\Omega}, t') + \sum_{g'=g}^1 \int_{4\pi} d\bar{\Omega}' \Sigma_s^{g' \rightarrow g}(\bar{r}', \bar{\Omega}' \rightarrow \bar{\Omega}) \phi_{g'}(\bar{r}', \bar{\Omega}', t') \right] \right\} = 0
 \end{aligned} \quad (14)$$

Multiply Eq. (14) by dR and integrate ($R = 0$ to $R = \infty$); then

$$\begin{aligned}
 & \phi_g(\bar{r}, \bar{\Omega}, t) - \phi_g(\infty, \bar{\Omega}, t_\infty) e^{-\int_0^\infty \Sigma_t^g(\bar{r} - R'\bar{\Omega}) dR'} \\
 & = \int_0^\infty dR e^{-\int_0^R \Sigma_t^g(\bar{r} - R'\bar{\Omega}) dR'} \left\{ S_g(\bar{r} - R\bar{\Omega}, \bar{\Omega}, t - \frac{R}{v}) \right. \\
 & \left. + \sum_{g'=g}^1 \int_{4\pi} d\bar{\Omega}' \Sigma_s^{g' \rightarrow g}(\bar{r} - R\bar{\Omega}, \bar{\Omega}' \rightarrow \bar{\Omega}) \phi_{g'}(\bar{r}', \bar{\Omega}', t') \right\}
 \end{aligned} \quad (15)$$

Require that

$$\left\{ \phi_g(\infty, \bar{\Omega}, t_\infty) e^{-\int_0^\infty \Sigma_t^g(\bar{r} - R'\bar{\Omega}) dR'} \right\} = 0, \quad (16)$$

and introduce the "optical thickness"

$$\beta_g(\bar{r}, R, \bar{\Omega}) \equiv \int_0^R \Sigma_t^g(\bar{r} - R'\bar{\Omega}) dR', \quad (17)$$

and Eq. (15) becomes

$$\begin{aligned}
 \phi_g(\bar{r}, \bar{\Omega}, t) &= \int_0^\infty dR e^{-\beta_g(\bar{r}, R, \bar{\Omega})} \left\{ S_g(\bar{r} - R\bar{\Omega}, \bar{\Omega}, t - R/v) \right. \\
 & \left. + \sum_{g'=g}^1 \int_{4\pi} d\bar{\Omega}' \Sigma_s^{g' \rightarrow g}(\bar{r} - R\bar{\Omega}, \bar{\Omega}' \rightarrow \bar{\Omega}) \phi_{g'}(\bar{r}', \bar{\Omega}', t') \right\}.
 \end{aligned} \quad (18)$$

Equation (18) will be referred to as the "Integral Flux Density Equation."

An effect of interest λ in group notation can be expressed as

$$\lambda_g = \iiint P_g^\phi(\bar{r}, \bar{\Omega}, t) \phi_g(\bar{r}, \bar{\Omega}, t) d\bar{r} d\bar{\Omega} dt, \quad (19)$$

where

$P_g^\phi(\bar{r}, \bar{\Omega}, t)$ = the response function of the effect of interest due to a unit angular group flux (group g , $\bar{r}$, $\bar{\Omega}$, time t),

$$= \frac{\int_{\Delta E_g} P^\phi(\bar{r}, E, \bar{\Omega}, t) \phi(\bar{r}, E, \bar{\Omega}, t) dE}{\int_{\Delta E_g} \phi(\bar{r}, E, \bar{\Omega}, t) dE}$$

λ_g = that portion of the effect of interest associated with the g th energy group.

The λ_g are so defined that the total effect of interest λ is given by the summation

$$\lambda = \sum_{g=1}^G \lambda_g. \quad (20)$$

Integral Event Density Equation

The "event density" $\psi_g(\bar{r}, \bar{\Omega}, t)$ describes the density of particles going into a collision and is related to the group angular flux in the following manner:

$$\psi_g(\bar{r}, \bar{\Omega}, t) \equiv \Sigma_t^g(\bar{r}) \phi_g(\bar{r}, \bar{\Omega}, t). \quad (21)$$

where

$\psi_g(\bar{r}, \bar{\Omega}, t) d\bar{\Omega}$ = the number of collision events per unit volume and time at the space point $\bar{r}$ and time t experienced by particles having energies within the g th energy group and directions in $d\bar{\Omega}$ about $\bar{\Omega}$.

The defining equation for the event density is obtained by multiplying both sides of Eq. (18) by the group total cross section $\Sigma_t^g(\bar{r})$ and identifying the product $\Sigma_t^g(\bar{r}) \phi_g(\bar{r}, \bar{\Omega}, t)$ as the event density $\psi_g(\bar{r}, \bar{\Omega}, t)$:

$$\psi_g(\bar{r}, \bar{\Omega}, t) = \int_0^\infty dR \Sigma_t^g(\bar{r}) e^{-\beta_g(\bar{r}, R, \bar{\Omega})} \left(S_g(\bar{r} - R\bar{\Omega}, \bar{\Omega}, t - R/v) \right. \\ \left. + \sum_{g'=g} \int_{4\pi} d\bar{\Omega}' \frac{\Sigma_s^{g' \rightarrow g}(\bar{r} - R\bar{\Omega}, \bar{\Omega}' \rightarrow \bar{\Omega})}{\Sigma_t^{g'}(\bar{r}')} \psi_{g'}(\bar{r}', \bar{\Omega}', t') \right). \quad (22)$$

Equation (22) will be referred to as the "Integral Event Density Equation."

The effect of interest λ_g can be expressed in terms of the event density; consider Eq. (19) rewritten as

$$\lambda_g = \iiint \frac{P_g^\phi(\bar{r}, \bar{\Omega}, t)}{\Sigma_t^g(\bar{r})} \Sigma_t^g(\bar{r}) \phi_g(\bar{r}, \bar{\Omega}, t) d\bar{r} d\bar{\Omega} dt \\ = \iiint P_g^\psi(\bar{r}, \bar{\Omega}, t) \psi_g(\bar{r}, \bar{\Omega}, t) d\bar{r} d\bar{\Omega} dt, \quad (23)$$

where

$$P_g^\psi(\bar{r}, \bar{\Omega}, t) = \text{the response function of the effect of interest due to a} \\ \text{particle which experiences an event at (group } g, \bar{r}, \bar{\Omega}, \text{ time } t), \\ P_g^\psi(\bar{r}, \bar{\Omega}, t) = P_g^\phi(\bar{r}, \bar{\Omega}, t) / \Sigma_t^g(\bar{r}) \\ \text{or} \\ P_g^\phi(\bar{r}, \bar{\Omega}, t) = \Sigma_t^g(\bar{r}) P_g^\psi(\bar{r}, \bar{\Omega}, t). \quad (24)$$

