AD _____

A GEOMETRIC DESCRIPTION TECHNIQUE SUITABLE FOR COMPUTER ANALYSIS OF BOTH THE NUCLEAR AND CONVENTIONAL VULNERABILITY OF ARMORED MILITARY VEHICLES

MAGI-6701

August 1967

Walter Guber, Project Supervisor
Roger Nagel
Robert Goldstein
Phillip S. Mittelman
Malvin H. Kalos

Work Performed Under Contract No. DAAD05-67-C-0041
for
The Department of the Army
Ballistic Research Laboratories
Aberdeen Proving Ground, Maryland

Mathematical Applications Group, Inc.
180 South Broadway
White Plains, New York

MAGI-6701

A GEOMETRIC DESCRIPTION TECHNIQUE SUITABLE FOR COMPUTER ANALYSIS OF BOTH THE NUCLEAR AND CONVENTIONAL VULNERABILITY OF ARMORED MILITARY VEHICLES

Walter Guber, Project Supervisor
Roger Nagel
Robert Goldstein
Phillip S. Mittelman
Malvin H. Kalos

August 1967

Work Performed Under Contract No. DAAD05-67-C-0041
for
The Department of the Army,
Ballistic Research Laboratories
Aberdeen Proving Ground, Maryland
Richard C. Hoyt, Technical Supervisor

Mathematical Applications Group, Inc.
180 South Broadway
White Plains, New York

ACKNOWLEDGMENTS

It is a pleasure to acknowledge the advice, guidance, and technical cooperation provided during the course of this study by Messers. Richard Hoyt and Ronald Marking of the Weapon Systems Laboratory of Ballistic Research Laboratory. We are also indebted to the computer operations staffs of the Aberdeen BRLESC and the NYU-AEC Computing Center for their wholehearted cooperation during the debugging and operation of the computer programs presented in this report.

ABSTRACT

A suitable three-dimensional geometrical description technique and two computer programs, MAGIC and SAM-C, were developed and demonstrated for application to the computer analysis of both the nuclear and conventional vulnerability of armored military vehicles. The geometric technique was applied, for demonstration purposes, to the M60A1 tank. A single body of geometric target data for the M60A1 was used by the MAGIC program (Mathematical Applications Group, Inc., Code) to address the vulnerability of the tank to attack from any attack angle by conventional armor-defeating projectile systems, and by the SAM-C program (Stochastic Approximation Method - Combinatorial) for the determination of internal nuclear radiation dose. The radiation dose (in selected target regions) is obtained for primary neutrons, primary gammas, and for secondary gammas produced by neutron interactions occurring either outside or within the vehicle. The critical feature of the effort is the

successful development of a technique which inputs the engineering design detail and allows for the subsequent vulnerability determination for either nuclear or conventional combat weapons.

The two programs, MAGIC and SAM-C, are currently operational on both the CDC-6600 and BRL-BRLESC computers. Both programs are written in FORTRAN.

FOREWORD

The Aberdeen Ballistic Research Laboratory has, as one of its missions, the evaluation of the vulnerability of armored vehicles to attack by both nuclear and conventional armor-defeating weapons. Prior to the present effort, independent computer programs had been developed to assist in such evaluations.

On the one hand, there was available the UNC-SAM Program (United Nuclear Corporation - Stochastic Approximation Method) for the determination of nuclear flux (and subsequently dose) through the use of the Monte Carlo technique. This program, given a geometrical and nuclear description of the target vehicle and surrounding atmosphere and ground environment, could calculate radiation fluxes and doses at specified locations inside and outside the vehicle. The geometric description, however, was restricted to an assembly of nonintersecting bodies and was only an approximate representation of the true geometry.

On the other hand, there were available the ASEB programs (Armored Systems Evaluation Branch) for the determination of target hit and target kill for armor-defeating direct fire weapon systems. These ASEB codes require that, for any given direction of attack, the vehicle be described in terms of the ordered sequence and thickness of material layers which a projectile would encounter in penetrating the vehicle. This description was established for each of a number of rectangular prisms defined by projecting a rectangular grid through the vehicle along the direction of attack. With such detailed knowledge of both the sequential composition of the target armor and vulnerable components, as well as the positions of this array within the target plane, the ASEB codes would compute the effectiveness of a weapon system. This effectiveness, a function of aim point and impact dispersion at the target plane, was given in terms of target hit, mobility kill, firepower kill, etc.

The restrictive portion of the analysis of conventional projectiles was preparation of the target descriptive data. These data were prepared manually by working directly with the engineering drawings for each attack azimuth of interest. This manual approach was readily applied to attack azimuths of 0° and 90°, or to direct overhead attack, since appropriate engineering drawing views were usually available. Securing target data for other attack azimuths such

as 30° and 60° was considerably more difficult. If attack analysis was desired where both azimuth and elevation angles were other than along an axis common to the engineering drawings, the preparation of the required target data was possible but realistically impractical.

It was obvious, therefore, that if the ASEB codes were to be used to any greater extent, a computerized method would have to be developed to provide the necessary target input data for any desired angle of attack. Further, since the nuclear analysis code also required detailed target geometry input for accurate results, it was felt that the geometry description technique should be compatible for both nuclear and conventional weapon analysis. In this way a single target description would serve both requirements. However, it was not at all obvious how one would either adequately describe a complicated target or generate the required sequential data from such geometric target data. One fact that had been established by experience was that the use of independent (nonoverlapping) geometric figures would necessarily result in an approximation of actual target geometry that would be of marginal usefulness.

Recognizing the need, therefore, for a computerized method to achieve target data for the ASEB conventional projectile lethality codes, and the need to input an

accurate target description for nuclear analysis, it was decided to support an effort in this direction. The MAGI "combinatorial geometry" technique was selected for development because it permitted the combination of geometric figures to form many practical geometric figures of interest, and the geometry developed could be used for both nuclear and conventional weapon analysis.

The end objective was to achieve development of a practical technique and realize actual results for an actual target. The M60A1 tank was chosen as the target to be described and the specific results to be obtained were (1) the sequential array data required by the ASEB conventional projectile lethality codes based on any angle of attack specified, and (2) the actual nuclear dose values within the tank. These objectives were met.

The "combinatorial geometry" technique was developed and the computer programs prepared and tested. These are: MAGIC, which provides the sequential target data for the ASEB codes; and SAM-C which computes the nuclear dose. The M60A1 tank was described and read into the computer. Several target sequential data arrays were prepared, and nuclear dose was computed with the vehicle at various ranges from a specified positioned source.

The report describes the development work accomplished by MAGI during the contractual period. In addition to theoretical discussions regarding the various programs de-

veloped, it can also serve as a programmer's guide for those who may want to implement the programs on their own computer for analysis of other target configurations. Although the report describes the work accomplished, it intentionally does not deal at any great length with the many possible applications to studies of the effectiveness of other weapon systems. Suffice it to say, the objectives of the work were met and a number of tools now exist that should materially assist those groups who are performing design, vulnerability, and effectiveness analyses of weapon systems.

Richard C. Hoyt
Technical Supervisor
Ballistic Research Laboratory
Aberdeen, Maryland

TABLE OF CONTENTS

TABLE

FIGURES

1. INTRODUCTION AND DESCRIPTION OF TECHNICAL EFFORT

1.1 INTRODUCTION

1.1.1 Scope

This report describes the development of a significant new geometric technique - the "combinatorial geometry" technique - for addressing the nuclear and conventional vulnerability of armored vehicles. It makes possible the computer representation of complicated three-dimensional objects in any required degree of detail. In addition, the report provides a detailed discussion of the two computer programs for which this technique was developed. The first, SAM-C, is a nuclear dose analysis code which combines the latest advances in Monte Carlo importance sampling techniques and detailed basic physical data capabilities with this new combinatorial geometry technique. The second, MAGIC, is a code which utilizes this new technique to convert geometric and functional (e.g., armor, road wheel, etc.) target description data into the penetration data (line-of-sight thicknesses, surface obliquities, etc.)

required as input to other programs which evaluate the vulnerability of the target when subjected to projectile attack.

1.1.2 Readership

This report is directed to three distinct categories of readers:

1. The weapons systems/operations research analysts who are constantly searching for improved methods and techniques applicable to weapon system vulnerability analysis.
2. The mathematician/physicist methodologist who must find ways and means of implementing and understanding the new techniques and approaches.
3. The computer programmers and system analysts who must translate the concepts of the analysts and the theorems of the methodologists into workable production computer codes.

1.1.3 Organization

In order to accomplish the objectives set forth in Section 1.1.1, the body of this report is divided into three primary sections. Section 2 presents an introduction to the basic geometric technique common to both the SAM-C and MAGIC programs. Section 3 presents further information on this combinatorial geometry technique and indicates how, through the addition of certain input data, a computer

program applicable to projectile vulnerability analysis results. Section 4 discusses the implementation of this technique in a Monte Carlo radiation transport code to evaluate nuclear dose levels.

Brief individual discussions of combinatorial geometry and its application to MAGIC and SAM-C follow.

1.2 DESCRIPTION OF TECHNICAL EFFORT

1.2.1 Combinatorial Geometry

Combinatorial geometry is a new and significant advance in the state-of-the-art of representing - in a computer - a complex three-dimensional structure. In effect, one represents a structure such as a tank in terms of sums, differences, and intersections of relatively simple bodies such as spheres, cylinders, etc. (described in Section 2). The input for such a description consists of the geometric location of the simple bodies and their dimensions, followed by a region definition table consisting of a series of equations defining each particular region of the structure in terms of the basic bodies. For example, if the total structure is a tank, then one region would be the gun barrel, which might be represented as the material located between two concentric cylinders.

When the geometrical description is completed, a material composition must be assigned to each region (e.g., steel, copper), and each region must be identified (e.g., gun, driver, etc.).

Thus, one can achieve a total geometric, material, and functional description of the vehicle as a three-dimensional object. It should be noted that this description can be as detailed as desired by including, for example, individual shells, motor components, etc., or it may be quite rudimentary depending upon the application. At any rate this description, once prepared for the computer, can be used for a variety of studies.

Under the current contractual effort, the combinatorial geometry technique has been applied to the evaluation of vehicle vulnerability to conventional weapons and to the evaluation of nuclear radiation. The programs have been thoroughly tested and applied to armored vehicles of current interest to the U. S. Army. The programs are operational on the Aberdeen BRLESC and the NYU CDC-6600 computers.

1.2.2 Application of Combinatorial Geometry to Conventional Vulnerability (MAGIC Program)

An evaluation of the vulnerability of a vehicle under attack by a particular weapon is an extremely complex task. One must first establish the various possible types of "kill" as, for example, "firepower kill," "mobility kill," or "total kill," etc. Each individual component of the target must be assessed to determine its importance to each type of kill. For example, destruction of a tank transmission will result in mobility kill but may have no effect

on firepower kill.

The probability of destruction of a "vulnerable" component (a component important to one of the kills) must be determined as a function of the properties of the incident penetration. Properties such as residual energy and mass are important parameters in such an assessment. The MAGIC program, discussed in detail in Section 3, has been designed to work in association with existing ASEB vulnerability programs at Ballistic Research Laboratories to perform the assessment of vehicle vulnerability.

The first task is to set up the geometrical description of the vehicle in combinatorial geometry input form. These data are entered into the MAGIC program, say, via punched cards. Subsequently, the vulnerability analyst selects the attack direction. Since the description is truly three dimensional, the attack direction need only be specified in terms of an azimuth and elevation angle. No modification of the vehicle description is necessary for different angles of attack. The program sets up an attack plane perpendicular to the attack direction and proceeds to shoot rays through the attack plane, with each ray directed parallel to the attack direction. The rays are started within rectangular areas on the attack plane. The size of the attack plane is specified as input. On hitting the target vehicle, the angle of incidence, nature

of region hit, and normal and line-of-sight thickness of the region hit are determined. The ray continues through the vehicle, determining the same information for each region it traverses, and is finally terminated on leaving the far side of the vehicle. The data, thus generated, are put out in printed and punched card form for use in a subsequent ASEB program. This latter program calculates not only the residual penetration energy after encountering a region with the defining features established by the MAGIC program, but, additionally, the significance of the encounter with this defined region in terms of the probability of the various kills in question. If there is residual energy, this process is continued with the next region and so on. Thus, the combination of the MAGIC and ASEB programs forms a complete mathematical framework in which one can accommodate all the geometrical and material information concerning the vehicle as well as the penetration and lethality properties of the projectile. Since each program runs several minutes on a large computer, a number of design parameters can be investigated.

1.2.3 Application of Combinatorial Geometry to Nuclear Analysis (SAM-C Program)

In addition to attack by conventional weapons, modern vehicles may be exposed to attack by nuclear weapons. Generally speaking, the crew of a vehicle is more sensitive to radiation than are any of the instruments or equipment.

Thus, the primary problem is to establish the radiation levels to which the crew is exposed.

A number of calculational techniques exist for determining the radiation dose in free air or at the air-earth interface. However, until the development of the combinatorial geometry technique and its incorporation into the SAM-C program, as discussed in Section 4, no technique existed for performing detailed calculations of the transport of radiation through an accurate model of both the vehicle and its environment.

Exactly the same geometrical input data used in the MAGIC program is used in the SAM-C program. This program, using Monte Carlo techniques, simulates the actual flight of radiation, from its source (burst point, for instance) through the environment to the dose region. The following interactions of radiation with matter are treated:

1. Gamma Rays
    a. Compton scattering
    b. Absorption by the photoelectric and pair production processes
2. Neutrons
    a. Inelastic scattering
        Discrete level
        Continuum

b. Elastic scattering

Isotropic

Anisotropic

c. Absorption.

The program also can treat time-dependent problems and can be used to evaluate transient radiation effects. The SAM-C program has been applied very successfully to studies of the transport of weapon radiation through the air to and through a detailed combinatorial geometry representation of an M60 tank. In these calculations, the neutrons were followed down to thermal energies and secondary gamma-ray production in the tank structure and personnel were taken into account.

1.3 EXTENSION OF TECHNICAL EFFORT

The objectives of this contract effort required that specific output be achieved by practical means. It should be noted that many portions of the MAGIC and SAM-C programs have not yet been optimized to reduce running time, computer memory requirements, or the labor required to prepare input. This aspect of the effort has been continually recognized by both MAGI and the Government, but optimization has been deferred in favor of producing a working practical system. Such optimization should be pursued.

It is now obvious that once the geometric description

has been prepared it can be used in a variety of other studies with appropriate programming. For example, these are:

1. Evaluation of gamma dose in vehicles traversing fall out fields;
2. Center of gravity and moments of inertia determinations;
3. Graphic display of the geometric configuration;
4. Evaluation of lethality of spall fragments resulting from target penetration;
5. Evaluation of lethality of arbitrary bursts;
6. Evaluation of shock wave effects on radiation dose.

This Document Contains
Missing Page/s That Are
Unavailable In The
Original Document

OR are
Blank pgs
that have
Been Removed

**BEST
AVAILABLE COPY**

2. DESCRIPTION OF THE COMBINATORIAL GEOMETRY TECHNIQUE

In order to perform computer studies concerning a vehicle one must first be able to prepare a mathematical model of the vehicle, and its environment. The combinatorial geometry technique has been developed to permit a model to be produced which is both accurate and suitable for a ray-tracing analysis program. The latter feature is important since both the conventional and nuclear vulnerability analysis programs involve the tracing of rays through geometrical models.

In effect the geometric description subdivides the problem space into unique regions. This is achieved through the use of nine specific geometric bodies (closed surfaces) and the orderly identification of the combination of those bodies which define a region (space volume). The bodies which will be discussed further in Input Preparation (Section 2.2) are as follows:

1. Rectangular Parallelepiped (RPP)

2. Box

3. Sphere

4. Right Circular Cylinder

5. Right Elliptical Cylinder

6. Truncated Right Angle Cone

7. Ellipsoid

8. Right Angle Wedge

9. Arbitrary Convex Polyhedron of four, five or six sides (each side having three or four vertices).

Except for the RPP's, all bodies may be arbitrarily oriented with respect to the x, y, z coordinate axes used to determine the space. The RPP's are special bodies used to divide the overall geometry into smaller regions. This reduces the time required in ray tracing when many physical regions are present. It should be noted that the sides of an RPP must be parallel to the coordinate axes.

2.1 REGION DESCRIPTION TECHNIQUE

The basic technique for the description of the geometry consists of defining the location and shape of the various physical regions (wall, equipment, etc.) in terms of the intersections and unions of the volumes contained in a set of simple bodies. A special operator notation involving the symbols (+), (-), and (OR) is used to describe the intersections and unions. These symbols are

used by the program to construct tables used in the ray-tracing portion of the problem.

If a body appears in a region description with a (+) operator, it means that the region being described is wholly contained in the body.

If a body appears in a region description with a (-) operator, it means that the region being described is wholly outside the body.