Integral Emergent Particle Density Equation

Define the emergent particle density $\chi_g(\bar{r}, \bar{\Omega}, t)$ as the density of particles leaving a source or emerging from a real collision with phase space coordinates (group g , $\bar{r}$, $\bar{\Omega}$, t),

$$\chi_g(\bar{r}, \bar{\Omega}, t) = S_g(\bar{r}, \bar{\Omega}, t) + \sum_{g'} \int_{4\pi} d\bar{\Omega}' \Sigma_s^{g' \rightarrow g}(\bar{r}, \bar{\Omega}' \rightarrow \bar{\Omega}) \phi_{g'}(\bar{r}, \bar{\Omega}', t). \quad (25)$$

Then Eq. (18) can be written as

$$\phi_g(\bar{r}, \bar{\Omega}, t) = \int_0^\infty dR e^{-\beta_g(\bar{r}, R, \bar{\Omega})} \chi_g(\bar{r}', \bar{\Omega}, t') . \quad (26)$$

The "Integral Emergent Particle Density Equation" is obtained by substituting Eq. (26) into Eq. (25):

$$\begin{aligned} \chi_g(\bar{r}, \bar{\Omega}, t) &= S_g(\bar{r}, \bar{\Omega}, t) + \sum_{g'=g}^1 \int_{4\pi} d\bar{\Omega}' \Sigma_s^{g' \rightarrow g}(\bar{r}, \bar{\Omega}' \rightarrow \bar{\Omega}) \int_0^\infty dR e^{-\beta_{g'}(\bar{r}, R, \bar{\Omega}')} \chi_{g'}(\bar{r}', \bar{\Omega}', t') \\ &= S_g(\bar{r}, \bar{\Omega}, t) + \sum_{g'=g}^1 \int_{4\pi} d\bar{\Omega}' \frac{\Sigma_s^{g' \rightarrow g}(\bar{r}, \bar{\Omega}' \rightarrow \bar{\Omega})}{\Sigma_t^{g'}(\bar{r})} \int_0^\infty dR \Sigma_t^{g'}(\bar{r}) e^{-\beta_{g'}(\bar{r}, R, \bar{\Omega}')} \chi_{g'}(\bar{r}', \bar{\Omega}', t') . \end{aligned} \quad (27)$$

The effect of interest λ_g can also be expressed in terms of the emergent particle density

$$\lambda_g = \iiint P_g^X(\bar{r}, \bar{\Omega}, t) \chi_g(\bar{r}, \bar{\Omega}, t) d\bar{r} d\bar{\Omega} dt . \quad (28)$$

The response function $P_g^X(\bar{r}, \bar{\Omega}, t)$ is obtained by considering a particle which emerges from a collision at $\bar{r}$ with phase space coordinates (group g , $\bar{\Omega}$, time t). This particle will experience an event in dR about $\bar{r}' = \bar{r} + R\bar{\Omega}$ at time $t' = t + R/v$ with the probability

$$\frac{1}{\Sigma_t^g(\bar{r}')} e^{-\int_0^R \Sigma_t^g(\bar{r} + R'\bar{\Omega}) dR'} dR ,$$

and the contribution of this event is the response function $P_g^\psi(\bar{r}', \bar{\Omega}, t')$. The sum of all such contributions to the effect of interest is given by

$$\int_0^\infty dR \Sigma_t^g(\bar{r}') e^{-\int_0^R \Sigma_t^g(\bar{r} + R'\bar{\Omega}) dR'} P_g^\psi(\bar{r}', \bar{\Omega}, t') ,$$

and should be the same as a response function $P_g^X(\bar{r}, \bar{\Omega}, t)$ which is based on emergent particle density. This leads to the following relationship:

$$P_g^X(\bar{r}, \bar{\Omega}, t) = \int_0^\infty dR \Sigma_t^g(\bar{r}') e^{-\beta_g^*(\bar{r}, R, \bar{\Omega})} P_g^\psi(\bar{r}', \bar{\Omega}, t') , \quad (29)$$

where

$P_g^X(\bar{r}, \bar{\Omega}, t)$ $\equiv$ the response function (of the effect of interest due to a particle which emerges from a collision having the phase space coordinates (group g , $\bar{r}$, $\bar{\Omega}$, time t))

$$\beta_g^*(\bar{r}, R, \bar{\Omega}) \equiv \int_0^R \Sigma_t^g(\bar{r} + R'\bar{\Omega}) dR' . \quad (30)$$

It is noted that $\beta_g^*(\bar{r}, R, \bar{\Omega})$ differs from the optical thickness $\beta_g(\bar{r}, R, \bar{\Omega})$ as defined by Eq. (17) in that the integration is performed in the positive $\bar{\Omega}$ direction and as such $\beta_g^*(\bar{r}, R, \bar{\Omega})$ is the adjoint of $\beta_g(\bar{r}, R, \bar{\Omega})$. $P_g^X(\bar{r}, \bar{\Omega}, t)$ can also be expressed in terms of $P_g^\phi(\bar{r}, \bar{\Omega}, t)$ by substituting Eq. (24) into Eq. (29), yielding

$$P_g^X(\bar{r}, \bar{\Omega}, t) = \int_0^\infty dR e^{-\beta_g^*(\bar{r}, R, \bar{\Omega})} P_g^\phi(\bar{r}', \bar{\Omega}, t') . \quad (31)$$

Operator Notation and Summary of the Forward Equations

Define the transport integral operator

$$T_g(\bar{r}' \rightarrow \bar{r}, \bar{\Omega}) \equiv \int_0^\infty dR \Sigma_t^g(\bar{r}) e^{-\beta_g(\bar{r}, R, \bar{\Omega})} , \quad (32)$$

and the collision integral operator

$$C_{g' \rightarrow g}(\bar{r}, \bar{\Omega}' \rightarrow \bar{\Omega}) = \frac{1}{\sum_{g'=g}} \int_{4\pi} d\bar{\Omega}' \frac{\Sigma_s^{g' \rightarrow g}(\bar{r}, \bar{\Omega}' \rightarrow \bar{\Omega})}{\Sigma_t^{g'}(\bar{r})} , \quad (33)$$

which can be rewritten as

$$C_{g' \rightarrow g}(\bar{r}, \bar{\Omega}' \rightarrow \bar{\Omega}) = \frac{1}{\sum_{g'=g}} \int_{4\pi} d\bar{\Omega}' \left[\frac{\Sigma_s^{g' \rightarrow g}(\bar{r}, \bar{\Omega}' \rightarrow \bar{\Omega})}{\Sigma_t^{g'}(\bar{r})} \right] \left[\frac{\Sigma_s^{g'}(\bar{r})}{\Sigma_t^{g'}(\bar{r})} \right] , \quad (34)$$

where

$$\Sigma_s^{g'}(\bar{r}) = \sum_g \int_{4\pi} d\bar{\Omega} \Sigma_s^{g' \rightarrow g}(\bar{r}, \bar{\Omega}' \rightarrow \bar{\Omega}) . \quad (35)$$

In Eq. (34), $[\Sigma_s^{g' \rightarrow g}(\bar{r}, \bar{\Omega}' \rightarrow \bar{\Omega}) / \Sigma_s^{g'}(\bar{r})]$ is a normalized probability density function from which the selection of a new energy group and direction can be accomplished and $[\Sigma_s^{g'}(\bar{r}) / \Sigma_t^{g'}(\bar{r})]$ is the nonabsorption probability.

Using the transport and collision integral operators, Eq. (22) can be rewritten as

$$\psi_g(\bar{r}, \bar{\Omega}, t) = T_g(\bar{r}' \rightarrow \bar{r}, \bar{\Omega}) (S_g(\bar{r}', \bar{\Omega}, t') + C_{g' \rightarrow g}(\bar{r}', \bar{\Omega}' \rightarrow \bar{\Omega}) \psi_g(\bar{r}', \bar{\Omega}', t')). \quad (36)$$

The term $T_g(\bar{r}', \bar{r}, \bar{\Omega}) S_g(\bar{r}', \bar{\Omega}, t')$ can be identified as the "first collision source" and denoted by

$$S_c^g(\bar{r}, \bar{\Omega}, t) \equiv T_g(\bar{r}' \rightarrow \bar{r}, \bar{\Omega}) S_g(\bar{r}', \bar{\Omega}, t'), \quad (37)$$

and the "Integral Event Density Equation" becomes

$$\psi_g(\bar{r}, \bar{\Omega}, t) = S_c^g(\bar{r}, \bar{\Omega}, t) + T_g(\bar{r}' \rightarrow \bar{r}, \bar{\Omega}) C_{g' \rightarrow g}(\bar{r}', \bar{\Omega}' \rightarrow \bar{\Omega}) \psi_g(\bar{r}', \bar{\Omega}', t'). \quad (38)$$

Using the relationship $\psi_g(\bar{r}, \bar{\Omega}, t) = \Sigma_t^{g'}(\bar{r}) \phi_g(\bar{r}, \bar{\Omega}, t)$, Eq. (38) can be transformed into the "Integral Flux Density Equation:"

$$\phi_g(\bar{r}, \bar{\Omega}, t) = \frac{S_c^g(\bar{r}, \bar{\Omega}, t)}{\Sigma_t^g(\bar{r})} + T_g(\bar{r}' \rightarrow \bar{r}, \bar{\Omega}) C_{g' \rightarrow g}(\bar{r}', \bar{\Omega}' \rightarrow \bar{\Omega}) \frac{\Sigma_t^{g'}(\bar{r}')}{\Sigma_t^g(\bar{r})} \phi_{g'}(\bar{r}', \bar{\Omega}', t'). \quad (39)$$

Finally, the integral operators are introduced into Eq. (27) and the following form for the "Integral Emergent Particle Density Equation" is obtained:

$$\chi_g(\bar{r}, \bar{\Omega}, t) = S_g(\bar{r}, \bar{\Omega}, t) + C_{g' \rightarrow g}(\bar{r}, \bar{\Omega}' \rightarrow \bar{\Omega}) T_{g'}(\bar{r}' \rightarrow \bar{r}, \bar{\Omega}') \chi_{g'}(\bar{r}', \bar{\Omega}', t'). \quad (40)$$

An examination of Equations (38), (39), and (40) would reveal that either the "Integral Event Density Equation" or the "Integral Emergent Particle Density Equation" would provide a reasonable basis for a Monte Carlo random walk. Equation (40) was selected for the MORSE code since the source particles would be introduced according to the natural distribution rather than the distribution of first collisions. However, it is noted that after the introduction of the source particle, the subsequent

random walk can be regarded in terms of either Eq. (38) or Eq. (40) with the particle's weight at a collision site being the weight before collision (WTBC) or the weight after collision (WATE), respectively.

The random walk based on the "Integral Emergent Particle Density Equation" would introduce a particle into the system according to the source function. The particle travels to the site of its first collision as determined by the transport kernel. Its weight is modified by the non-absorption probability and a new energy group and flight direction are selected from the collision kernel. The transport and collision kernels are applied successively determining the particle's emergent phase space coordinates corresponding to the second, third, etc., collision sites until the random walk is terminated due to the reduction of the particle's weight below some cut-off value or because the particle escapes from that portion of phase space associated with a particular problem (for example, escape from the system, slowing down below an energy cut-off, or exceeding some arbitrarily specified age cut-off).

Random Walk Procedure

The actual implementation of the random walk procedure is accomplished by approximating the integrals implied in the collision and transport integral operators by the sum

$$\chi_g(\bar{r}, \bar{\Omega}, t) = \sum_{n=0}^{\infty} \chi_g^n(\bar{r}, \bar{\Omega}, t), \quad (41)$$

where

$\chi_g^n(\bar{r}, \bar{\Omega}, t) d\bar{\Omega}$ = the emergent particle density of particles emerging from its n th collision and having phase space coordinates (group g , $\bar{r}$, $d\bar{\Omega}$ about $\bar{\Omega}$, time t),

$$\chi_g^0(\bar{r}, \bar{\Omega}, t) = S_g(\bar{r}, \bar{\Omega}, t),$$

$$\chi_g^n(\bar{r}, \bar{\Omega}, t) = C_{g' \rightarrow g}(\bar{r}, \bar{\Omega}' \rightarrow \bar{\Omega}) T_{g'}(\bar{r}' \rightarrow \bar{r}, \bar{\Omega}') \chi_{g'}^{n-1}(\bar{r}', \bar{\Omega}', t').$$

Thus, the source coordinates (group g_0 , $\bar{r}_0$, $\bar{\Omega}_0$, time t_0) are selected from $S_g(\bar{r}, \bar{\Omega}, t)$ and a flight distance R is picked $\Sigma_t^{g_0}(\bar{r}) e^{-\beta_{g_0}(\bar{r}, R, \bar{\Omega}_0)}$ to determine the site for the first collision $\bar{r}_1$ and the particle's age $t_1 = t_0 + R/v_{g_0}$. The probability of scattering is $\Sigma_s^{g_0}(\bar{r}_1)/\Sigma_t^{g_0}(\bar{r}_1)$. All particles are forced to scatter and their weight is modified with this probability. A new group g_1 is selected according to the distribution

The $\hat{\Sigma}_t^g(\bar{r})$, $\hat{\Sigma}_s^{g \rightarrow g'}(\bar{r}, \bar{\Omega} \rightarrow \bar{\Omega}')$, and $\hat{v}_g$ are adjoint weighted group parameters and their use in the solution of Eq. (51) provides group adjoint fluxes defined by Eq. (52) where $\phi^*(\bar{r}, E, \bar{\Omega}, t)$ represents the solution of Eq. (50).