If the body appears with an (OR) operator, it means that the region being described includes all points in the body. In some instances, a region may be described in terms of subregions lumped together by (OR) statements.

The technique of describing a physical region is best illustrated by an example. Consider an object composed of a sphere into which is inserted a cylinder. This is shown in cross section in Fig. 2.1(a).

To describe the object, we take a spherical body penetrated by a cylindrical body (Fig. 2.1(b)). Each body is numbered. Consider the sphere as body No. 1 and the cylinder as body No. 2. If the materials in the sphere and cylinder are the same, then they can be considered as one physical region, say region 100 (Fig. 2.1(c)).

The description of region 100 would be:

100 = (OR 1) (OR 2).

This means that a point is in region 100 if it is either

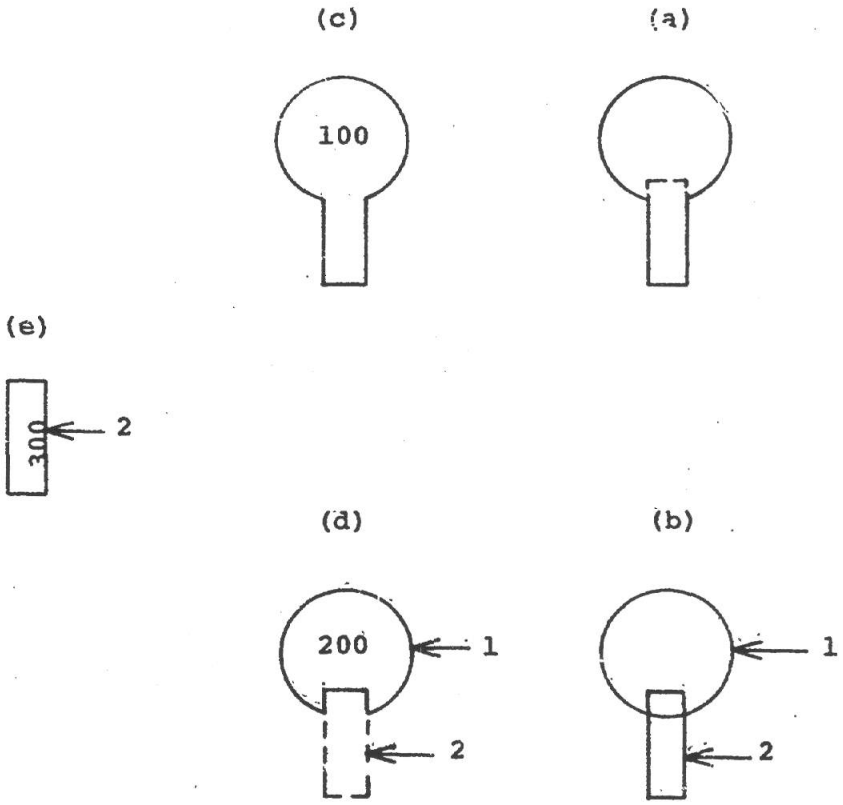


Fig. 2.1 - Regions Produced by Intersections and Unions of Sphere and Cylinder

inside body 1 or inside body 2.

If different materials are used in the sphere and cylinder, then the sphere with a cylindrical hole in it would be given a different region number (say 200) from that of the cylinder (300).

The description of region 200 would be {Fig. 2.1(d)}:

200 = (+1) (-2).

This means that points in region 200 are all those points inside body 1 which are not inside body 2.

The description of region 300 is simple {Fig. 2.1(e)}:

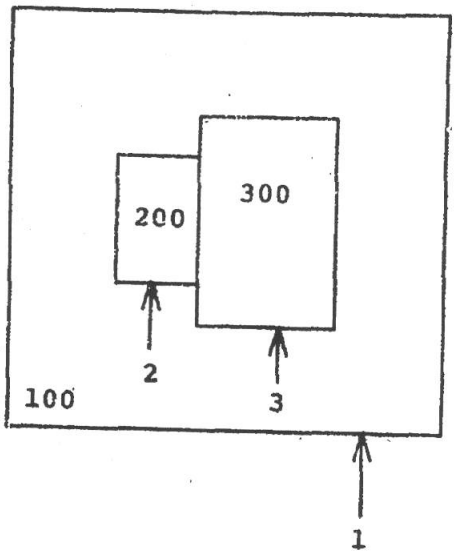
300 = (+2).

That is, all points in region 300 lie inside body 2.

This technique, of course, can be applied to combinations of more than two bodies and such region descriptions could conceivably contain a long string of (+), (-) and (OR) operators. The important thing to remember is that every spatial point in the geometry must be located in one and only one region. Further examples are given in Section 2.2.2.

A rule of construction imposes the additional restriction that region descriptions include negation (-) of buttressing surfaces which are not otherwise necessary to the logical description of the region. That is, if boxes 2 and 3 are in contact, as on the following page, the description of region 200 must be 200 = (+2) (-3). Region

300 should be defined as 300 = (+3) (-2).



The user of the program will specify the geometry by establishing two tables. The first table will describe the type and location of the set of bodies used in the geometrical description. The second table will identify the physical region in terms of these bodies. The computer program processes these tables to put the data in the form required for ray tracing. All of the space must be divided into regions, and once again no point may be in more than one region.

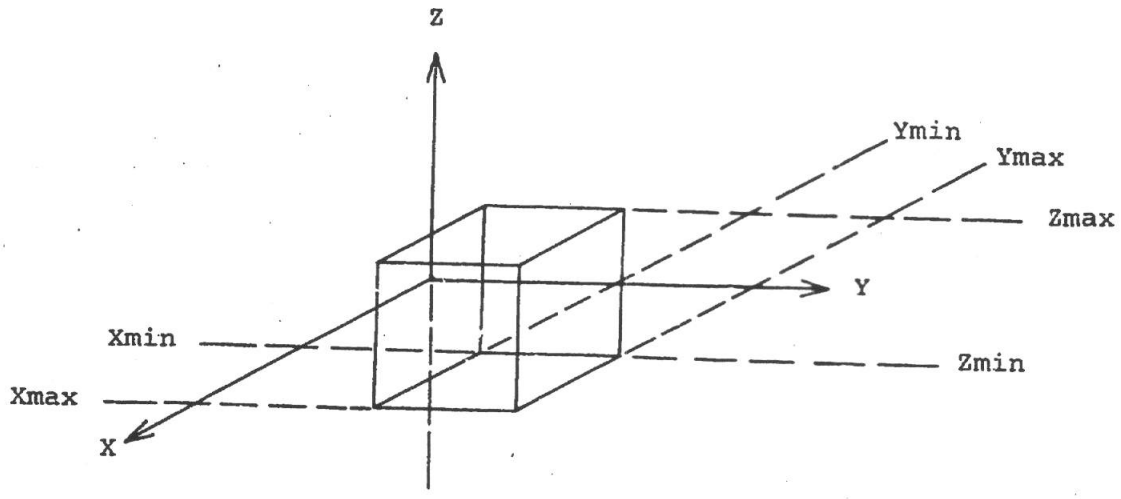
2.2 INPUT PREPARATION

2.2.1 Description of Input Parameters

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

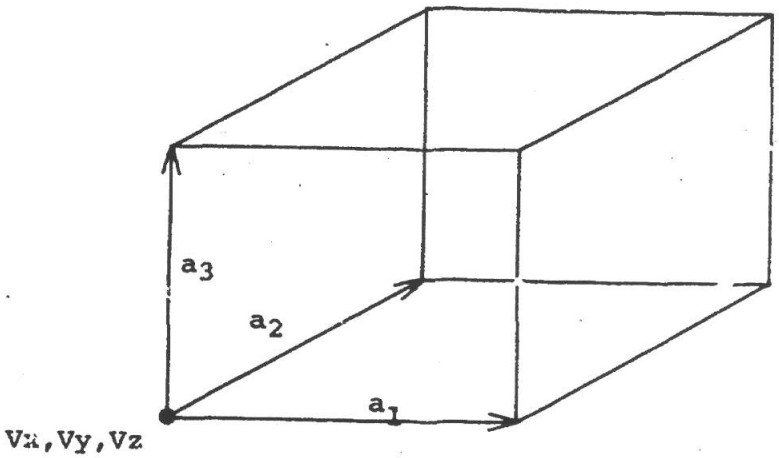
1. Rectangular Parallelepiped (RPP)

   These bodies are used for gross subdivisions of the geometry and must have bounding surfaces parallel to the coordinate axes. The entire geometry must be enclosed in an RPP. Specify the maximum and minimum values of the x, y, and z coordinates which bound the parallelepiped.



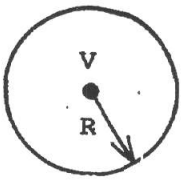
2. Box (BOX)

Specify the vertex $\underline{V}$ at one of the corners by giving its (x,y,z) coordinates. Specify a set of three mutually perpendicular vectors, $\underline{a_i}$, representing the height, width, and length of the box, respectively. That is, the x,y, and z components of the height, width, and length vectors are given.



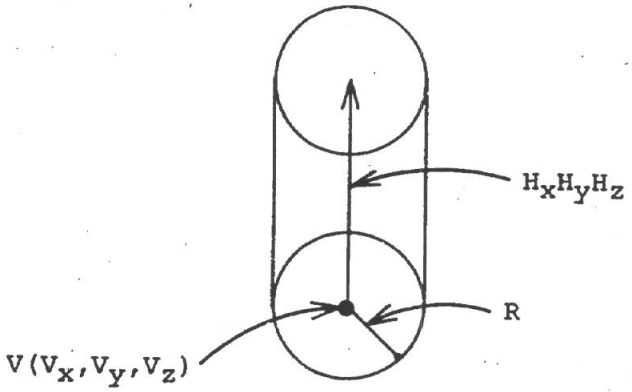
3. Sphere (SPH)

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



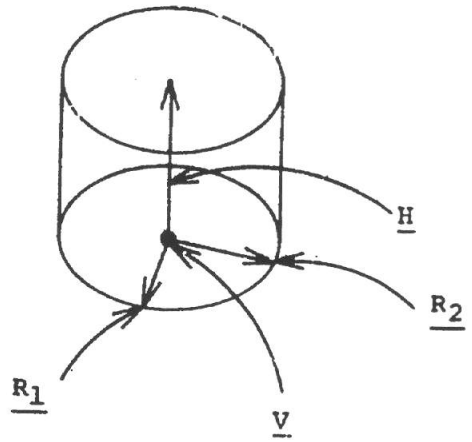
4. Right Circular Cylinder (RCC)

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 base radius.



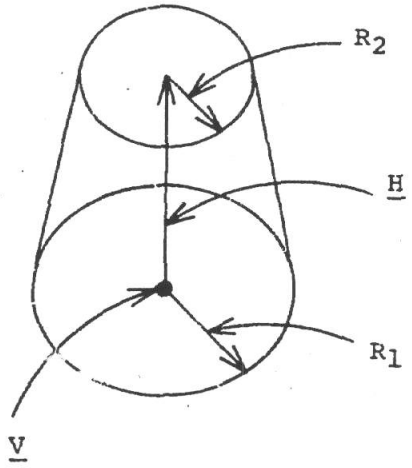
5. Right Elliptical Cylinder (REC)

Specify coordinates of the center of the base ellipse, a height vector, and two vectors in the plane of the base defining the major and minor axes.



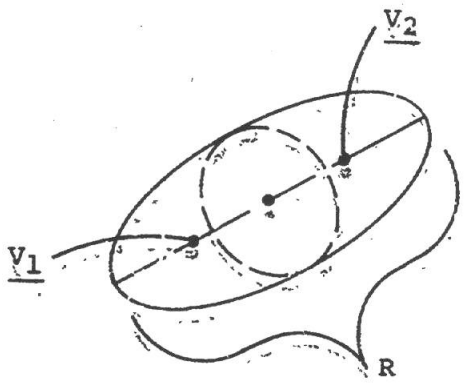
6. Truncated Right Angle Cone (TRC)

Specify a vertex $\underline{V}$ at the center of the lower base, the height vector, $\underline{H}$, expressed in terms of its x, y, z components, and two scalars, R_1 and R_2, denoting the radii of the lower and upper bases.



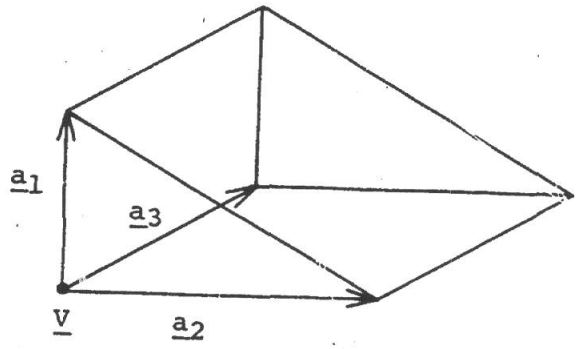
7. Ellipsoid (ELL)

Specify two vertices, V_i, denoting the coordinates of the foci and a scalar, R, denoting the length of the major axis.



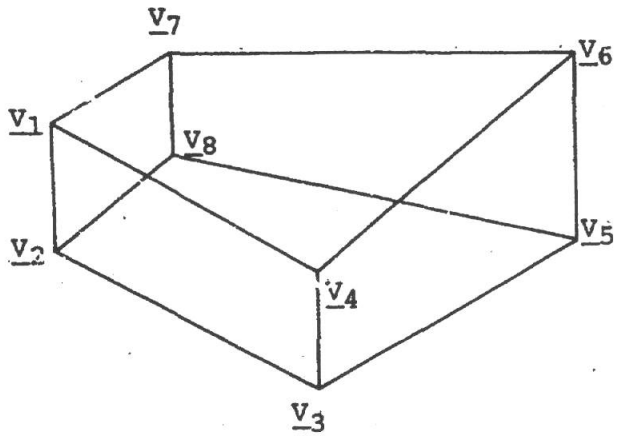
8. Right Angle Wedge (RAW)

Same input as for the boxes. However, $\underline{a}_1$ and $\underline{a}_2$ describe the two legs of the right triangle of the wedge.



9. Arbitrary Polyhedron (ARB)

Assign an ordinal number (1 to 8) to each vertex. For each vertex, give the x, y, z coordinates. For each side of the figure list the ordinal vertex numbers. The vertices and side descriptions may be given in any order. An example is given later.



2.2.2 Examples of Region Descriptions

Some representative geometries and their input descriptions are shown below.

Example 1 - Two Spheres Within an RPP (See Fig. 2.2)

The body input table is shown below.

TABLE I - BODY INPUT DESCRIPTION

Body	Type of Data Required
1	List the six bounding coordinate values (x_{min}, x_{max}, Y_{min}, Y_{max}, z_{min}, z_{max})
2	List the vertex and radius of sphere 2
3	List the vertex and radius of sphere 3

One possible region input table is shown below.

TABLE II - REGION DESCRIPTION

Region	Input
100	(+1) (-2) (-3) (Region 100 is composed of all points interior to RPP No. 1 and exterior to spheres 2 and 3)
200	(+3) (-2) (Region 200 is composed of all points interior to sphere 3 and exterior to sphere 2)
300	(+2) (+3) (Region 300 is composed of all points which are in sphere 2 and are also in sphere 3)

Region	Input
400	(+2) (-3) (Region 400 is composed of all points interior to sphere 2 and exterior to sphere 3)
500	(OR 2) (OR 3) (If desired, one region, the total of regions 200, 300, and 400, can be defined as region 500)

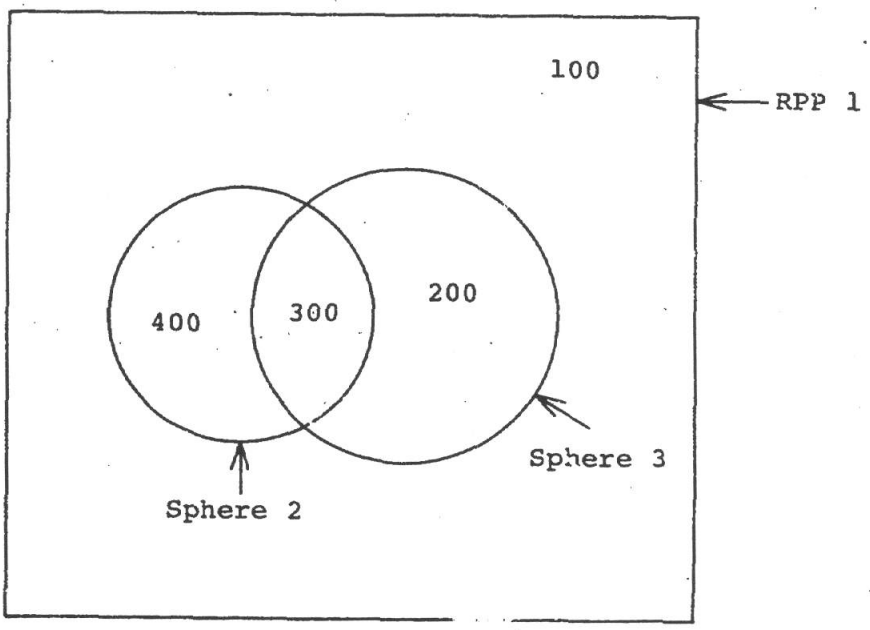


Fig. 2.2 - Regions Produced by Intersections and Unions of Two Spheres

Example 2 - Cylinder Divided into Two Regions by a Box and with a Sphere at One End (See Fig. 2.3)

TABLE I - BODY INPUT DESCRIPTION

Body	Type of Data Required
1	List the six bounding coordinates of the RPP
2	List the vertex, radius, and height vector of cylinder
3	List center and radius of sphere
4	List coordinates of one corner and components of three vectors representing sides of box

The region input is as follows.