Another approach for defining group adjoint fluxes is to directly devise the equation which is adjoint to the group form of the Boltzmann equation [Eq. (9)]. The group adjoint equation so obtained* is given by

$$\begin{aligned}
 & -\frac{1}{v_g} \frac{\partial}{\partial t} \phi_g^*(\bar{r}, \bar{\Omega}, t) - \nabla \cdot \bar{\Omega} \phi_g^*(\bar{r}, \bar{\Omega}, t) + \Sigma_t^g(\bar{r}) \phi_g^*(\bar{r}, \bar{\Omega}, t) \\
 & = S_g^*(\bar{r}, \bar{\Omega}, t) + \sum_{g'=1}^G \int d\bar{\Omega}' \Sigma_s^{g \rightarrow g'}(\bar{r}, \bar{\Omega} \rightarrow \bar{\Omega}') \phi_{g'}^*(\bar{r}, \bar{\Omega}', t),
 \end{aligned} \tag{57}$$

$$g = 1, 2, \dots, G.$$

where v_g , $\Sigma_t^g(\bar{r})$ are forward weighted group parameters identical to those which occur in Eq. (9) and the matrix $\Sigma_s^{g \rightarrow g'}(\bar{r}, \bar{\Omega} \rightarrow \bar{\Omega}')$ is simply the transposition of the forward weighted group-to-group differential scattering cross-section matrix.

The group adjoint fluxes $\phi_g^*(\bar{r}, \bar{\Omega}, t)$ which represent the solution of Eq. (57) are adjoint to the group fluxes ϕ_g and do not necessarily assume the same values as the group adjoint fluxes $\hat{\phi}_g^*(\bar{r}, \bar{\Omega}, t)$, i.e.,

$$\phi_g^*(\bar{r}, \bar{\Omega}, t) \neq \frac{1}{\Delta E_g} \int_{\Delta E_g} \phi^*(\bar{r}, E, \bar{\Omega}, t) dE.$$

This follows since $\Sigma_t^g(\bar{r})$, $\Sigma_s^{g \rightarrow g'}(\bar{r}, \bar{\Omega} \rightarrow \bar{\Omega}')$, and v_g are, in general, different from the adjoint weighted values. Usually forward weighted group parameters, as implied by Eq. (57), are used in MORSE. However, other weighting schemes, such as adjoint or adjoint and forward, deserve consideration when cross-section weighting is a problem. When a sufficiently fine group structure is employed, the group parameters become less sensitive to the weighting scheme and the corresponding group adjoint fluxes are also nearly the same.

*The derivation of Eq. (57) is not presented here because of its similarity with the previous derivation of Eq. (50); the integrals over energy are simply replaced by appropriate group summations.

Integral Point-Value Equation

Equation (57) is now transformed into an integral form following essentially the same procedures used with the forward equations. As shown below, let $\bar{r}' = \bar{r} + R\bar{\Omega}$ rather than $\bar{r}' = \bar{r} - R\bar{\Omega}$ as was the convention with the forward equations. The total derivative of $\phi_g^*(\bar{r}', \bar{\Omega}, t)$ with respect to R is given by

$$\frac{d}{dR} \phi_g^*(\bar{r}', \bar{\Omega}, t') = \frac{1}{v_g} \frac{\partial}{\partial t} \phi_g^*(\bar{r}', \bar{\Omega}, t') + \nabla \cdot \bar{\Omega} \phi_g^*(\bar{r}', \bar{\Omega}, t') . \quad (58)$$

$$- \int_0^R \Sigma_t^g(\bar{r} + R'\bar{\Omega}) dR'$$

Use of the integrating factor e provides the following relationship:

$$\begin{aligned} & \frac{d}{dR} \left\{ \phi_g^*(\bar{r}', \bar{\Omega}, t') e^{- \int_0^R \Sigma_t^g(\bar{r} + R'\bar{\Omega}) dR'} \right\} \\ &= \frac{d\phi_g^*}{dR}(\bar{r}', \bar{\Omega}, t') e^{- \int_0^R \Sigma_t^g(\bar{r} + R'\bar{\Omega}) dR'} - \Sigma_t^g(\bar{r}') \phi_g^*(\bar{r}', \bar{\Omega}, t') \\ & \quad \times e^{- \int_0^R \Sigma_t^g(\bar{r} + R'\bar{\Omega}) dR'} = e^{- \int_0^R \Sigma_t^g(\bar{r}' + R'\bar{\Omega}) dR'} \\ & \quad \times \left(\frac{d}{dR} \phi_g^*(\bar{r}', \bar{\Omega}, t') - \Sigma_t^g(\bar{r}') \phi_g^*(\bar{r}', \bar{\Omega}, t') \right). \end{aligned} \quad (59)$$

Equation (59), together with Eq. (58), can be arranged to give

$$\begin{aligned} & \left(-\frac{1}{v} \frac{\partial}{\partial t} \phi_g^*(\bar{r}', \bar{\Omega}, t') - \nabla \cdot \bar{\Omega} \phi_g^*(\bar{r}', \bar{\Omega}, t') + \Sigma_t^g(\bar{r}') \phi_g^*(\bar{r}', \bar{\Omega}, t') \right) \\ &= e^{- \int_0^R \Sigma_t^g(\bar{r} + R'\bar{\Omega}) dR'} \frac{d}{dR} \left(\phi_g^*(\bar{r}', \bar{\Omega}, t') e^{- \int_0^R \Sigma_t^g(\bar{r} + R'\bar{\Omega}) dR'} \right). \end{aligned} \quad (60)$$

It is noted that Eq. (60) is identically the left-hand side of Eq. (57) which can now be rewritten as

$$\begin{aligned} & - \frac{d}{dR} \left\{ \phi_g^*(\bar{r}', \bar{\Omega}, t') e^{- \int_0^R \Sigma_t^g(\bar{r} + R'\bar{\Omega}) dR'} \right\} = e^{- \int_0^R \Sigma_t^g(\bar{r} + R'\bar{\Omega}) dR'} \\ & \quad \times \left\{ S_g^*(\bar{r}', \bar{\Omega}, t') + \sum_{g'=g}^G \int d\bar{\Omega}' \Sigma_s^{g \rightarrow g'}(\bar{r}', \bar{\Omega} \rightarrow \bar{\Omega}') \phi_{g'}^*(\bar{r}', \bar{\Omega}', t') \right\}. \end{aligned} \quad (61)$$

Integrate Eq. (61) from $R = 0$ to $R = \infty$ and assume that

$$\left\{ \phi_g^*(\infty, \bar{\Omega}, t_\infty) e^{-\int_0^\infty \Sigma_t^g(\bar{r} + R'\bar{\Omega}) dR'} \right\} \equiv 0 ; \quad (62)$$

then the following integral expression for $\phi_g^*(\bar{r}, \bar{\Omega}, t)$ is obtained:

$$\begin{aligned} \phi_g^*(\bar{r}, \bar{\Omega}, t) = & \int_0^\infty dR e^{-\beta_g^*(\bar{r}, R, \bar{\Omega})} \{ S_g^*(\bar{r} + R\bar{\Omega}, \bar{\Omega}, t + R/v_g) \\ & + \sum_{g'} \int d\bar{\Omega}' \Sigma_s^{g \rightarrow g'}(\bar{r} + R\bar{\Omega}, \bar{\Omega} \rightarrow \bar{\Omega}') \phi_{g'}^*(\bar{r}', \bar{\Omega}', t') \} . \end{aligned} \quad (63)$$