TABLE II - REGION DESCRIPTION

Region	Input
100	(+1) (-2) (-3) (All points interior to the RPP and exterior to the cylinder and sphere. Note that region 100 includes all of the space contained inside body 4, except that portion inside cylinder 2. This space can be assigned a special region number, if desired. If, as in this example, it is not desired, it is not necessary)
200	(+2) (-4) (All points interior to the cylinder, and outside the box)

Region	Input
300	(+3) (-2) (All points interior to the sphere and external to the cylinder)
400	(+2) (+4) (All points interior to the cylinder and also inside the box)

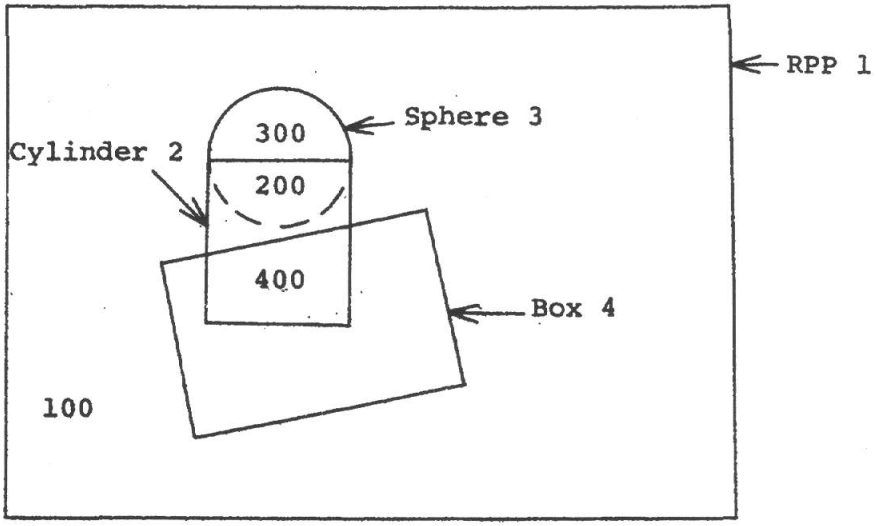


Fig. 2.3 - Regions Produced by Intersections and Unions of Sphere, Circular Cylinder, and Box

Example 3 - Multiple Region Capability - Cylinder Containing Two Spheres, All Inside an RPP (See Fig. 2.4)

TABLE I - BODY INPUT DESCRIPTION

Body	Type of Data Required
1	List RPP data
2	List cylinder input
3	List sphere input
4	List sphere input

TABLE II - REGION DESCRIPTION

Region	Input
100	(+1) (-2)
200	(OR 3) (OR 4) (All points interior to 3 or 4)
300	(+2) (-3) (-4) (All points in the cylinder but not in the spheres)

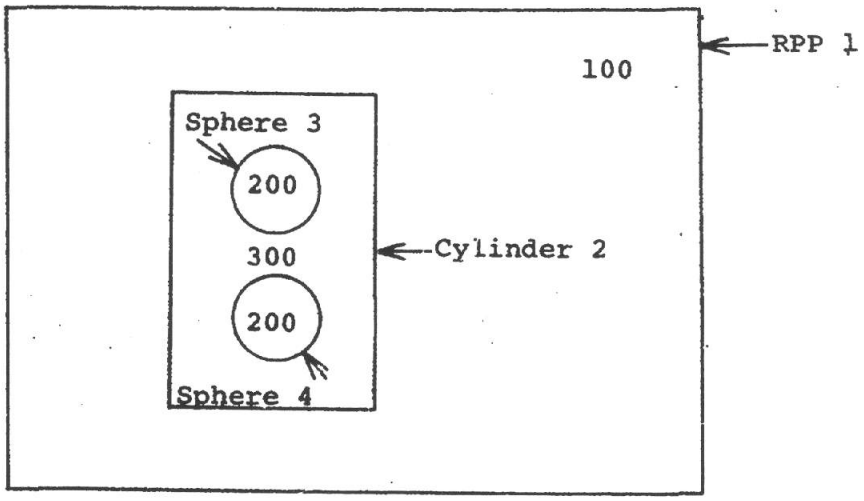


Fig. 2.4 - Example of Physical Region Produced from Unconnected Regions Using "OR" Statement

2.2.3 Card Input Formats

The following punched cards are needed to describe the geometry and must appear in the order in which they are described below.

1. *Comment Card*

This card contains a 60-column alphanumeric title of the problem and will appear in the printed output. The input format is (10A6).

2. Quantity and Scaling Card

This card contains seven quantities, the first six of which are integers. The input format is (6I10,E10.2).

Quantity	Definition	Columns
NRPP	The number of RPP's in the geometry	1-10
NTRIP	The number of triplets. (A triplet is a set of vector coordinates used in the description of more than one body.)	11-20
NSCAL	The number of scalars. (A scalar is a number used in the description of more than one body.)	21-30
NBODY	The number of bodies in the geometry (not counting RPP's)	31-40
NRMAX	The number of regions in the geometry	41-50
IPRIN	Printing option. If other than zero, the geometry data stored in the	51-60

Quantity	Definition	Columns
IPRIN (con'd.)	(MASTER-ASTER) array will appear in the printed output. If zero, this printing will be suppressed.	
DIST	Scale factor (floating point format) If 0.0, blank, or 1.0, the scale factor will be assumed equal to 1.0 by the program. Otherwise, all dimensions will be scaled by this quantity.	61-70

3. RPP Cards

Give one card for each RPP (a total of NRPP cards). The card form is as follows. Where each coordinate defines one of the bounding planes of the RPP, the input format is (6E12.6).

Coordinate	Xmin	Xmax	Ymin	Ymax	Zmin	Zmax
Columns	1-12	13-24	25-36	37-48	49-60	61-72

4. Triplet Cards

Give one card for each triplet (a total of NTRIP cards). Each card contains three floating point numbers describing the (x,y,z) components of the triplet. The input format is (3E12.6) starting in Column 1. Omit if NTRIP = 0.

5. Scalar Cards

Give one card for each scalar (a total of NSCAL cards). Each card contains one floating point number giving the value of the scalar. The input format is (E12.6) starting in Column 1. Omit if NSCAL = 0.

6. Body Cards

The computer assigns to each body an ordinal number which depends on the order in which the body cards are read in. Therefore, it is most important that the card sequence match the numbering sequence used in the region descriptions. Note that no gaps may be left in the body numbering sequence and that the RPP numbers must precede numbers assigned to all other bodies.

The data describing each body may be given in one of two formats. The first of these is in more general use and employs floating point input data to specify the actual location and dimensions of a body. This format will be described first.

Format 1

Eight different body types may be employed. The standard format for each body is as follows.

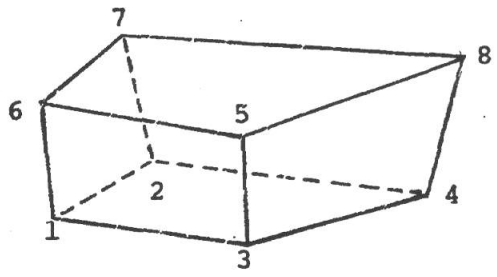
Columns	Input
1	Any nonblank Hollerith character (directs the code that floating point input will be used)
2-3	Two characters of arbitrary Hollerith data
4-6	Three-letter body identifier. Note that the numbers assigned to RPP's must be lower than the numbers assigned to other bodies.
7-10	Four characters of arbitrary Hollerith data
11-70	Divided into six floating point fields of 10 columns each. Body dimensions are given here.

Table 2.1 describes the input required for each body. The quantities V, H, etc., were defined in Section 2.2.

TABLE 2.1 - INPUT REQUIRED FOR EACH BODY TYPE

Body Type	3-letter ID	Card Columns 11-20	21-30	31-40	41-50	51-60	61-70	Number of Cards Needed
Box	BOX	Vx H2x	Vy H2y	Vz H2z	H1x H3x	H1y H3y	H1z H3z	1 of 2 2 of 2
Sphere	SPH	Vx	Vy	Vz	R	-	-	1
Right Circular cylinder	RCC	Vx R	Vy -	Vz -	Hx -	Hy -	Hz -	1 of 2 2 of 2
Right elliptic cylinder	REC	Vx R1x	Vy R1y	Vz R1z	Hx R2x	Hy R2y	Hz R2z	1 of 2 2 of 2
Ellipse	ELL	V1x L	V1y -	V1z -	V2x -	V2y -	V2z -	1 of 2 2 of 2
Truncated cone	TRC	Vx L1	Vy L2	Vz -	Hx -	Hy -	Hz -	1 of 2 2 of 2
Right angle wedge	RAW	Vx H2x	Vy H2y	Vz H2z	H1x H3x	H1y H3y	H1z H3z	1 of 2 2 of 2
Arbitrary polyhedron	ARB	V1x V3x V5x V7x	V1y V3y V5y V7y	V1z V3z V5z V7z	V2x V4x V6x V8x	V2y V4y V6y V8y	V2z V4z V6z V8z	1 of 5 2 of 5 3 of 5 4 of 5
		Face Descriptions (See Note on following page)						5 of 5

Note: Each of the six faces of an ARB are described by a four-digit integer giving the number of the four vertex points at the corners. The point numbers for each face must be entered in either clockwise or counterclockwise order. The format is 6(1X, I4) starting in Column 11. An example is shown below.



FACE	1	2	3	4	5	6
PTS	1653	3584	4278	1672	1342	5678

Format 2

The second format for body input makes reference to previously read in triplet and scalar data. This format cannot be used to describe the ARB body. The following input format is used for all other bodies.

Columns	Input
1-3	Blank (the blank in Column 1 informs the code that triplets and scalars will be referenced)

Columns	Input
4-6	Three-letter body identifier. Note that RPP numbers must be lower than those for any other body.
7-10	Blank
11-70	Divided into six integer fields of 10 columns each. These fields contain the identifying number of the triplet or scalar which is used to specify a particular input quantity. For example, instead of specifying Vx, Vy, Vz for the vertex of a box, as in Format 1, the number of the triplet vector having those components is given. The number of the triplet or scalar is determined by its position in the sequence of triplet or scalar identification cards. The first triplet defined is triplet No. 1. The first scalar defined is scalar No. 1.

It should be noted that each body must be described completely by either Format 1 or Format 2. However, different formats may be employed for different bodies (the ARB is restricted to Format 1).

7. Region Cards

Each region must be numbered and described by a logical combination of the bodies which make up that region. A total of NRMAX regions must be described. Use as many cards as necessary to describe each region and begin each region on a new card. The input format, described below, is (I5, 1X, 9(A2, I5)).

Columns	Input
1-5	Region Number.
6	Blank.
7-8	Insert (OR) if needed. Otherwise, leave blank.
9	Insert either (+) or (-).
10-13	Number of body.
14-69	Divided into eight fields, each being similar to Columns 7-13. The first two columns of each field are reserved for the OR operator. The third column is for the (+) or (-) operator. The last four columns are for the body number.

Use as many cards of the above type as needed to complete a region description, but leave Columns 1-6 blank on all continuation cards.

The last region description card must be followed by a card containing a (-1) in Columns 4 and 5. This informs the code that all regions have been described.

3. DESCRIPTION OF THE MAGIC PROGRAM FOR CONVENTIONAL PROJECTILE ANALYSIS

This section of the report contains a complete description of the MAGIC program. A general discussion of the program (Section 3.1) is followed by a description of the nongeometry input requirements (Section 3.2). The geometry-dependent data have been discussed in Section 2.

The organization of the MAGIC program and detailed descriptions of the subroutines are described in Section 3.3. The output of the program is also discussed. Section 3.4 will discuss the actual running of the problem on the computer, including deck configuration, operating instructions, and tape assignments. A sample problem is displayed in Section 3.5.

3.1 GENERAL DISCUSSION

The MAGIC program accepts as input a vehicle description (as described in Section 2) and an attack plane description (to be described in Section 3.2.2). The attack plane defines an area over which rays are to be "fired" toward the vehicle and the direction in which such rays are to be fired.

The output for each ray is a sequential listing of region information for each region penetrated. Each region will be identified by a region identifier as described in Section 3.2.1. A brief list of the items that are output for each region penetrated by the ray is given below:

1. Region identifier
2. Region thickness (line-of-sight distance through the region)
3. Normal thickness
4. Angle of obliquity
5. Space identification (the type of space following the region)
6. Thickness of the space.

The actual MAGIC program operates in two phases. First there is an input processing phase which reads and processes both the geometry (target description) and nongeometry (attack plane, region identification codes, etc.) data. The processing consists of validity checking of input and the

generation of data arrays to be used by the second phase.

The second phase performs the ray tracing and generates the sequential region data. The ray tracing is performed by subroutines TRACK and CALC under the control of subroutine GRID. The routines are discussed in Section 3.3.1.

3.2 NONGEOMETRY INPUT REQUIREMENTS

For conventional vehicle vulnerability, each material region (not an air region) is assigned an identification number "ID". Each air region where no material exists is assigned a "space" number. Thus each geometric region has either an "ID" number or a "space" number, but not both.

For compatibility with the ASEB vulnerability program, certain "ID" and "space" number conventions are used. All "ID" numbers are three digits and all "space" numbers are two digits. In addition, the geometric region containing both the attack plane and the vehicle must have a "space" code of 01 and no "ID". A detailed discussion appears in Section 3.2.1.

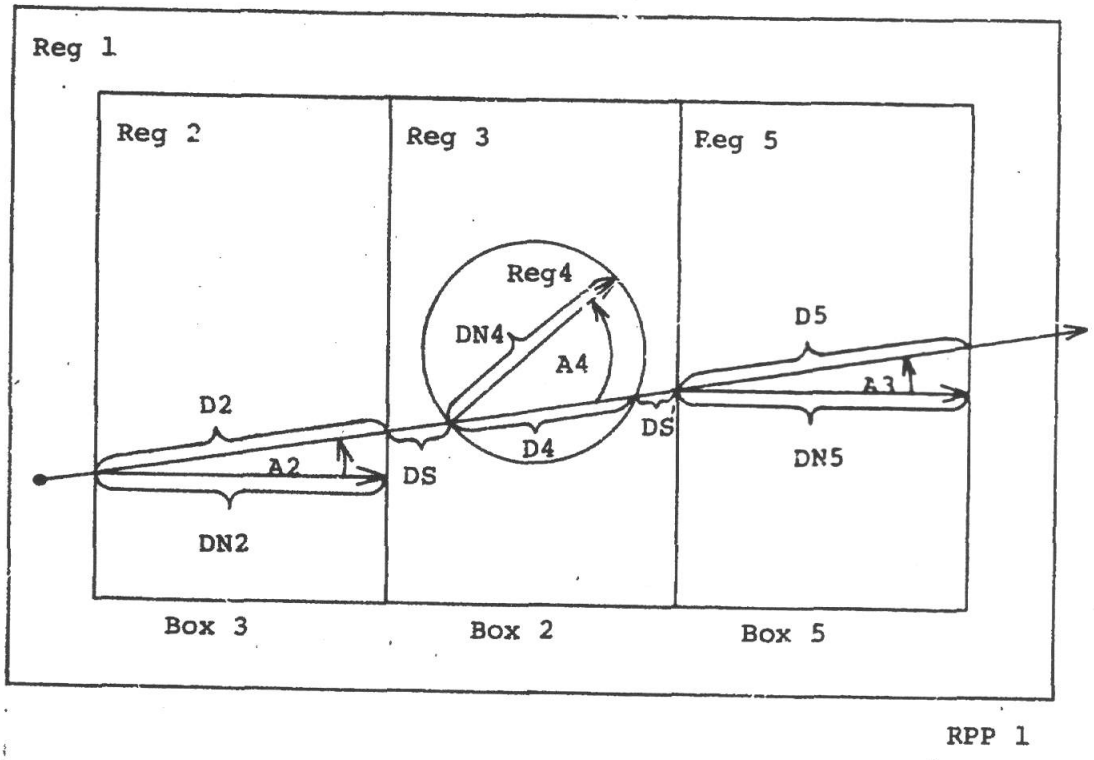
Additional input is required to establish the size and orientation of the attack plane with respect to the vehicle. A discussion appears in Section 3.2.2.

3.2.1 Identification and Space Code Data

The identification (ID) and space code data are used to identify the regions penetrated by each ray. These ID and space numbers will be punched as output together with region thicknesses and other data generated by the ray-tracing procedure. The ID and space numbers serve to identify the regions for subsequent vulnerability programs.

An example of the relationship between the original region numbers and the ID and space numbers will illustrate the usage.

Consider the following simple geometry composed of a single RPP, three boxes, and a sphere.



The region data for the preceding configuration is:

Region 1 = 1-2-3-5
Region 2 = 3-2 (Note the negation of body with a common boundary)
Region 3 = 2-3-5-4
Region 4 = 4
Region 5 = 5-2

The ID and space data table is:

Region	ID	Space
1	Blank	01
2	100	Blank
3	Blank	02
4	200	Blank
5	300	Blank

Thus the punched output would appear as:

100	The ID number of Region 2
D_2	The actual distance through Region 2
DN_2	The normal distance through Region 2
A_2	The angle of obliquity
2	The space identifier of Region 3
DS	The distance through the space up to Region 4
200	The ID number of Region 4
D_4	The actual distance through Region 4
DN_4	The normal distance through Region 4
A_4	The angle of obliquity
2	The space identifier of Region 3
DS'	The distance through Region 3 up to Region 5
300	The ID number of Region 5
D_5	The actual distance through Region 5
DN_5	The normal distance through Region 5
A_5	The angle of obliquity
9	The "9" indicates that this is the last space and that nothing else follows. The "9" is generated by the program.

The program has the capability of combining all contiguous regions having the same ID or space number. Thus, if a series of regions all having the same ID or space number are penetrated sequentially, the sum of the individual thicknesses would be output instead of the individual thicknesses. This feature is useful in dividing a region containing many bodies into several regions each composed of fewer bodies. The several regions then would have the same ID number. It is suggested that this device be employed whenever possible because it will shorten the computation time. The actual card input formats appear in Section 3.2.3.

3.2.2 Attack Plane Data

The attack plane is the plane from which the penetrating rays will emanate. The rays will be orthogonal to the attack plane. The plane is divided into a two-dimensional mesh of 4-in. cells. The following data are required to position the plane in space and to establish the number of 4-in. cells to be processed.

NX	The number of 4-in. cells in the horizontal direction.
NY	The number of 4-in. cells in the vertical direction.
IRSTART	The number of the region (an RPP) enclosing the vehicle.
IENC	Same as IRSTART.
A	An azimuthal angle.
E	An elevation angle.

ENGTH	The normal distance from the center of the vehicle to the attack plane.
Xc Yc Zc	The coordinates of the vehicle center.

Fig. 3.1 will illustrate the usage of the above items.

For the purpose of clarity the attack plane is shown as a slab in the illustration. The NX and NY define the size of the attack plane at position 1. The angle A measured from X to Y in a counterclockwise direction specifies the azimuthal angle. The angle E, as shown, specifies the elevation angle. The distance ENGTH specifies the distance, from vehicle center, of the plane. As shown in the figure the plane is first rotated in the X-Y plane, resulting in position 2, and then rotated in Z, giving position 3, the actual attack position.

Note that the region enclosing both the attack plane and the vehicle must be large enough to contain both. Note also that the attack plane must not intersect the vehicle.

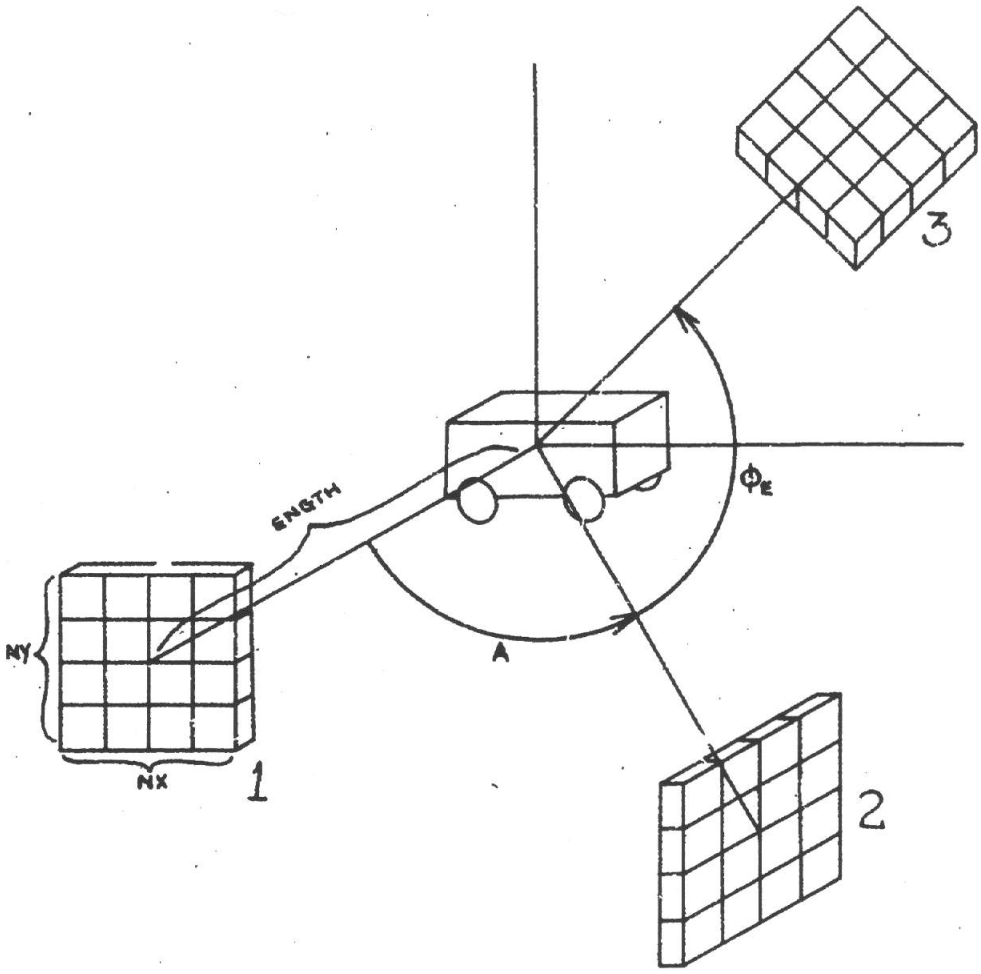


Fig. 3.1 - Attack Plane Orientation

3.2.3 Card Input Formats

A. Region Identifier and Space Code

This part of the input has exactly NRMAX cards. Each card corresponds to a region and has the region number (an integer ending in cc 10), the vehicle component identification number (ID) ending in cc 20, and the "space" number ending in cc 30.

In addition to the region number, identification number, and space code, cc 31-36 may contain alphanumeric data which will appear on the output listing only.

B. Aspect Card

This card contains the number of attack planes to be processed. It should be an integer ending in cc 5.

C. Grid Data Set Cards

Each attack plane requires a set of these cards. A set can be either two or three cards.

Card One (NX, NY, IRSTART, IENC)

NX is the number of grid squares in the horizontal direction, and NY is the number of grid squares in the vertical direction.

IRSTART and IENC are both equal to the region number of the enclosing region. The format for the four items is (4I10).

Card Two

Angle of azimuth and elevation, floating point numbers (degrees) in cc 1-12, 13-24, respectively.

ENGTH the normal distance from vehicle center to the attack plane, a floating point number in cc 24-36.

If the vehicle center is not 0, 0, 0, then cc 37-48 must contain a nonzero (any) floating point number. If the center is at zero, leave cc 37-48 blank.

When cc 37-48 is nonzero, then card three (3) is necessary. It contains the X, Y, Z coordinates of the vehicle center, punched as floating point numbers in cc 1-12, 13-24, 25-36.

3.3 ORGANIZATION OF THE MAGIC PROGRAM

The purpose of this section is to provide the reader with a basic understanding of the main logic flow in the conventional vulnerability code, MAGIC. The MAGIC routine directs the entire calculation, calling on various subprograms to perform specific functions. This section, then, is concerned primarily with the basic role of each of these subprograms in the overall scheme (see Fig. 3.2). Detailed descriptions of these routines, intended mainly for programmers wishing to implement the program on their computer, can be found in Section 3.3.1.

At the start of a problem, MAGIC first initializes certain system constants and parameters. Its next function is to store the data describing the geometry of the problem. A card is read containing either a "1" or "blank" in Column 10. The presence of a 1 informs the program that the geometry data have already been processed in a previous problem and are available on input tape 4. This tape is then read into the computer and no further geometry input data processing is performed. The code will then bypass all of the operations outlined in the next paragraph. A blank, however, instructs the code that geometry input is to be read from cards and in this case subroutine GENI is called.

GENI is the main control routine for geometry input processing. In essence, its purpose is to put the geometry data into the form required by subsequent sections of the program. Suitably processed data are stored in a large array called the MASTER-ASTER array. The processing of the body input descriptions are accomplished by first calling subroutine RPPIN to read and process the RPP data. All other bodies, with the exception of the arbitrary polyhedron (ARB), are processed directly by GENI. Subroutine ALBERT is called whenever input for an ARB is encountered. After the body input has been processed, the region descriptions are read in by GENI and stored in the MASTER-

ASTER array. This completes the processing of geometry data. Before concluding the discussion of GENI, however, mention should be made of three other important subroutines which it employs. Subroutines FLOCON and DIGCON are used to convert Hollerith input to floating point and integer formats, respectively. The SEE3 routine is used to store triplet and scalar data.

Having completed all geometry input processing, control is returned to MAGIC and a binary tape (logical 4) is written containing all of the processed geometry. This tape can be employed in subsequent problems which require no geometry changes. Having written this tape, MAGIC reads the region identifiers and space code numbers into the MASTER-ASTER array.

At this point, the ray-tracing phase of the program is initiated. A card is read informing MAGIC of the number of angles of attack (i.e., aspect angles) to be treated. All calculations for a given aspect angle are directed by subroutine GRID, and MAGIC will continue to call GRID until all angles have been completed.

GRID is the main control program of the ray-tracing segment of the code. It reads in the data for and generates an attack plane at the given angle of attack. This plane is divided into a grid of 4-in. square cells, with each cell acting as the point of origin of one ray. The tracing of

each ray from the grid through the vehicle geometry is accomplished by subroutine TRACK and GRID will, therefore, call TRACK once for each ray.

TRACK has two primary functions to perform. The first is to coordinate all the processing for a ray until it emerges from the "far side" of the vehicle. The second is to provide calculated output for each ray, for subsequent use in vulnerability analysis codes. This output consists of the "line-of-sight" thickness of each geometric region traversed by the ray, the obliquity (or angle of incidence) of the ray with respect to each region (excluding "spaces") encountered, and the normal or perpendicular distance through each region (excluding "spaces").

Tracing of a ray is initiated by a call to subroutine G1. This routine computes the line-of-sight distance from any point in a given region to the outer surface of that region. To accomplish this, the individual bodies which make up the region in question are obtained from the region descriptions in the MASTER-ASTER array. Nine separate body routines are available to compute the distance through each body in the path of the ray. This process continues until the ray emerges from the region. The next region in the path of the ray is then determined by retrieving a list of possible regions (created by GENI) from the MASTER-ASTER array. Subroutine WOWI is then called to determine which

of the possible regions actually has been entered by the ray. The region ID number of this new region is now compared to that of the region just left. If they are identical, then the two regions are part of the same vehicle component. The ray is then traced through the new region, with line-of-sight distances being accumulated. When a new region identifier is encountered, indicating that the ray is leaving a vehicle component, control is returned to TRACK. The total line-of-sight distance through the component is packed and stored, and G1 is called to continue tracing the ray. This process is repeated until the ray finally emerges from the vehicle. The net result of these calculations is the total line-of-sight distances through vehicle components traversed by the ray.

TRACK now calls subroutine CALC which computes the normal distance through each component and the angle of obliquity. The calculated data for the ray are then printed out and control is returned to GRID, where the starting point of the next ray is established. After all rays at a given aspect angle have been traced, GRID returns to MAGIC. The entire ray-tracing sequence is then started again for the next aspect angle and continues until all angles have been completed. The problem ends at that point.

During the ray-tracing procedure, several other subroutines are employed to perform the following specialized functions.

OPENK	Unpacks the component line-of-sight distances.
DIREC	Computes the direction cosines of the normal to a given ray.
XDIST	Computes the distance between two points.
TROPIC	Picks random direction cosines.
RANF	Computes random numbers.
SETUP	Converts integral and decimal parts of a floating point number to two fixed point integers.
ISIGN	Determines a Hollerith sign.
UN3	Unpacks integer variables in the MASTER-ASTER array.
RPP2	When a ray goes from one RPP into another, this routine computes the number of the new RPP.
DCOSP	Computes direction cosines of the line between two points.

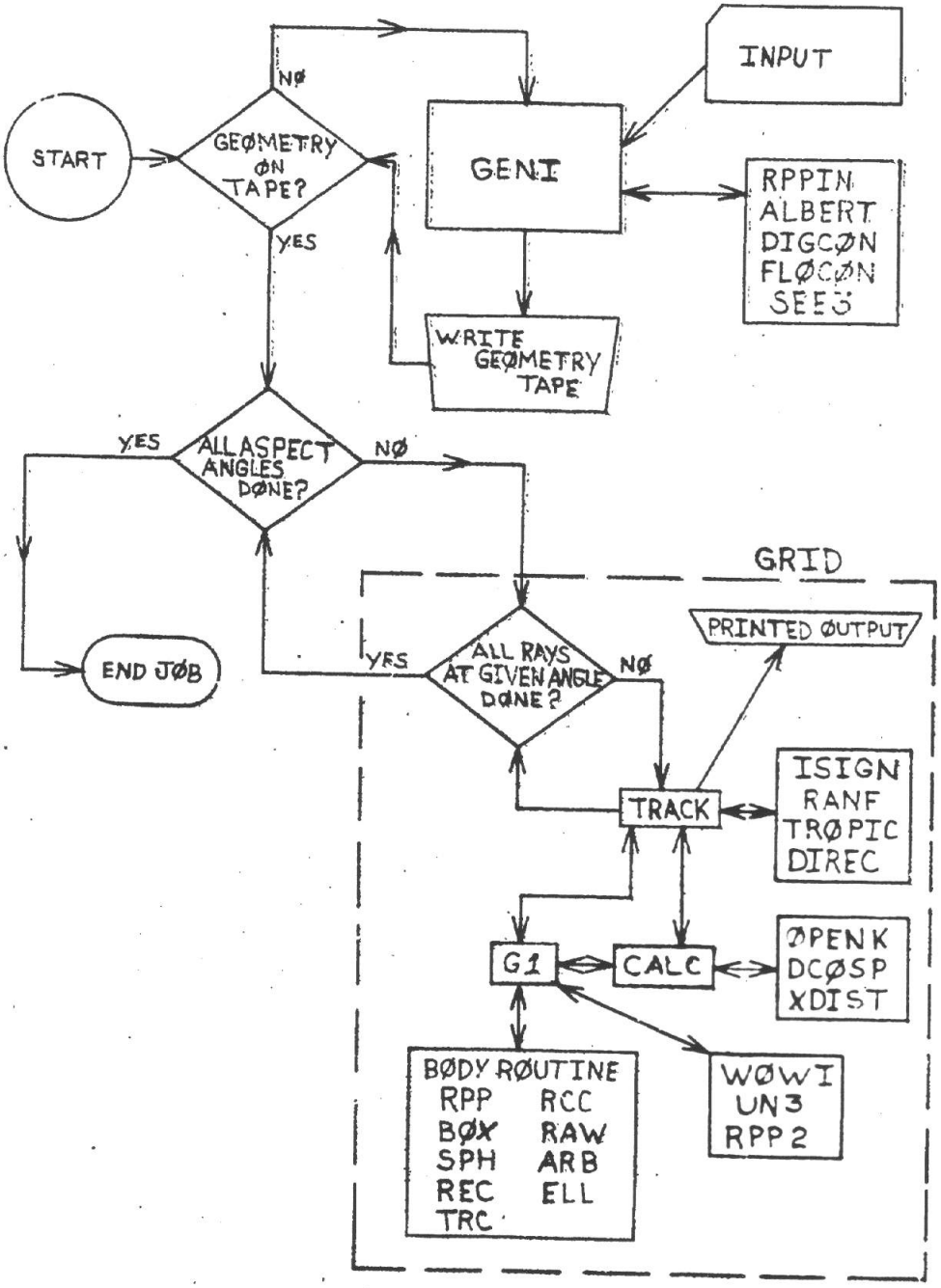


Fig. 3.2 - Magic Flow Chart

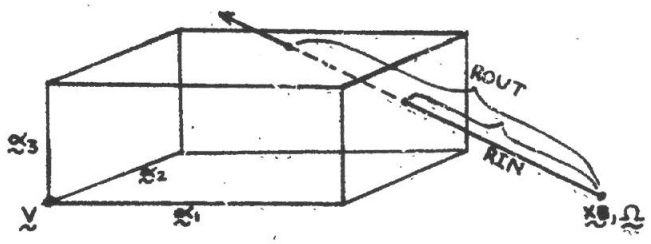
3.3.1 Description of Routines

The following pages contain detailed descriptions of the individual routines that comprise the MAGIC program. The routine descriptions are given in alphabetical order.