Equation (63) contains the adjoint optical thickness $\beta_g^*(\bar{r}, R, \bar{\Omega})$ which was defined earlier by Eq. (30) as

$$\beta_g^*(\bar{r}, R, \bar{\Omega}) \equiv \int_0^R \Sigma_t^g(\bar{r} + R'\bar{\Omega}) dR' .$$

Redefine the source term as

$$S_{Tg}^*(\bar{r}, \bar{\Omega}, t) = \int_0^\infty dR e^{-\beta_g^*(\bar{r}, R, \bar{\Omega})} S_g^*(\bar{r} + R\bar{\Omega}, \bar{\Omega}, t + R/v_g) , \quad (64)$$

and Eq. (63) can be rewritten as

$$\begin{aligned} \phi_g^*(\bar{r}, \bar{\Omega}, t) = & S_{Tg}^*(\bar{r}, \bar{\Omega}, t) + \int_0^\infty dR e^{-\beta_g^*(\bar{r}, R, \bar{\Omega})} \sum_{g'} \int d\bar{\Omega}' \Sigma_s^{g \rightarrow g'}(\bar{r}', \bar{\Omega} \rightarrow \bar{\Omega}') \\ & \times \phi_{g'}^*(\bar{r}', \bar{\Omega}', t') = S_{Tg}^*(\bar{r}, \bar{\Omega}, t) + \int_0^\infty dR \Sigma_t^g(\bar{r} + R\bar{\Omega}) e^{-\beta_g^*(\bar{r}, R, \bar{\Omega})} \\ & \times \sum_{g'} \int d\bar{\Omega}' \frac{\Sigma_s^{g \rightarrow g'}(\bar{r}', \bar{\Omega} \rightarrow \bar{\Omega}')}{\Sigma_t^g(\bar{r}')} \phi_{g'}^*(\bar{r}', \bar{\Omega}', t') , \end{aligned} \quad (65)$$

Substitution of Eq. (71) into Eq. (64) yields

$$S_{Tg}^*(\bar{r}, \bar{\Omega}, t) = \int dR e^{-\beta_g^*(\bar{r}, R, \bar{\Omega})} P_g^\phi(\bar{r}', \bar{\Omega}, t'), \quad (72)$$

and according to Equations (24) and (29), Eq. (72) can be rewritten as Equations (73) and (74), respectively:

$$S_{Tg}^*(\bar{r}, \bar{\Omega}, t) = \int dR \Sigma_t^g(\bar{r}') e^{-\beta_g^*(\bar{r}, R, \bar{\Omega})} P_g^\psi(\bar{r}', \bar{\Omega}, t') \quad (73)$$

and

$$S_{Tg}^*(\bar{r}, \bar{\Omega}, t) = P_g^\chi(\bar{r}, \bar{\Omega}, t). \quad (74)$$

Substitution of Eq. (73) into Eq. (67) and Eq. (74) into Eq. (67) yields the following forms for the "Integral Point-Value Equation:"

$$\chi_g^*(\bar{r}, \bar{\Omega}, t) = T_g(\bar{r} \rightarrow \bar{r}', \bar{\Omega}) \{P_g^\psi(\bar{r}', \bar{\Omega}, t') + C_{g \rightarrow g'}(\bar{r}', \bar{\Omega} \rightarrow \bar{\Omega}') \chi_g^*(\bar{r}', \bar{\Omega}', t')\} \quad (75)$$

and

$$\chi_g^*(\bar{r}, \bar{\Omega}, t) = P_g^\chi(\bar{r}, \bar{\Omega}, t) + T_g(\bar{r} \rightarrow \bar{r}', \bar{\Omega}) C_{g \rightarrow g'}(\bar{r}', \bar{\Omega} \rightarrow \bar{\Omega}') \chi_g^*(\bar{r}', \bar{\Omega}', t'). \quad (76)$$

Integral Event-Value Equation

At this point let us introduce a value function based on the event density and to relate this quantity to the point-value function by considering a particle leaving a collision at $\bar{r}$ with phase space coordinates (group g , $\bar{\Omega}$, time t). The value of this particle to the effect of interest is the point-value function $\chi_g^*(\bar{r}, \bar{\Omega}, t)$. This particle will experience an event in dR about $\bar{r}' = \bar{r} + R\bar{\Omega}$ with the probability $[\Sigma_t^g(\bar{r}') e^{-\beta_g^*(\bar{r}, R, \bar{\Omega})} dR]$ and the value of this event (to the effect of interest) will be referred to as the "event-value" and be denoted by $W_g(\bar{r}', \bar{\Omega}, t')$. That is, the "event-value" $W_g(\bar{r}', \bar{\Omega}, t')$ is defined as the value (to the effect of interest) of having an event at $\bar{r}'$ with an incoming particle which has

phase space coordinates (group g , $\bar{\Omega}$, time t'). The sum of all such contributions to the effect of interest is given by

$$\int_0^{\infty} dR \Sigma_t^g(\bar{r}') e^{-\beta_g^*(\bar{r}, R, \bar{\Omega})} W_g(\bar{r}', \bar{\Omega}, t') ,$$

and, if the event-value function is properly defined, should equal the point-value function; that is,

$$\chi_g^*(\bar{r}, \bar{\Omega}, t) = \int_0^{\infty} dR \Sigma_t^g(\bar{r}') e^{-\beta_g^*(\bar{r}, R, \bar{\Omega})} W_g(\bar{r}', \bar{\Omega}, t') \quad (77)$$

or

$$\chi_g^*(\bar{r}, \bar{\Omega}, t) = T_g(\bar{r} \rightarrow \bar{r}', \bar{\Omega}) W_g(\bar{r}', \bar{\Omega}, t') . \quad (78)$$

A comparison of Eq. (78) with Eq. (75) would show that $W_g(\bar{r}, \bar{\Omega}, t)$ can be identified as

$$W_g(\bar{r}, \bar{\Omega}, t) = P_g^{\psi}(\bar{r}, \bar{\Omega}, t) + C_{g \rightarrow g'}(\bar{r}, \bar{\Omega} \rightarrow \bar{\Omega}') \chi_{g'}^*(\bar{r}, \bar{\Omega}', t) , \quad (79)$$

and substitution of Eq. (78) into Eq. (79) yields the defining equation for the "Event-Value Function"

$$W_g(\bar{r}, \bar{\Omega}, t) = P_g^{\psi}(\bar{r}, \bar{\Omega}, t) + C_{g \rightarrow g'}(\bar{r}, \bar{\Omega} \rightarrow \bar{\Omega}') T_{g'}(\bar{r} \rightarrow \bar{r}', \bar{\Omega}') W_{g'}(\bar{r}', \bar{\Omega}', t') . \quad (80)$$

Equation (80) will be referred to as the "Integral Event-Value Equation." A comparison of Eq. (80) with Eq. (38) would show that the event-value function $W_g(\bar{r}, \bar{\Omega}, t)$ is adjoint to the event density $\psi_g(\bar{r}, \bar{\Omega}, t)$. Therefore the effect of interest is given by

$$\lambda = \sum_g \iiint S_c^g(\bar{r}, \bar{\Omega}, t) W_g(\bar{r}, \bar{\Omega}, t) d\bar{r} d\bar{\Omega} dt . \quad (81)$$