Subroutine BOX

This routine will calculate the distance from a fixed point $\underset{\sim}{XB}$ in the direction $\underset{\sim}{WB}$ to the first and last contacts with a "BOX".

The box is defined by giving its vertex $\underset{\sim}{V}$ and three length vectors $\underset{\sim}{\alpha}_1, \underset{\sim}{\alpha}_2, \underset{\sim}{\alpha}_3$. The length vectors denote the lengths and directions of the box edges.



The ray $\underset{\sim}{XB} + r\underset{\sim}{\Omega}$ will intersect the box if

$$0 \le \left(\underset{\sim}{XB} - \underset{\sim}{V} + r\underset{\sim}{\Omega}\right) \cdot \underset{\sim}{\alpha}_i \le \underset{\sim}{\alpha}_i \cdot \underset{\sim}{\alpha}_i \qquad (1)$$

for some value of r.

Let

$$a_i = \sqrt{\underset{\sim}{\alpha}_i \cdot \underset{\sim}{\alpha}_i}$$
$$p_i = \underset{\sim}{XB} \cdot \underset{\sim}{\alpha}_i / a_i$$
$$V_i = \underset{\sim}{V} \cdot \underset{\sim}{\alpha}_i / a_i$$
$$\Omega_i = \underset{\sim}{\Omega} \cdot \underset{\sim}{\alpha}_i / a_i \qquad (2)$$

Thus equation (1) becomes

$$0 \le p_i - V_i + r\Omega_i \le a_i$$

defining

$$H(X) = 1 \quad \text{for } X>0$$

$$H(X) = 0 \qquad X<0$$

and equation (2) becomes

$$\frac{v_i - p_i + a_i H(-\Omega_i)}{\Omega_i} \le r \le \frac{v_i - p_i + a_i H(\Omega_i)}{\Omega_i} \tag{3}$$

or

$$\xi_i^-(v,p,a) \le r \le \xi_i^+(v,p,a) . \tag{4}$$

Thus, the ray XB + rΩ intersects the box if and only if there exists an 'r' satisfying inequality (4).

This 'r' must be nonnegative since only travel in the direction Ω from XB is of interest. Thus we define

$$\text{ROUT} = \min \xi_i^+$$

$$\text{RIN} = \max \xi_i^-$$

where RIN is the distance from XB to first contact and ROUT is the distance from XB to last contact.

The ray XB + rΩ intersects the box if and only if ROUT $\ge$ 0 and RIN $\le$ ROUT. The above statement implies that $\xi_i^+ > 0$ for all i in order for the ray to intersect the box.

Subroutine CALC

Subroutine TRACK will call this routine to perform normal distance and angle of obliquity calculations. The TRACK routine will call CALC at each region intersection. CALC will determine if normal distance and angle of obliquity are required by examining the "space" code input.

The calculations are required for material regions and not for "space" regions. The CALC routine also determines if the ray has left the RPP containing the vehicle and assigns a "9" to the last space.

The routine contains separate coding sections for the normal distance and angle of obliquity calculations. Each section refers to a distinct body type (sphere, cylinder, etc.).

The following paragraphs contain a microscopic discussion of the calculation procedure and should be read in conjunction with a FORTRAN listing of the subroutine.

CALC begins by a call to subroutine OPENK to retrieve the pertinent information about this contact as stored by subroutine TRACK.

The value of NIR is tested; if it is negative the calculation is finished.

If NIR is positive the calculation begins.

SIT is always the line-of-sight distance for the present contact and its value is retrieved from the TR array where it was put by TRACK.

The DO Loop to Statement 10 updates the point of contact from its original position XS to its position of contact with this region. WS are the direction cosines of the ray, and TRAVEL always contains the total line-of-sight distance. Having calculated XI, the contact point, the

value of TRAVEL is updated by the line-of-sight distance through this region.

Seven (7) is subtracted from the value of LSURF as unpacked by OPENK since, to avoid negative packed numbers, 7 was added to LSURF when it was packed in TRACK.

A test to determine the sign of LSURF is made. If the value is negative, the ray is leaving the surface; the direction cosines of the computed normal must be reversed in sign. The variable "XNOS" is a constant multiplier of the direction cosines and will be set to ±1 depending on the sign of LSURF.

Statement 16

LOC is set to the location of the descriptive data for body number NBO.

A call to UN3 is made to find the body type "ITYPE" which will be one of the basic figures.

If sense switch 6 is on, the routine will output the values of L, XI, WS, TRAVEL, LSURF, NBO, ITYPE, and LDATA for debugging purposes.

By setting LOC = LOC+1 and calling the unpacking routine, the location of the physical data for the body we are working on is retrieved.

Preparing to enter a computed "go to", ITYPE is set to ITYPE+1 to avoid a zero index.

In future calculations the surface number must be positive so it is set to its absolute value.

A test for a valid ITYPE (1-9) is made, and an error print occurs if the value is outside these limits.

Statement 18

The computed "go to" transfers control to a routine which will find the normal to each type of body. (The REC and RCC have the same beginning processing, and therefore, share part of one routine, as do BOX and RAW.)

Statement 50

This section computes normals to any of the six surfaces of an RPP.

The sign of XNOS is changed if LSURF = 1, 3, 5. The variable I is set for LSURF = 1, 2; and I = 2 for LSURF = 3, 4; and I = 3 for LSURF = 5, 6.

Statement 60

This section computes normals for the RPP.

Statement 100

The normal to any of the six surfaces to a box is computed by this coding.

LSURF is tested for odd or even. This is done by "adding" a Boolean "1" to LSURF and comparing with zero. If LSURF is even, the sign of XNOS is changed.

Statement 102

Check LSURF-3. If negative LSURF is 1 or 2, set I = 1. If LSURF is 3, set I = 2. If LSURF>3, test LSURF-5. Negative now means set I = 2; $\geq$ 0 means set I = 3.

The length vector I = 1, 2 or 3 is now normal to the surface of contact. The odd-even check indicated whether the contact surface was front or back, and the sign of XNOS was changed to reflect this.

Statement 115

A call to UN3 to retrieve the location of the vertex, and three height vectors (in that order). IEMP is a temporary storage array.

Set LH+IEMP(I) the location of the data for the height vector of interest.

LV+IEMP(4) the location of the vertex data.

The DO 117 Loop transfers the vertex and height vector coordinates from the main storage (ASTER) array to two temporary arrays, TEMP and TEMP1, respectively.

A call to DCOSP (TEMP1, TEMP) computes the direction cosines from TEMP1 to TEMP and stores the answers in WB.

The DO 120 Loop orients WB (direction cosines of normal) according to the sign of XNOS.

Control is now sent to Statement 1000.

Statement 150

This is the section where the normal to the sphere is calculated.

A call to UN3 picks out the location of the vertex. The DO 115 Loop takes the vertex coordinates from the ASTER array and puts it in the temporary array TEM.

A call to DCOSP (XI, TEM) computes the direction cosines from XI to TEM and stores the results in WB. These are direction cosines of the normal.

DO 156 Loop sets the correct sign to the WB direction cosines.

Control is now transferred to Statement 1000.

Statment 200

This section of the code computes normals to the RCC.

Test LSURF-2. If positive, transfer to Statement 210. If zero, change sign of XNOS. If negative, no change is made.

For LSURF = 1 or 2 we have a hit on a planar surface, that is the top or bottom of the cylinder.

A call to UN3 retrieves the location of the vertex data, as well as the height vector.

The DO 213 Loop transfers the vertex and height vectors from main memory ASTER array to two temporary arrays "TEM" and "TEM1".

A call to DCOSP (TEM, TEM1) computes the direction cosines and puts them in SB.

The DO 204 Loop, as usual, causes the direction cosines to reflect the sign (orientation) of XNOS.

Statement 210

The code transfers here for an LSURF = 3 (spherical or elliptical side) and ITYPE = 4 or 5.

A call to UN3 retrieves the location of the vertex, height and first radius data.

The DO 212 Loop takes the vertex and height vectors from main memory and places them in temporary arrays, TEM and TEM1, respectively.

DCOSP is called twice to compute direction cosines, "WN" from TEM to XI (vertex to point of contact on surface) and "WI" those of the height vector TEM to TEM1.

In preparation for taking a dot product, set SUM = 0.

The DO 217 Loop computes the dot product of WN · WI.

The DO 214 Loop translates the point of contact to a point on the height vector XP.

SUM is now the cosine (dot product of the direction cosines) of the angle formed by the height vector and the line from TEM to XI.

SUM * XDIST (TEM, XI) (where XDIST is a function to calculate the distance from TEM, XI) represents that portion of the height vector's total length taken up by XI under translation.

Finally, XP = XP* WI + TEM sets the value of XP to be just the translated line desired.

If ITYPE-5 is 0, the body is a REC and control is transferred to Statement 250. Otherwise control is transferred to Statement 216.

Statement 216

The direction cosines from XI to XP are those of the normal and are put into WB.

The DO 215 is a loop to orientate the WB's to the sign of XNOS.

Statement 250

Control was transferred from the cylinder routine when ITYPE = 5.

The DO 253 Loop takes the two radius vectors and places them at XP, the translated point on the height vector. The resultant vectors are called TEMP and TEMP1, respectively.

A1 is the distance from XP to TEMP.

A2 is the distance from XP to TEMP1.

Test (A1, A2) and set the larger of the two to A1. Change the value of TEMP to be the one farthest from XP if it is not.

Statement 255

$C = \sqrt{A1^2 - A2^2}$ (the distance from the center to the focus)

Compute the direction cosines from XP to TEMP and call them WN.

The DO 260 Loop computes the coordinates of the two foci and calls them TEM and TEM1, respectively.

Now the direction cosines from TEM (one foci) to XP (point of contact) is computed and called WN.

The DO 263 Loop sets TEM to the end point of a line from one of the foci through XI with an overall length of 2A1.

The direction cosines from this new end point TEM to the other focis TEM1 are those of the normal WB.

The DO 270 Loop, as usual, reorients the WB's to the sign of XNOS.

Control is transferred to Statement 1000.

Statement 300

This section computes normals to a truncated cone.

If LSURF = 3 (not top or bottom), unpack the vertex, height, and two radius pointers. (These are numbers which index the position of the floating data in main storage.)

Compute the difference (DIF) in the size of radius (1) top and radius (2) bottom.

Switch LR1 and LR2 so that the LR1 is the largest and the sign of DIF is positive.

The DO 305 Loop sets TEMP1 = vertex.

TEMP = the top of the "extended height vector" (i.e., H.V. multiplied by FACTR).

Compute TDIS distance from XI to TEMP

QDIS distance from TEMP1 to TEMP

WN direction cos from TEMP to XI

WA direction cos from TEMP to TEMP1

The DO 310 Loop takes the dot product of WN·WA and stores the result in the variable SUM. QSUM = TDIS/SUM is that portion of the height vector used by the triangle made from XP→XI-TEMP.

QPLS = QDIS-QSUM is the remainder of the height vector.

Now the DO 312 Loop computes TEMP, the point on the height vector normal to the point XI on the side surface.

A call to DCOSP computes the direction cosines WB from XI to TEMP of the normal.

The DO 313 Loop, as usual, changes WB to reflect the sign of WB and control is sent to Statement 1000.

Statement 315

Change the sign of XNOS so that top and bottom can be treated the same.

Unpack via UN3 the location of the vertex and height vectors. Then by the DO 320 Loop, TEMP = vertex coordinates; TEMP1 = height vector coordinates.

Compute WB the direction cosines from TEMP to TEMP1. (The normal to top or bottom.)

The DO 321 Loop again changes WB to reflect the sign of XNOS and control goes to Statement 1000.

Statement 350

This section is for computing normals to the ellipse. First, retrieve the location via UN3 of the two foci (LR1, LR2), and the scalar distance of the major axis (LS).

The DO 352 Loop sets TEM = focus 1; TEM1 = focus 2.

A = Scalar distance for major axis.

Compute direction cosines WN from one focus (TEM) to XI (contact point).

The DO 353 Loop is to find the end point of the line of length A and direction cosine WN which starts at focus TEM. Call its other end TEM also.

Now the direction cosine from TEM to TEM1 is called WB and is the normal.

The DO 354 Loop, as usual, sets WB to the sign of XNOS.

Transfer of control is made to Statement 1000.

Statement 400

This section is to choose the normals to the five sides of a RAW. But since four of the five surfaces are the same as those of a box, the routine will go to 100 (BOX) for surfaces 1, 3, 5, and 6.

LSURF = 2 represents the slant surface and LSURF = 4 is an error condition which goes to Statement 415.

Statement 406

A call to UN3 to unpack pointers to data on vertex (LV) and height vectors LV1, LV2.

Statement 450

Computation of normals to the ARB polyhedron.

The equation of the plane is AX+BY+CZ+D = 0. The direction numbers of the normal are A, B, and C.

LSPT is the place in the ASTER array where A, B, and C are stored.

The DO 453 Loop computes $A^2+B^2+C^2$.

DIV = $A^2+B^2+C^2$

The DO 460 Loop computes direction cosines and multiplies by "XNOS". Transfer of control is made to Statement 1000.

Statement 415

Error halt, for a bad LSURF number in TRC (LSURF = 4).

Statement 1000

At this point the normal direction cosines have been computed.

The DO 1001 Loop moves the contact point very slightly into the region in order to avoid tracking from a boundary.

The angle of obliquity in degrees is now arccosine of $\frac{WB.WS.180}{\pi}$.

If the body is an ARB, the angle must be made $\leq 90^\circ$ by changing the sign of WB. (Note: This was done in all other figures by implicitly testing LSURF odd or even.)

Statement 1004

NASC = -2. Set cumulative dist = 0. Starting a new ray.

IR = NIR

Call G1 to shoot the ray from XB in the WB direction.

SN = S1 The normal distance traveled by the ray.

GO TO 20.

Statement 20

Call OPENK to get the region number (NTYPE).

ISPOT = LIRFO+NTYPE-1

ISPOT is the location in MASTER of region and space code data for NTYPE.

IDENT = IDENT-1 because one was added when it was packed in MAGIC.

If sense switch 6 is on, debug printout occurs with the variables NIR (region identifier), and NTYPE (region number).

SIN	Line-of-sight distance
ANGLE	Angle of obliquity
SSPACE	Distance through a space
SN	Normal distance
WB	Direction cosines of normal.

Statement 24

ISPOT and the call to UN3 change NIR, the region number, from internal code to identifier for printing by TRACK.

If IDENT is >9, there is no space following the region and NTYPE (type of space) and SSPACE (distance through space) = 0.

If IDENT = 0, it is also not a space (space codes are -1, 1, 2, 3, ... 9 only).

Statement 40

Test for last contact. Space "IDENT" code 9 (last space).

NTYPE = IDENT (type of space).

SSPACE = TR (L+1) - next line-of-sight distance.

TRAVEL = TRAVEL+SSPACE update distance counter.

RETURN.

Subroutine DCOSP (XA, XB, WA)

Purpose of Routine

To compute the direction cosines from point XA to point XB and to store the direction cosines in the array WA.

Description of Routine

The variables XA, XB, WA are all singly subscripted arrays of length 3. The quantities XA(1), XA(2), XA(3) are the X, Y, Z coordinates, respectively, of the point XA. The WA(1), WA(2), WA(3) are the WX, WY, WZ direction cosines to be computed.

Let DIS be the distance between point XA and point XB. The direction cosines are then

WA(I) = {XB(I) - XA(I)} /DIS for I = 1, 2, 3.

Subroutine DIGCON (IC, IX, LE)

This routine is used by GENI to convert the body data read in Hollerith format into integer format for use by GENI if triplet/scalar data was used for body descriptions.

The IC array contains the Hollerith data which were read in by GENI. The IX array is to hold the resultant integer, and LE is one more than the number of integers to be created.

Blanks are ignored in each integer field, except that a field of all blanks will be converted to zero. The valid characters are the numericals 0-9. Any other character, when detected, causes an error message to be output.

At the completion of the routine, the IX array will contain LE-1 entries each of which is an integer from the field. Thus, IX(1) will be the integer from card columns 11 through 20, and IX(2) will be from card columns 21 through 30, and so on up to IX(LE-1).

Subroutine FLOCON (IC, FX, LE)

This subroutine is used by GENI to convert a body data card read in Hollerith format into floating point numbers.

IC is the data array which contains, in positions IC(11) through IC(70), the Hollerith data read in by GENI. FX is an array to contain the floating point numbers which result from the FLOCON calculations. There will be LE-1 such floating point numbers created.

The routine ignores blank character and accepts in addition to the numerals 0 through 9 the characters (-), (+), (E), (.). Any other character causes an error printout. Omission of a decimal point or failure to complete an exponent also will cause an error printout. Each field is 10 characters long and starts at IC(11),IC(2),IC(31), etc., and ends at IC(20), IC(30), etc. The program will translate the Hollerith information to floating point and store the results in FX(1), FX(2), up to FX(LE-1) for a maximum LE of seven.

Subroutine G1 (S1, IR, XP)

Subroutine G1 is the main ray-tracing routine of the MAGIC program. It performs the following function. Given a ray in region IR at point XB with direction cosines WB, find the distance to the next region and the number of that region (IRPRIM).

The following description outlines the method used to accomplish this task. The flow chart provided at the end of this discussion is a replica of the coding in G1.

Assume that a new ray has been started at its point of origin XB_o in region IR. The variable NASC is set to -1 by the calling program, indicating that a new ray is going to be processed. Let us define a variable DIST as the total distance traveled by the ray since it left XB_o. Thus, DIST is initialized to zero for a new ray. G1 first investigates the region description for IR, obtaining all the bodies which the ray might strike. Then, for each body mentioned in the description, the distances from XB_o to the point where the ray enters the body (RIN) and to the point where the ray leaves the body (ROUT) are computed. Nine different body routines, each applicable to a different body type, are available to G1 for this purpose. The value of ITYPE for that body, as obtained from the MASTER array, determines which routine is called (i.e., if the body being examined is a sphere, ITYPE = 2 and G1 will call subroutine SPH to compute RIN and ROUT for a sphere). G1 then scans all of the RIN and ROUT values and selects the smallest of these distances, subject to the constraint that the distance chosen be greater than DIST. The body number corresponding to the chosen distance is then placed in the NASCT table. In general, the ray will not hit all of the bodies mentioned in the region description. If so, ROUT is set equal to a constant PINF (equal to -10^{50}).

Three possibilities exist at this point.

1. A unique value of RIN was selected, in which case the NASCT table contains only one body number. This is the next body in the path of the ray.
2. The selected value of RIN may correspond to more than one body, indicating that two or more bodies have a common surface. Each of these body numbers then will be stored in NASCT.
3. The value of ROUT for the body in which the ray is currently in may be selected. This means that the ray will leave this body before encountering any other.

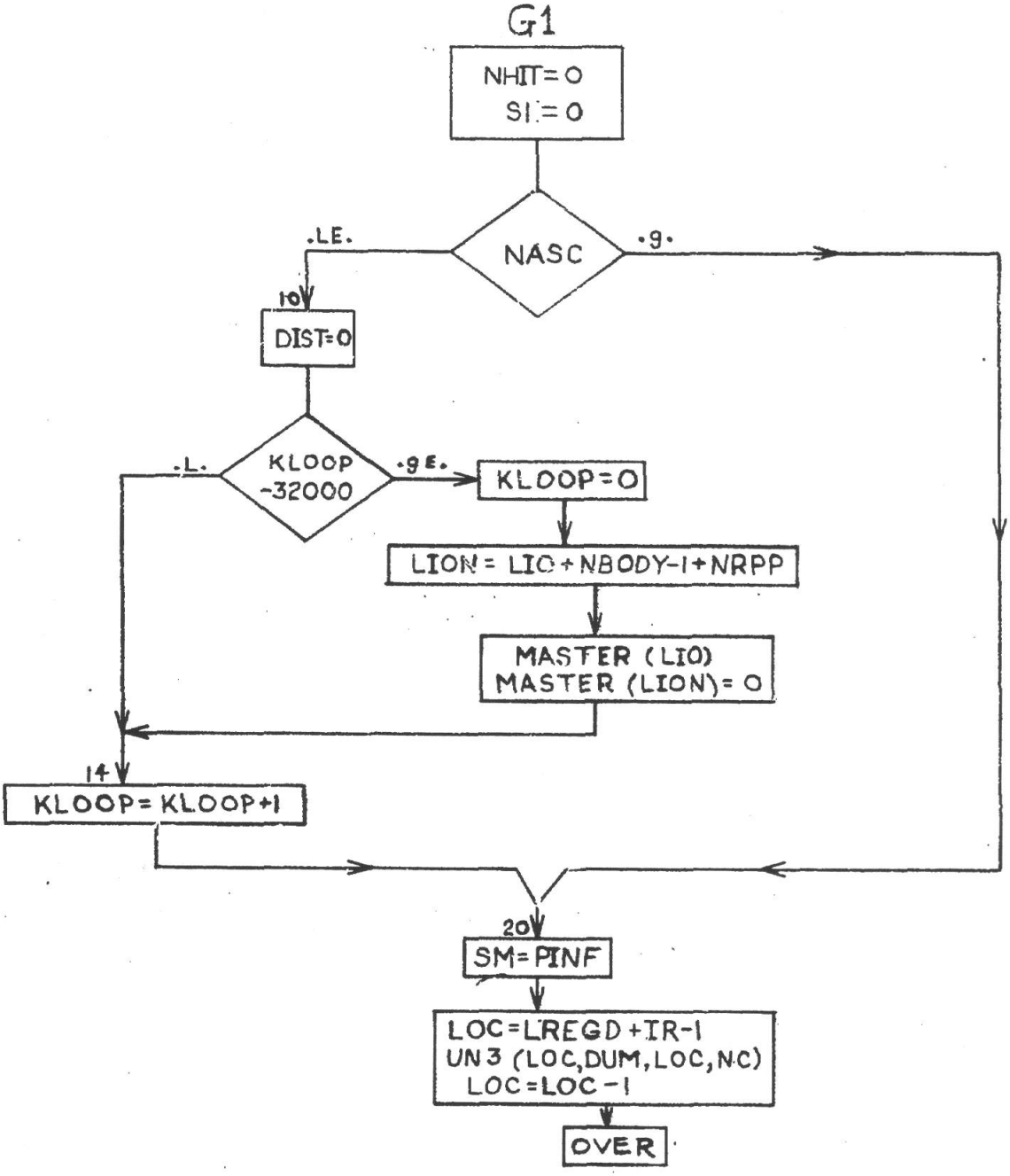
In any event, the ray is advanced to the next body surface (at point XP) and DIST is equated to the chosen value of RIN or ROUT.

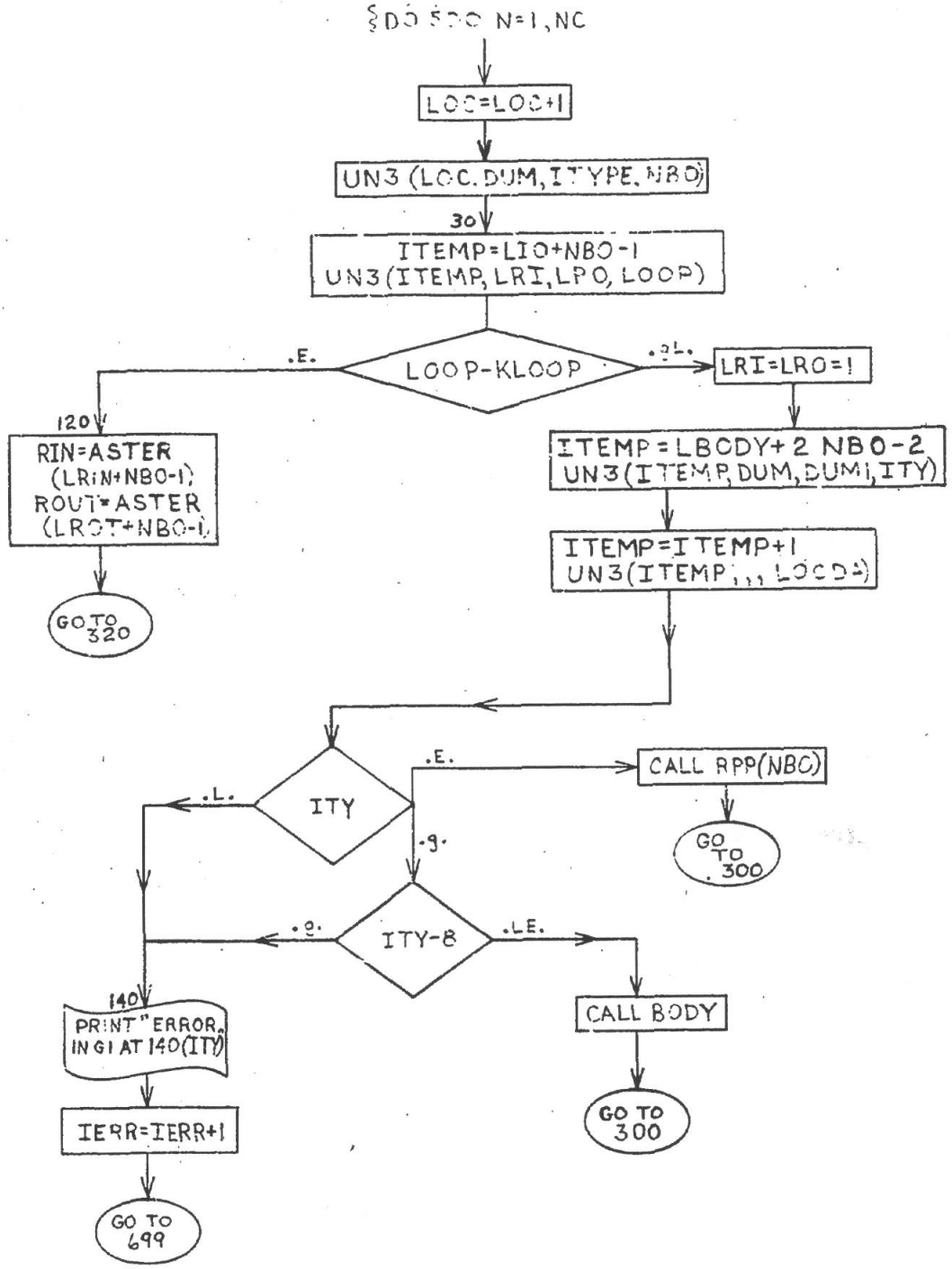
G1 must now determine the next region (IRPRIM) where the ray, at point XP, is located. The first body number is retrieved from NASCT and its entering (if RIN was selected) or leaving (if ROUT was selected) table is examined (see discussion of WOWI for explanation of these tables). These tables provide all the possible regions which may contain the current position of the ray. The variable JREG is set equal to the first region in the table and subroutine WOWI

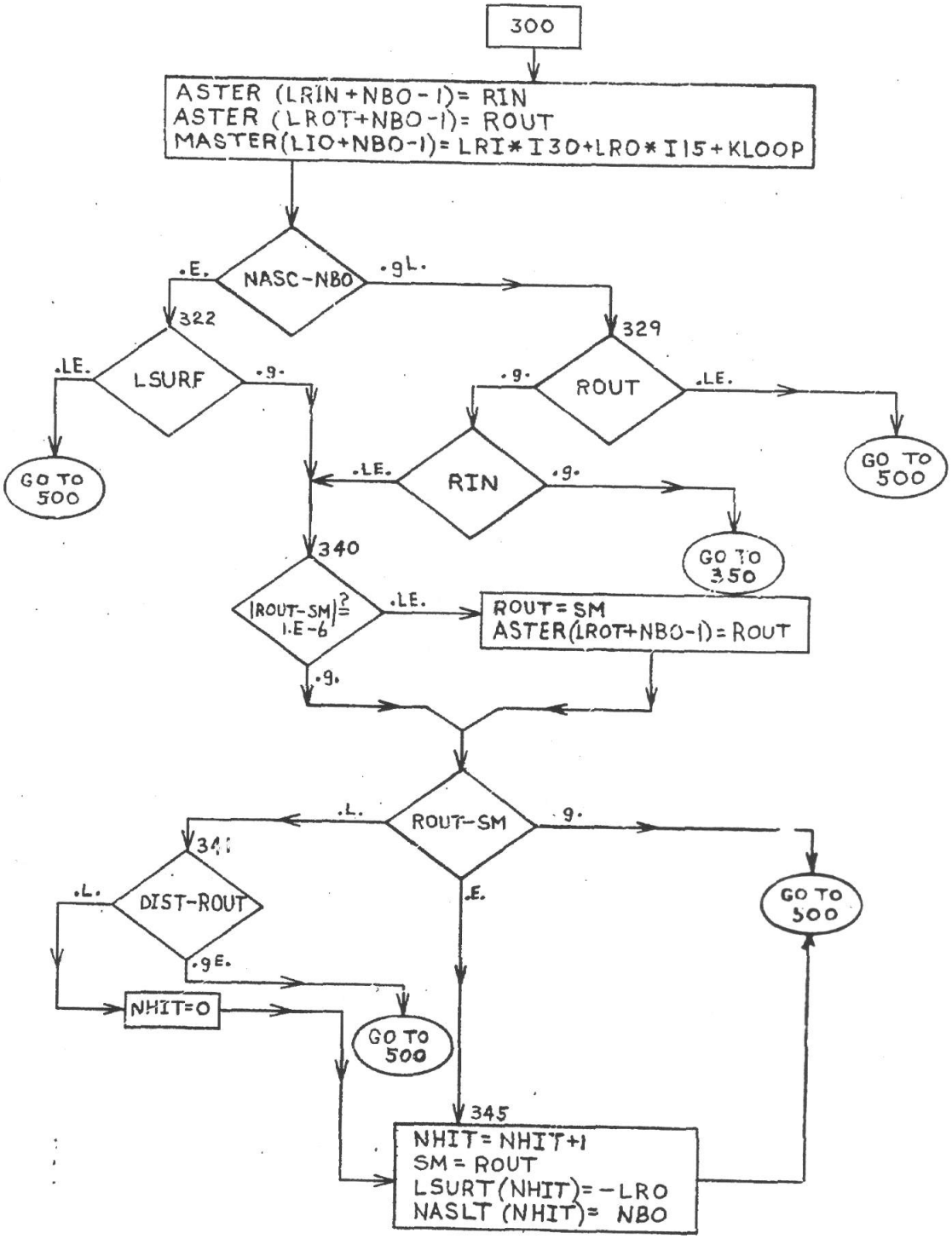
is called. WOWI will test whether the ray is in JREG and will set LTRUE = 1 if it is, or LTRUE = 0 if it is not. If LTRUE = 0, G1 will equate JREG to the next possible region and call WOWI again. This process continues until the correct region (LTRUE = 1) is found. If the ray cannot be located in any of the regions in the entering or leaving table, the next body is taken from the NASCT table and a new list of possible regions is compiled. If the NASCT table is exhausted without locating a region, an error has occurred and the error counter IERR is incremented by 1. The ray is terminated and G1 returns to its calling program with IRPRIM set to zero. The entire job is terminated if IERR exceeds 10. Assuming, however, that the correct region IRPRIM has been determined, the value of IR is compared to IRPRIM.* If they are the same, G1 remains in control repeating the above procedure with a new value of DIST. The cumulative distance, S1, traveled in a region is updated throughout the tracking. If, however, IR $\neq$ IRPRIM, a new region has been entered and G1 returns to TRACK, where the total distance through the region just left is stored. IR