Integral Emergent Adjuncton Density Equation

The solution of either the point-value equation, Eq. (76), or the event-value equation, Eq. (80), could be accomplished by Monte Carlo procedures; however, the random walk would not be the same as that implied by Eq. (40)*. Consider the following altered form of Eq. (76),

* The desire in MØRSE is to use the same random walk logic for both forward and adjoint calculations.

$$\chi_g^*(\bar{r}, \bar{\Omega}, t) = P_g \chi_g(\bar{r}, \bar{\Omega}, t) + \int dR \Sigma_t^g(\bar{r}) e^{-\beta_g^*(\bar{r}, R, \bar{\Omega})} \left(\frac{\Sigma_t^g(\bar{r}')}{\Sigma_t^g(\bar{r})} \right) \times \sum_{g'} \int d\bar{\Omega}' \frac{\Sigma_s^{g \rightarrow g'}(\bar{r}', \bar{\Omega} \rightarrow \bar{\Omega}')}{\sum_g \Sigma_s^{g \rightarrow g'}(\bar{r}', \bar{\Omega} \rightarrow \bar{\Omega}')} \left(\frac{\sum_g \Sigma_s^{g \rightarrow g'}(\bar{r}', \bar{\Omega} \rightarrow \bar{\Omega}')}{\Sigma_t^g(\bar{r}')} \right) \chi_g^*(\bar{r}', \bar{\Omega}', t'). \quad (82)$$

The additional weight factor $[\Sigma_t^g(\bar{r}')/\Sigma_t^g(\bar{r})]$ arises since Eq. (76) and its altered form (Eq. (82)), are actually flux-like equations, even though $\chi_g^*(\bar{r}, \bar{\Omega}, t)$ is adjoint to the emergent particle density $\chi_g(\bar{r}, \bar{\Omega}, t)$.

In a fashion analogous to the forward problem, the following new quantities are defined:

$$H_g(\bar{r}, \bar{\Omega}, t) \equiv \Sigma_t^g(\bar{r}) \chi_g^*(\bar{r}, \bar{\Omega}, t) \quad (83)$$

and

$$H_g(\bar{r}, \bar{\Omega}, t) = T_g(\bar{r} \rightarrow \bar{r}', \bar{\Omega}) G_g(\bar{r}', \bar{\Omega}, t'). \quad (84)$$

Since $\chi_g^*(\bar{r}, \bar{\Omega}, t)$ is a flux-like variable, the new variable $H_g(\bar{r}, \bar{\Omega}, t)$ can be regarded as an event density and $G_g(\bar{r}, \bar{\Omega}, t)$ like an emergent particle density. The defining integral equation for $G_g(\bar{r}, \bar{\Omega}, t)$ should be the proper basis for an adjoint random walk.

The defining equation for the adjoint event density function $H_g(\bar{r}, \bar{\Omega}, t)$ is obtained by considering the following altered form of Eq. (75):

$$\chi_g^*(\bar{r}, \bar{\Omega}, t) = \int dR \Sigma_t^g(\bar{r}') e^{-\beta_g^*(\bar{r}, R, \bar{\Omega})} [P_g^\psi(\bar{r}', \bar{\Omega}, t') + C_{g \rightarrow g'}(\bar{r}', \bar{\Omega} \rightarrow \bar{\Omega}') \chi_g^*(\bar{r}', \bar{\Omega}', t')] \quad (85)$$

Multiply Eq. (85) by $\Sigma_t^g(\bar{r})$ and rearrange as follows:

$$\Sigma_t^g(\bar{r}) \chi_g^*(\bar{r}, \bar{\Omega}, t) = \int dR \Sigma_t^g(\bar{r}) e^{-\beta_g^*(\bar{r}, R, \bar{\Omega})} [\Sigma_t^g(\bar{r}') P_g^\psi(\bar{r}', \bar{\Omega}, t') + C_{g \rightarrow g'}(\bar{r}', \bar{\Omega} \rightarrow \bar{\Omega}') \Sigma_t^{g'}(\bar{r}') \chi_g^*(\bar{r}', \bar{\Omega}', t')] \quad (86)$$

where

$$\check{C}_{g \rightarrow g'}(\bar{r}', \bar{\Omega} \rightarrow \bar{\Omega}') \equiv \frac{\Sigma_t^g(\bar{r}')}{\Sigma_t^{g'}(\bar{r}')} C_{g \rightarrow g'}(\bar{r}', \bar{\Omega} \rightarrow \bar{\Omega}') . \quad (87)$$

Noting that:

$$H_g(\bar{r}, \bar{\Omega}, t) = \Sigma_t^g(\bar{r}) \chi_g^*(\bar{r}, \bar{\Omega}, t) ,$$

$$\int dR \Sigma_t^g(\bar{r}) e^{-\beta_g^*(\bar{r}, R, \bar{\Omega})} = T_g(\bar{r} \rightarrow \bar{r}', \bar{\Omega}) ,$$

and

$$\Sigma_t^g(\bar{r}) P_g^\psi(\bar{r}, \bar{\Omega}, t) = P_g^\phi(\bar{r}, \bar{\Omega}, t) ,$$

Eq. (86) becomes

$$H_g(\bar{r}, \bar{\Omega}, t) = T_g(\bar{r} \rightarrow \bar{r}', \bar{\Omega}) [P_g^\phi(\bar{r}', \bar{\Omega}, t') + \check{C}_{g \rightarrow g'}(\bar{r}', \bar{\Omega} \rightarrow \bar{\Omega}') H_{g'}(\bar{r}', \bar{\Omega}', t')] . \quad (88)$$

A comparison of Eq. (88) with Eq. (84) reveals that

$$G_g(\bar{r}, \bar{\Omega}, t) = P_g^\phi(\bar{r}, \bar{\Omega}, t) + \check{C}_{g \rightarrow g'}(\bar{r}, \bar{\Omega} \rightarrow \bar{\Omega}') H_{g'}(\bar{r}, \bar{\Omega}', t) , \quad (89)$$

and the subsequent substitution of Eq. (84) into Eq. (89) yields the following defining equation for the adjoint emergent particle density:

$$G_g(\bar{r}, \bar{\Omega}, t) = P_g^\phi(\bar{r}, \bar{\Omega}, t) + \check{C}_{g \rightarrow g'}(\bar{r}, \bar{\Omega} \rightarrow \bar{\Omega}') T_{g'}(\bar{r} \rightarrow \bar{r}', \bar{\Omega}') G_{g'}(\bar{r}', \bar{\Omega}', t') . \quad (90)$$

Equation (90) is almost identical with Eq. (40) which defines the forward emergent particle density $\chi_g(\bar{r}, \bar{\Omega}, t)$ and also serves as the formal basis for the forward random walk. At this point, let us interpret Eq. (90) in terms of the transport of pseudo-particles called "adjunctons" in the $(P' \rightarrow P)$ direction of phase space. This presents two immediate problems:

- 1) The transport of the adjunctons from $\bar{r}' = \bar{r} + R\bar{\Omega}$ to $\bar{r}$ would be in a direction opposite to the direction vector $\bar{\Omega}$ -- therefore, the direction vector for the adjuncton should be $\hat{\bar{\Omega}} \equiv -\bar{\Omega}$, and $\bar{r}' = \bar{r} - R\hat{\bar{\Omega}}$.