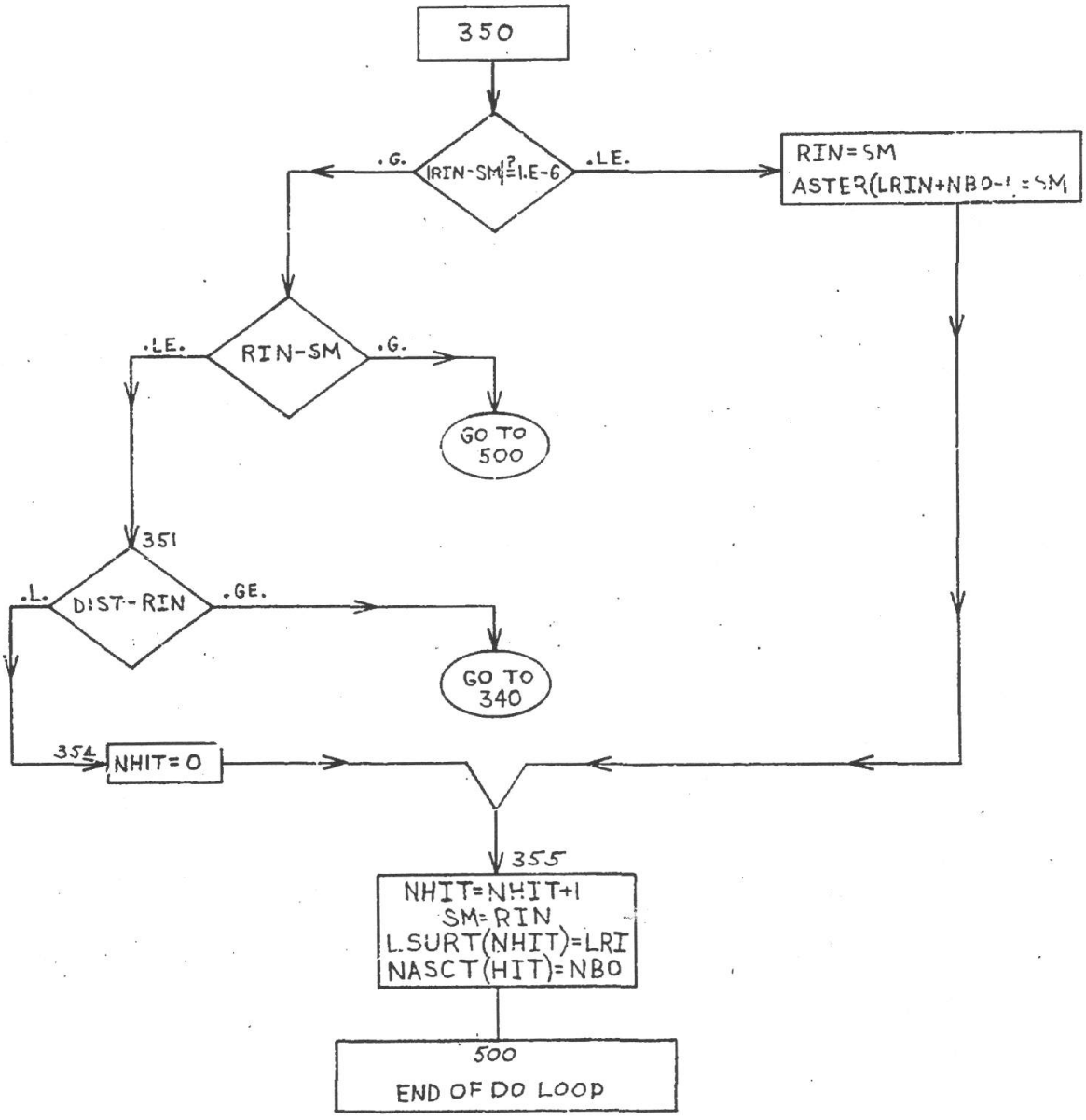
---

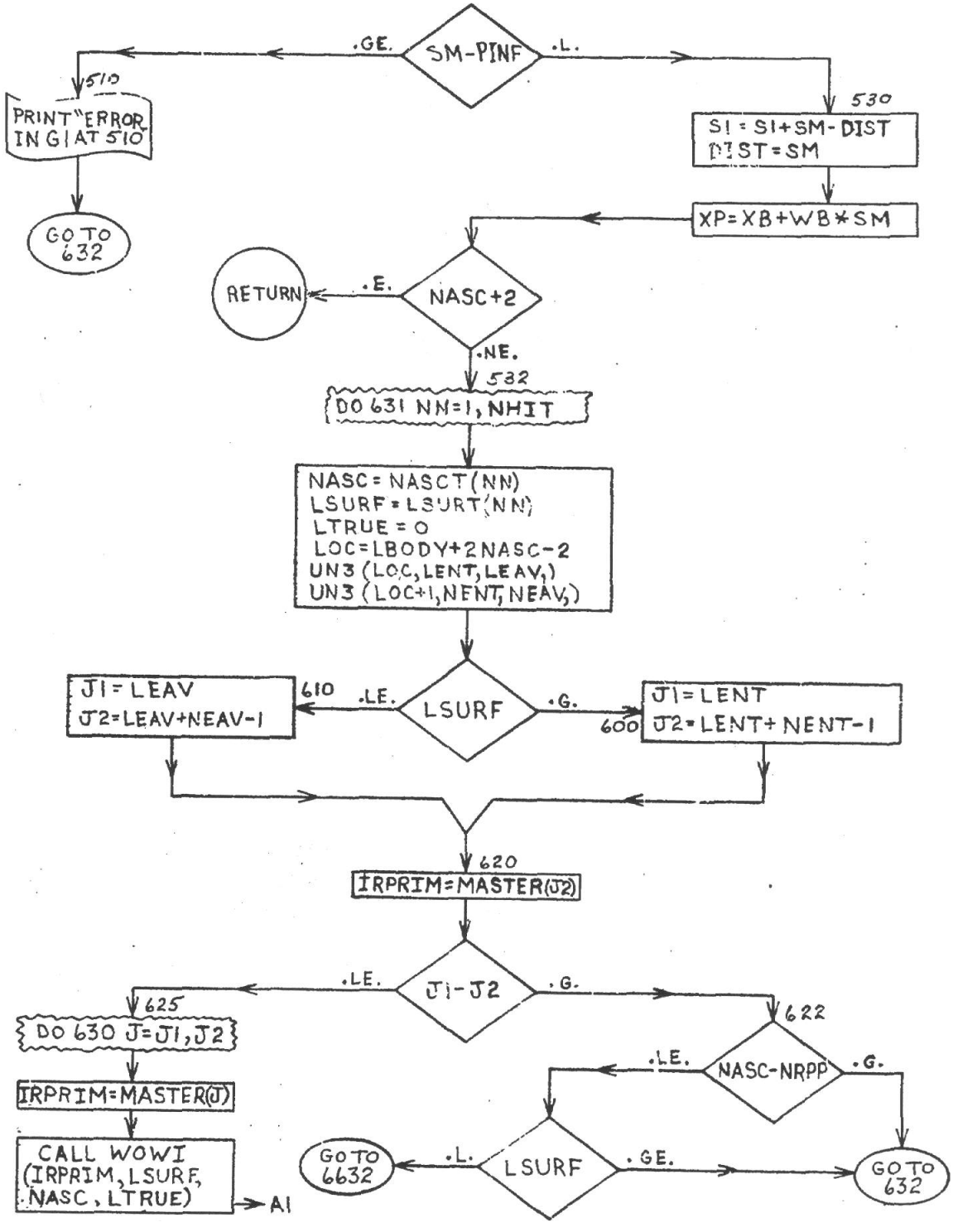
* The conventional and nuclear versions of G1 diverge at this point. In SAM-C, the work of G1 is finished as soon as IRPRIM is determined. SAM-C does not contain an error counter and the job terminates when IRPRIM = 0. The remainder of this discussion, then, applies only to the MAGIC version of G1.

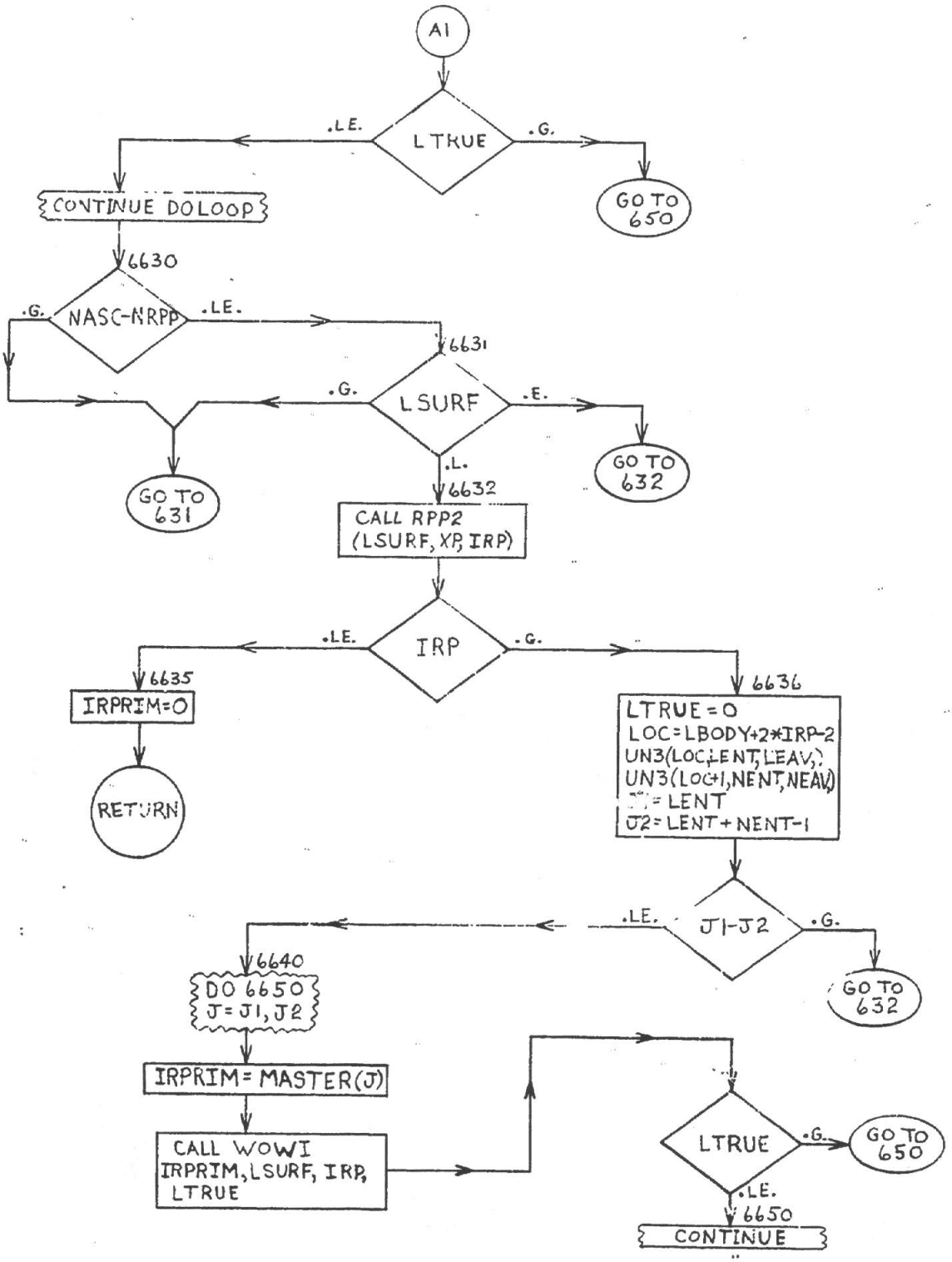

G1
NHIT=0
SI=0
NASC
.LE.
.9.
10
DIST=0
KLOOP -32000
.L.
.9E.
KLOOP=0
LION = LIO+NBODY-1+NRPP
MASTER (LIO)
MASTER (LION)=0
14
KLOOP=KLOOP+1
20
SM=PINF
LOC=LREGD+IR-1
UN3 (LOC,DUM,LOC,NC)
LOC=LOC-1
OVER

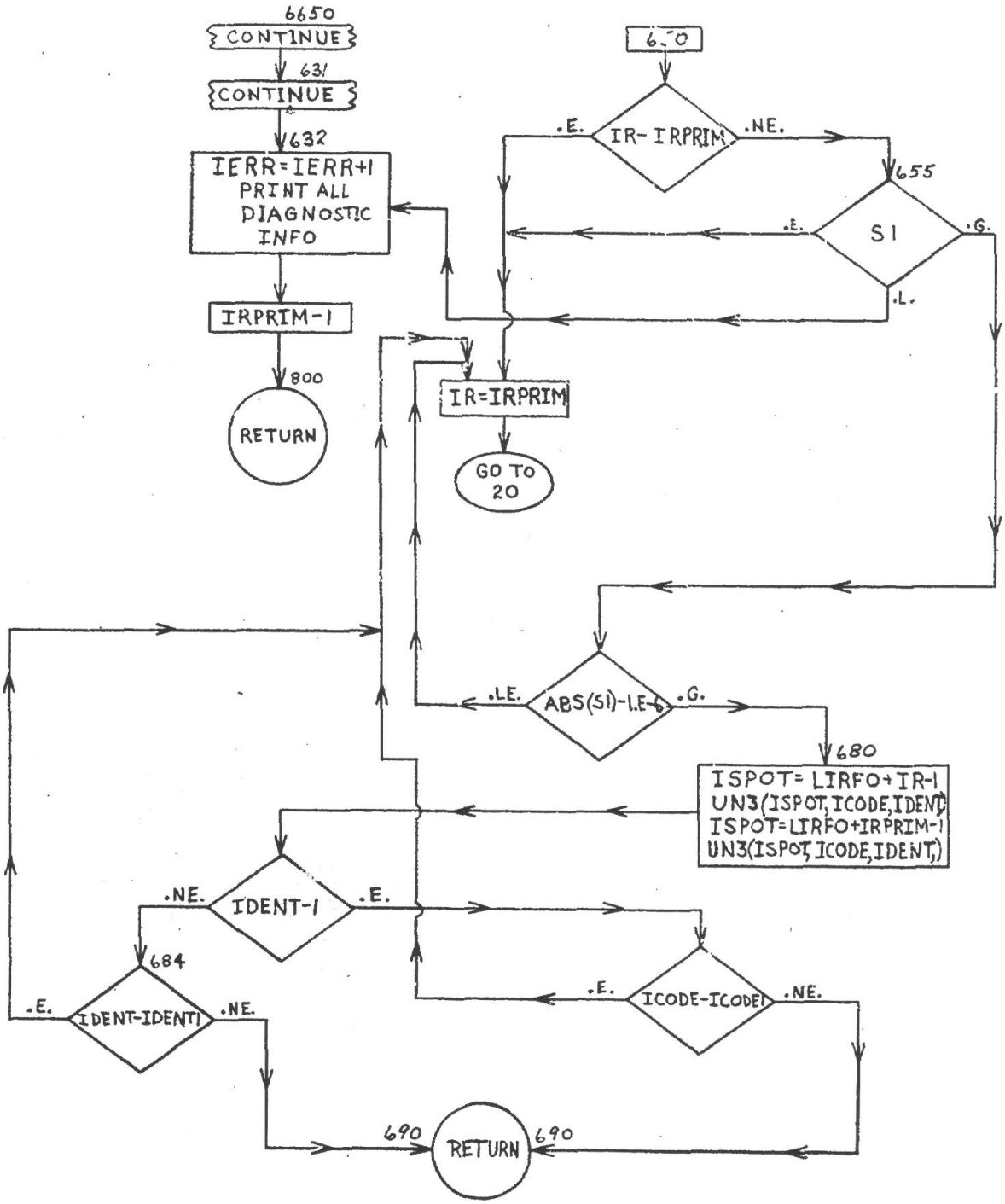

§DO 500 N=1,NC
LOC=LOC+1
UN3 (LOC.DUM,ITYPE,NBO)
30
ITEMP=LIO+NBO-1
UN3(ITEMP,LRI,LPO,LOOP)
LOOP-KLOOP
.E.
.GT.
LRI=LRO=1
120
RIN=ASTER
(LRIN+NBO-1)
ROUT=ASTER
(LROT+NBO-1)
GO TO 320
ITEMP=LBODY+2 NBO-2
UN3(ITEMP,DUM,DUM1,ITY)
ITEMP=ITEMP+1
UN3(ITEMP,,,LOCDA)
ITY
.E.
CALL RPP(NBO)
GO TO 300
.L.
.G.
ITY-8
.LT.
.E.
CALL BODY
GO TO 300
140
PRINT "ERROR IN G1 AT 140(ITY)
IERR=IERR+1
GO TO 699


300
ASTER (LRIN+NBO-1)= RIN
ASTER (LROT+NBO-1)= ROUT
MASTER(LIO+NBO-1)= LRI*I30+LRO*I15+KLOOP
NASC-NBO
.E.
.GT.
322
329
LSURF
.LE.
.GT.
ROUT
.GT.
.LE.
GO TO 500
GO TO 500
RIN
.LE.
.GT.
GO TO 350
340
|ROUT-SM|/1.E-6 ≤ 1
.LE.
ROUT = SM
ASTER(LROT+NBO-1) = ROUT
.GT.
ROUT-SM
.L.
.GT.
341
DIST-ROUT
.L.
.GE.
GO TO 500
NHIT=0
GO TO 500
.E.
345
NHIT = NHIT+1
SM = ROUT
LSURT(NHIT) = -LRO
NASLT (NHIT) = NBO


350
.G.
IRIN-SM|?1.E-6
.LE.
RIN=SM
ASTER(LRIN+NBO-1)=SM
.LE.
RIN-SM
.G.
GO TO 500
351
.L.
DIST-RIN
.GE.
GO TO 340
354
NHIT=0
355
NHIT=NHIT+1
SM=RIN
L.SURT(NHIT)=LRI
NASCT(HIT)=NBO
500
END OF DO LOOP


SM-PINF
.GE.
.L.
510
PRINT "ERROR IN G1 AT 510"
GO TO 632
530
SI = SI+SM-DIST
DIST = SM
XP = XB+WB*SM
RETURN
.E.
NASC+2
.NE.
532
DO 631 NN=1, NHIT
NASC = NASCT(NN)
LSURF = LSURT(NN)
LTRUE = 0
LOC = LBODY+2NASC-2
UN3 (LOC, LENT, LEAV1)
UN3 (LOC+1, NENT, NEAV1)
J1 = LEAV
J2 = LEAV+NEAV-1
610
.LE.
LSURF
.G.
600
J1 = LENT
J2 = LENT+NENT-1
620
IRPRIM = MASTER(J2)
.LE.
J1-J2
.G.
625
DO 630 J=J1, J2
IRPRIM = MASTER(J)
CALL WOWI (IRPRIM, LSURF, NASC, LTRUE)
A1
622
.LE.
NASC-NRPP
.G.
.L.
LSURF
.GE.
GO TO 6632
GO TO 632


A1
LTRUE
.LE.
.G.
CONTINUE DOLOOP
GO TO 650
6630
NASC-NRPP
.G.
.LE.
6631
LSURF
.G.
.E.
.L.
GO TO 631
GO TO 632
6632
CALL RPP2 (LSURF, XP, IRP)
IRP
.LE.
.G.
6635
IRPRIM=0
RETURN
6636
LTRUE = 0
LOC = LBODY+2*IRP-2
UN3(LOC,LENT,LEAV)
UN3(LOC+1,NENT,NEAV)
J1 = LENT
J2 = LENT + NENT-1
J1-J2
.LE.
.G.
GO TO 632
6640
DO 6650 J = J1, J2
IRPRIM = MASTER(J)
CALL WOWI IRPRIM, LSURF, IRP, LTRUE
LTRUE
.G.
GO TO 650
.LE.
6650
CONTINUE


6650
CONTINUE
631
CONTINUE
632
IERR=IERR+1
PRINT ALL
DIAGNOSTIC
INFO
IRPRIM-1
800
RETURN
6_0
IR-IRPRIM
.E.
.NE.
655
S1
.E.
.G.
.L.
IR=IRPRIM
GO TO
20
ABS(S1)-1.E-6
.LE.
.G.
680
ISPOT=LIRFO+IR-1
UN3(ISPOT,ICODE,IDENT)
ISPOT=LIRFO+IRPRIM-1
UN3(ISPOT,ICODE,IDENT)
IDENT-1
.NE.
.E.
684
IDENT-IDENT1
.E.
.NE.
ICODE-ICODE1
.E.
.NE.
690
RETURN
690

is set equal to the new region number, G1 is called again, and tracing of the ray is continued with the new values of IR and DIST.