- 2) The collision kernel should be interpreted as describing the ($E' \rightarrow E$) change in phase space experienced by the adjunction during its random walk; therefore, let

$$C_{g' \rightarrow g}(\bar{r}, \hat{\Omega}' \rightarrow \hat{\Omega}) \equiv \check{C}_{g \rightarrow g'}(\bar{r}, \bar{\Omega} \rightarrow \bar{\Omega}') = \sum_{g'} \int d\bar{\Omega}' \frac{\Sigma_s^{g \rightarrow g'}(\bar{r}, \bar{\Omega} \rightarrow \bar{\Omega}')}{\Sigma_t^{g'}(\bar{r})}. \quad (91)$$

Equation (91) may be rewritten in terms of a normalized collision kernel and a weight factor:

$$C_{g' \rightarrow g}(\bar{r}, \hat{\Omega}' \rightarrow \hat{\Omega}) = \sum_{g'} \int d\bar{\Omega}' \frac{\Sigma_s^{g \rightarrow g'}(\bar{r}, \bar{\Omega} \rightarrow \bar{\Omega}')}{\sum_g \int d\bar{\Omega} \Sigma_s^{g \rightarrow g'}(\bar{r}, \bar{\Omega} \rightarrow \bar{\Omega}')} \left(\frac{\sum_g \int d\bar{\Omega} \Sigma_s^{g \rightarrow g'}(\bar{r}, \bar{\Omega} \rightarrow \bar{\Omega}')}{\Sigma_t^{g'}(\bar{r})} \right)^*. \quad (92)$$

The selection of new phase space coordinates (group g , $\hat{\Omega} = -\bar{\Omega}$) is made from the normalized kernel and the weight of the adjunction is modified by the weight factor $[]^*$ which is no longer a simple non-absorption probability and may assume values in excess of unity. Therefore, there is no "analogue" scattering for adjunctions and the adjunction's weight may increase at some collisions.

Equation (90) can be rewritten as

$$G_g(\bar{r}, \hat{\Omega}, t) = P_g^\phi(\bar{r}, \hat{\Omega}, t) + C_{g' \rightarrow g}(\bar{r}, \hat{\Omega}' \rightarrow \hat{\Omega}) T_{g'}(\bar{r}' \rightarrow \bar{r}, \hat{\Omega}') G_{g'}(\bar{r}', \hat{\Omega}', t'), \quad (93)$$

which now corresponds to the transport of adjunctions and provides the desired basis for the adjoint random walk in the MORSE code. Note that the source of adjunctions is provided by $P_g^\phi(\bar{r}, \hat{\Omega}, t)$ which is related to $P_g^\phi(\bar{r}, \bar{\Omega}, t)$ as follows:

$$P_g^\phi(\bar{r}, \hat{\Omega}, t) = P_g^\phi(\bar{r}, -\bar{\Omega}, t), \quad (94)$$

which must be taken into consideration if the response function $P_g^\phi(\bar{r}, \bar{\Omega}, t)$ has angular dependence -- however, many physical situations permit an isotropic assumption for the $\bar{\Omega}$ -dependence.

A Monte Carlo solution of Eq. (93), the "integral emergent adjunction density equation," will generate data from which the adjunction flux $\chi_g^*(\bar{r}, \hat{\Omega})$ and

other quantities of interest can be determined. The general use of $\chi_g^*(\bar{r}, \hat{\Omega})$ must take into account the reversal of direction between adjunctons and real particles, i.e., $\hat{\Omega} = -\bar{\Omega}$. For example, consider the various ways of calculating the answer of interest:

$$\lambda = \sum_g \iiint P_g^\phi(\bar{r}, \bar{\Omega}, t) \phi_g(\bar{r}, \bar{\Omega}, t) d\bar{r} d\bar{\Omega} dt \quad (95)$$

$$= \sum_g \iiint \frac{P_g^\phi(\bar{r}, \bar{\Omega}, t)}{\Sigma_t^g(\bar{r})} T_g(\bar{r}' \rightarrow \bar{r}, \bar{\Omega}) \chi_g(\bar{r}', \bar{\Omega}, t') d\bar{r} d\bar{\Omega} dt$$

$$\lambda = \sum_g \iiint S_g(\bar{r}, \bar{\Omega}, t) \chi_g^*(\bar{r}, \bar{\Omega}, t) d\bar{r} d\bar{\Omega} dt = \iiint S_g(\bar{r}, \bar{\Omega}, t) \chi_g^*(\bar{r}, -\hat{\Omega}, t) d\bar{r} d\bar{\Omega} dt \quad (96)$$

$$\lambda = \sum_g \iiint S_c^g(\bar{r}, \bar{\Omega}, t) W_g(\bar{r}, \bar{\Omega}, t) d\bar{r} d\bar{\Omega} dt = \iiint S_c^g(\bar{r}, \bar{\Omega}, t) W_g(\bar{r}, -\hat{\Omega}, t) d\bar{r} d\bar{\Omega} dt \quad (97)$$

$$\lambda = \sum_g \iiint \frac{S_g(\bar{r}, \bar{\Omega}, t)}{\Sigma_t^g(\bar{r})} H_g(\bar{r}, \bar{\Omega}, t) d\bar{r} d\bar{\Omega} dt = \iiint \frac{S_g(\bar{r}, \bar{\Omega}, t)}{\Sigma_t^g(\bar{r})} H_g(\bar{r}, -\hat{\Omega}, t) d\bar{r} d\bar{\Omega} dt \quad (98)$$

$$\lambda = \sum_g \iiint \frac{S_g(\bar{r}, \bar{\Omega}, t)}{\Sigma_t^g(\bar{r})} T_g(\bar{r}' \rightarrow \bar{r}, \bar{\Omega}) G_g(\bar{r}', \bar{\Omega}, t') d\bar{r} d\bar{\Omega} dt \quad (99)$$

$$= \sum_g \iiint \frac{S_g(\bar{r}, \bar{\Omega}, t)}{\Sigma_t^g(\bar{r})} T_g(\bar{r}' \rightarrow \bar{r}, -\hat{\Omega}) G_g(\bar{r}', -\hat{\Omega}, t') d\bar{r} d\bar{\Omega} dt .$$

Further, if outward boundary crossings would be scored in the forward problem, the corresponding source adjunctons would be introduced in the inward direction. Likewise, adjunctons would be scored for entering a volume from which the source particles in the forward problem would be emitted. It should be noted that many sources and response functions are isotropic and the problem of direction reversal need not be considered.

4.12. REFERENCES

1. K. D. Lathrop, "DTF-IV, A FORTRAN-IV Program for Solving the Multigroup Transport Equation with Anisotropic Scattering," LA-3373, Los Alamos Scientific Laboratory (1965).
2. W. W. Engle, Jr., "A User's Manual for ANISN," K-1693 (1967).
3. W. A. Rhoades and F. R. Mynatt, "The DOT-III Two-Dimensional Discrete Ordinates Transport Code," ORNL-TM-4280 (1973).
4. W. Guber, et al., "A Geometric Description Technique Suitable for Computer Analysis of Both the Nuclear and Conventional Vulnerability of Armored Military Vehicles," MAGI-6701 (August, 1967).
5. M. O. Cohen, W. Guber, et al., "SAM-CE - A Three Dimensional Monte Carlo Code for the Solution of the Forward Neutron and Forward and Adjoint Gamma Ray Transport Equations," MR-7021 (DNA2830F) pp. 3.3-3.18 (November, 1971).
6. D. C. Irving, R. M. Freestone, Jr., and F. B. K. Kam, "Ø5R, A General Purpose Monte Carlo Neutron Transport Code," ORNL-3622 (1965).
7. C. L. Thompson and E. A. Straker, "Ø6R-ACTIFK, Monte Carlo Neutron Transport Code," ORNL-CF-69-8-36 (1969).
8. D. C. Irving, "Description of the CDC-1604 Version of the Ø6R Neutron Monte Carlo Transport Code," ORNL-CF-71-5-14 (1971).
9. C. E. Burgart and E. A. Straker, "XCHEKR - A Multigroup Cross Section Editing and Checking Code," ORNL-TM-3518 (1971).
10. F. H. Clark and N. A. Betz, "Importance Sampling Devices for Selecting Track Lengths and Directions After Scatter in Ø5R," ORNL-TM-1484 (1966).
11. F. H. Clark, "The Exponential Transform as an Importance-Sampling Device - A Review," ORNL-RSIC-14 (1966).

12. V. R. Cain, E. A. Straker, and G. Thayer, "Monte Carlo Path Length Selection Routines Based on Some Specific Forms of the Importance Function," ORNL-TM-1967 (1969).
13. M. B. Emmett, C. E. Burgart and T. J. Hoffman, "DOMINO, A General Purpose Code for Coupling Discrete Ordinates and Monte Carlo Radiation Transport Calculations," ORNL-4853 (1973).
14. Private Communication from W. W. Engle, Jr., ORNL.
15. F. H. Clark, Nucl. Sci. & Eng., 27, 235-239 (1967).
16. T. J. Tyrrell, "TDUMP - A Translation Routine for Use in Dumping FORTRAN Arrays," ORNL-CF-70-7-8 (1970).
17. S. N. Cramer, "Adjoint Gamma-Ray Estimation to the Surface of a Cylinder - Analyses of a Remote Reprocessing Facility," Nucl. Sci. & Eng., 79, 417-425 (1981).

INTERNAL DISTRIBUTION

1. Central Research Library	124. F. B. K. Kam
2. ORNL - Y-12 Technical Library	125. J. R. Knight
Document Reference Section	126. R. A. Lillie
3-4. Laboratory Records Department	127. J. L. Lucius
5. Laboratory Records, ORNL RC	128. R. E. Maerker
6-105. RSIC Library	129. F. C. Maienschein
106. L. S. Abbott	130. G. W. Morrison
107. J. M. Barnes	131. W. A. Rhoades
108. T. J. Burns	132. R. W. Roussin
109. S. N. Cramer	133. R. T. Santoro
110. H. L. Dodds	134. C. O. Slater
111-120. M. B. Emmett	135. J. S. Tang
121. W. W. Engle, Jr.	136. D. B. Trauger
122. T. A. Gabriel	137. G. E. Whitesides
123. T. J. Hoffman	138. A. Zucker

EXTERNAL DISTRIBUTION

139. W. H. Scott, Science Applications, Inc., P. O. Box 2351, La Jolla, CA 92037
140. V. R. Cain, Science Applications, Inc., 800 Oak Ridge Turnpike, Oak Ridge, TN 37830
141. W. H. Harless, General Electric Co., 310 DeGuigne Drive, Sunnyvale, CA 94086
142. John Kinch, U.S. Army Ballistic Research Laboratory, AMXBR-VL, Aberdeen Proving Ground, MD 21005
143. P. Stevens, Department of Nuclear Engineering, University of Tennessee, Knoxville, TN 37916
144. E. A. Straker, Science Applications, Inc., P. O. Box 2351, La Jolla, CA 92037
145. W. Zobel, Tennessee Valley Authority, 400 Commerce Avenue, Knoxville, TN 37902
146. Charles Myers, Tennessee Valley Authority, 400 Commerce Avenue, Knoxville, TN 37902
147. F. A. R. Schmidt, Institute for Kernenergetik, 7000 Stuttgart 80, Pffaffenwoldring 31, Germany
148. R. K. Disney, Westinghouse Electric Corporation, Advance Reactor Division, P. O. Box 158, Waltz Mill Site, Madison, PA 15663
149. R. W. Tayloe, Jr., Nuclear Criticality Safety Engineering Development, Goodyear Atomic Corp., P. O. Box 628, Piketon, Ohio 45661
150. Roy Castelli, Knolls Atomic Power Lab., Bldg. D-2, Room 114, 1 P. O. Box 1072, Schenectady, N. Y. 12301
151. George Schneider, U. S. Bureau of Mines Bldg. 20, Denver Federal Center, Denver, CO 80225

152. B. J. McGregor, Physics Division, AAEC, RES EST, Lucas Heights, Sidney, Australia
153. Dean C. Kaul, Science Applications, Inc., 1 Woodfield Place Bldg., Suite 819, 1701 E. Woodfield Road, Schaumburg, IL 60195
154. P. B. Hemmig, Reactor Physics Branch, Reactor Research & Technology Division, U. S. Department of Energy, Washington, DC 20545
155. John Kallfelz, Georgia Institute of Technology, Atlanta, GA
156. W. L. Bunch, Radiation and Shield Analysis, WADCO, P. O. Box 1970, Richland, WA 99352
157. C. M. Kim, Nuclear Analysis Group, Burns and Roe, Inc., 700 Kinderkamack Road, Oradell, NJ 07649
158. E. M. Gelbard, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439
159. L. Levitt, Rockwell International, P. O. Box 309, Canoga Park, CA 91304
160. M. Shirbacheh, M. S. 77102, Jet Propulsion Lab, 4800 Oak Grove Drive, Pasadena, CA 91109
161. M. Moghari, Energy Technology Center, 6003 Executive Blvd., Rockville, MD 20852
162. W. C. Hopkins, Bechtel Power Corporation, P. O. Box 607, 15740 Shady Dell Road, Gaithersburg, MD 20760
163. J. T. West, III, Los Alamos National Laboratory, P. O. Box 1663, Los Alamos, NM 87544
- 164-231. DNA Radiation Transport Distribution
232. Office of Asst. Manager for Energy Research and Development, Department of Energy/ORO, Oak Ridge, TN 37830
- 233-234. Technical Information Center, Oak Ridge, TN 37830