Subroutine GENI

GENI's purpose is three-fold:

1. To enter into the machine the input describing the geometry.
2. To check for some input errors and print suitable diagnostic messages.
3. To set up the MASTER-ASTER array for reference by the ray-tracing subroutines.

The following data sets are read in and processed by GENI.

Set 1 - The Title Card

This makes it possible to distinguish between different jobs run.

Set 2 - Contains the Limitations of the Problem

NRPP	The number of rectangular parallelepipeds.
NTRIP	The number of triplets to be read in.
NSCAL	The number of scalars to be read in.
NBODY	The number of different bodies used to describe the geometry other than RPP's.
NRMAX	The maximum number of regions made by the RPP's and the other legal geometric configurations.

IPRIN — This is the printing option. If the number is other than zero the MASTER-ASTER array will be printed out upon completion of GENI.

SCALE — This number is used to scale all geometry data.

Set 3 - Contains the Six Bounding Planes for each RPP

The bounds are given in the following order: X-lower, X-upper, Y-lower, Y-upper, Z-lower, Z-upper.

Set 4 - Contains the Triplet Data

This is read in only if NTRIP >0.

Set 5 - Contains the Scalar Data

This is read in only if NSCAL >0.

Set 6 - Contains the Body Description

Numbers are type integer and refer to specific triplet and scalar data, or they are floating point and the actual description.

Set 7 - Contains the Region Combination Data

Each region is described by the surfaces that compose it.

Rectangular Parallelepipeds

If NRPP is other than zero, GENI calls subroutine RPPIN which reads in Set 3 and places in the main array this data returning with the next available location in the main array. (This will be described in detail later.)

Triplet and Scalar Data

If NTRIP and NSCAL are positive, Sets 4 and 5 are read into the bottom of the ASTER array.

Description of Bodies

After reading the first card of the sixth set, a check is made to see if the type of body is one of the legal configurations. If so, there is a second test to see whether the card should contain floating or fixed point numbers. If fixed point input is used, the information on the card is then converted by DIGCON and the number is assumed to refer to triplet and scalar data. If floating point input is used, the information on the card is converted by FLOCON and is stored at the bottom of the main array in its proper location by SEE3. This operation is performed NBODY times. The array, which was built up at the end of the main array, is then shifted up to the next available location in the main array and the location numbers in the MASTER array referring to this data are adjusted accordingly. Note that ARB input must be in floating point form.

Region Combination Data

The first card of each region in the seventh set gives the region number, logical operations, and the bodies that compose a particular region. A test is made to see that the body numbers are valid and that the logical operations are true. Both the logical operations and the body data are

stored in the MASTER array. The next card is read in and tested to see if the information it contains is a continuation of the previous card or a new region. This procedure is done NRMAX times. Then a test is made on all the data to see if any points are in more than one region. This would be an error because it means that the region described is actually the same as, or part of, another region.

Entering and Leaving Tables

After reading and checking of all data sets, the entering and leaving tables are prepared. These tables denote the possible regions that a ray may enter upon entering or leaving a given surface.

The total room taken by the geometry is then calculated and a test is made on the printing option IPRIN.

Subroutine GRID

The GRID routine is called by MAGIC. GRID will control all the input and processing for a single attack plane and then return to MAGIC. The routine has an input reading phase, an input calculation phase, and a ray calculation phase.

The input phase will read:

NX	The number of 4-in. cells in the horizontal direction.
NY	The number of 4-in. cells in the vertical direction.

IRSTART	The region number enclosing the target and attack plane.
IINC	Same as above.
KWG	The ray number at which ray processing will begin.
A	Azimuthal angle (degrees).
E	Elevation angle (degrees).
ENGTH	The back-off distance.
TESTR	A flag indicating the presence of XNP data.
XNP	The coordinates of the vehicle origin, if other than (0, 0, 0). The XNP data is read only if TESTR is not zero.

The input processing phase will convert the azimuthal and elevation angles to radians and compute:

CA = COS(A)

CE = COS(E)

SA = SIN(A)

SE = SIN(E).

The error flag IERR is set to zero.

The calculation phase will set up the ray coordinates on the attack plane and compute the ray direction cosines. Subroutine TRACK is called to perform the ray-tracing calculations for the ray. After all calculations are performed, TRACK will return to GRID. At this point the error flag (IERR) is examined. If more than 10 errors have occurred during the calls to TRACK, the processing of the attack plane will be terminated and GRID will return to MAGIC.

If 10 errors have not occurred, the next ray will be processed. The calculation procedure is outlined below. A figure detailing the orientation of the grid plane is given in Section 3.2.2.

DO 250 KK = KWG, KL (a loop to control all rays, KK counts rays)

```
WB(1) = -CE*CA
WB(2) = -CE*SA
WB(2) = -SE
```

Calculation of direction cosines from the input azimuthal and elevation angles.

```
II = ((KK-1)/NX)+1
J  = KK- (II-1)*NX
V  = ((NY/2)-II)*4.0+2.
H  = ((NX/2)-J)*4.0+2.
```

Calculation of grid square coordinates (V, H) in the grid plane.

```
IV = RANF(-1)*10.
IH = RANF(-1)*10.
```

Calculation of two random digits in the range (0, 9).

```
V = V+.4*FLOATF(IV)+.2
H = H+.4*FLOATF(IH)+.2
```

The "V, H" coordinates are now at a random point within the grid square.

```
IVIH = IH*10+IV
```

Form a two digit random number for printing by CALC.

```
XBS(1) = -V*CA*SE-H*SA
XBS(2) = -V*SA*SE+H*CA
XBS(3) = V*CE
```

Calculation of X, Y, Z coordinates in the coordinate system of the vehicle.

```
DI 210 MKJ = 1, 3
210 XBS(MKJ) = XBS(MKJ)+XNP(MKJ)
```

Translate coordinates to real vehicle center.

```
Call TROPIC (WP)
```

A routine to generate random direction cosines (WP) from an isotropic distribution.

```
XBS(1) = XBS(1)+WP(1)*1.0E-4
XBS(2) = XBS(2)+WP(2)*1.0E-4
XBS(3) = XBS(3)+WP(3)*1.0E-4
```

The grid coordinates are perturbed to avoid rays exactly on grid boundaries.

```
XBS(1) = XBS(1)-ENGTH*WB(1)
XBS(2) = XBS(2)-ENGTH*WB(2)
XBS(3) = XBS(3)-ENGTH*WB(3)
```

The XB which are X, Y, Z coordinates are not a distance ENGTH from the vehicle center.

At this point, the position on the grid XB and the direction WB have been calculated and TRACK is called.

Function ISIGN (V)

Purpose of Routine

To return to the calling program an alphanumeric "blank" if the variable V is positive or zero and to return an alphanumeric "minus" if V is negative.

Description of Routine

This routine is a function subprogram and is called in subroutine TRACK. It is used in the section of TRACK that outputs the identification and data cards.

Routines Called by This Routine	Routines Which Call This Routine
none	TRACK

Subroutine OPENK (I, J, K, L, M)

This routine unpacks the four items I, J, K, L which are stored in a single computer word in the array ITR at location ITR(M).

The four items were packed by subroutine TRACK.

Description of Routine

The four items are packed using 12 binary bits per item. The packed word appears as

	12 bits	12 bits	12 bits	12 bits	
not used	I	J	K	L	{computer word}

The actual unpacking is described below:

LI = ITR(M)	LI contains the complete packed word
L = LI*7777	The "*" denotes a logical "and" operation. L now contains the 12 right-most bits of LI.
LI = LI/I12	The word LI is shifted 12 bits to the right by a division. The variable I12 has the value $2^{12} = 1024$. LI now appears as:

not used	I	J	K

```
K = LI*7777          Again perform the logical "and" and
                     extract the correct value of K.

LI = LI/I12          Again shift LI and extract the
J  = LI*777          desired packed item - and so on.

LI = LI/I12
I  = LI*777

  RETURN.
```

NOTE: The array ITR is communicated through Common.

Function S (I, N)

The routine is used to retrieve the coordinates of any of the six sides of a rectangular parallelepiped (RPP). Given "I" the ordinal number of the RPP and "N" the side number where N = (1, 2, 3, 4, 5, 6) refer to (X_L, X_U, Y_L, Y_U, Z_L, Z_U), the routine will compute the location in the ASTER array of the required coordinate.

To understand the procedure it is necessary to know the structure of the data in the ASTER array. Each RPP occupies six computer words starting at location LBASE. Each of the six words contain three packed integer variables as shown on the following page.

ASTER Array

LBASE				
LBASE + 1	The	Location	Location of	
+ 2	number	of	floating point	
+ 3	of	complex	word containing	
+ 4	complex	surface	the	RPP_1
+ 5	surfaces	data	coordinate	
+ 6				
+ 7	As	As	As	RPP_2
+ 8	above	above	above	
	15 bits	15 bits	15 bits	

Thus the N^{th} coordinate of the I^{th} RPP is found in the right-most 15 bits of ASTER {LBASE + 6(I-1) + N}.

The actual FORTRAN statements are:

```
L = LBASE + 6(I-1) + N-1      Retrieve the location of
                              the packed word
LL = ASTER(L).AND.77777B      Retrieve the right-most
                              coordinate
S = ASTER(LL)                 S is now the desired
                              coordinate.
```

Subroutine SEE3

Subroutine SEE3 is called by GENI and ALBERT. The routine accepts either triplets or scalars and places them in the MASTER array. A search is made through the MASTER

array to determine if the triplet or scalar already appears in the array. If so, the data will not be stored again and the location IWH of the data is returned to the calling program.

If the data do not appear, they are first scaled and then added to the array. Data location IWH in the array is returned.

The triplet data are passed to the subroutine by the arguments FX, FXX, FXXX. The scalar data are passed by the argument FX.

The argument LSI denotes whether triplet or scalar data are to be stored. LSI is zero for triplet data and nonzero for scalar data.

Subroutine SETUP (DD, I, IF, N)

The integer and fractional parts of the floating point variable DD are stored in the array IP. The variables I and IF denote the locations in the array IP at which the integer and fractional parts of DD are to be stored. The variable N indicates the number of digits in the fractional part. The routine is used by subroutine TRACK to convert thicknesses and normal distances into the proper form for output on the identification and data cards.

The routine considers only the absolute value of DD. The correct sign is output by the function ISIGN. The details of the program are given by the flow chart below.

D = ABSF(DD)+.005	Form the absolute value and round off.
IP(I) = D	The integer part of D is stored in IP(I)
DI = IP(1)	DI now contains the floating point representation of the integer part.
F = (D-DI).10^n	D-DI is the fractional part. (D-DI) 10^n gives the desired number of digits.
IP(IF) = F	F is converted to an integer and stored in IP (IF).

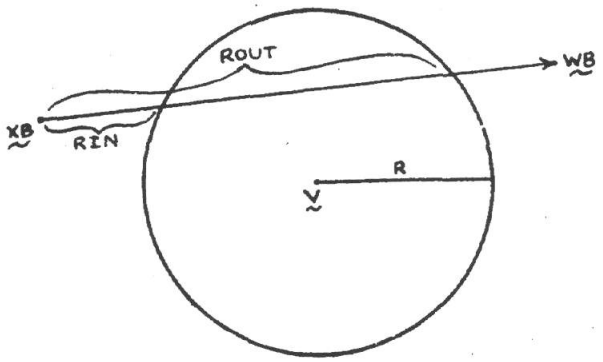
RETURN.

NOTE: The array IP is communicated through Common.

Subroutine SPH

Given a point in space XB(1), XB(2), XB(3), a direction WB(1), WB(2), WB(3), this routine will compute distances to a sphere of radius R and center V(1), V(2), V(3). Two distances, RIN and ROUT, are computed. RIN is the distance to first contact and ROUT is the distance to last contact.

The distances RIN and ROUT are used by subroutines G1 and WOWI to determine which body and region the ray is to intersect.



The equation of a sphere is given by:

$$(X-Vx)^2 + (Y-Vy)^2 + (Z-Vz)^2 - R^2 = 0 \qquad (1)$$

where Vx, Vy, Vz are the center of the sphere, and R is the radius.

The parametric equations of the ray are given by:

$$Xo + WxS = X$$
$$Yo + WyS = Y$$
$$Zo + WzS = Z \qquad (2)$$

where Xo, Yo, Zo are the starting point of the ray and Wx, Wy, Wz are the direction cosines of the ray.

Substituting (2) in equation (1) we obtain

$$(Xo+WxS-Vx)^2 + (Yo+WyS-Vy)^2 + (Zo+WzS-Vz)^2 - R^2 = 0.$$

Let $Dx = Xo-Vx$, $Dy = Yo-Vy$, $Dz = Zo-Vz$

$$(Dx+WxS)^2 + (Dy+WyS)^2 + (Dz+WzS)^2 - R^2 = 0$$

and

$$S^2 \{Wx^2+Wy^2+Wz^2\} + S.2 \{DxWx+DyWy+DzWz\} + Dx^2+Dy^2+Dz^2-R^2 = 0$$

A B C

Thus,

$$S = -B \pm \sqrt{B^2-C} .$$

Note that $A = Wx^2 + Wy^2 + Wz^2 = 1$.

The two roots are the desired RIN and ROUT, where

$$RIN = -B- \sqrt{B^2-C}$$

$$ROUT = -B+ \sqrt{B^2-C}$$

Note that if $B^2-C = 0$ the roots are imaginary and no intersection occurred. In the case of no intersection we set RIN = ∞

ROUT = $-\infty$

where ∞ is taken to be 10^{50}.

Subroutine TRACK

TRACK has the function of accepting grid coordinates from the GRID routine and initiating a ray. It follows the ray through all points of contact, records the important data, and punches both the identification and data cards for each ray.

The routine begins by starting the ray, and computing the distance from the first vehicle component hit to a plane parallel to the attack plane and passing through the vehicle center. It then proceeds to record distances through each component encountered, checking each hit for armor shell, skirting, or interior volume, as noted on the region identification card. When the last component is hit, it once more computes the distance to the attack plane.

After hitting the last component, the punching phase is begun. TRACK decides first if any card(s) should be punched (it is possible to miss the target, and thereby get no useful information). In the event that the vehicle was hit, TRACK first punches the ID card, then calls on subroutine CALC to compute normal distances and organize the outputs for the data card(s). CALC is called twice per data card, as two hits appear on each. After the results are all punched, control goes back to GRID.

A detailed description of the CALC routine appears on the following pages. The description should be followed in conjunction with a FORTRAN listing.

Begin Track Computation

NASC = -1	(Get ready for a new ray)
KHIT = 0	(Number of hits = 0)
IR = IRSTART	(Starting region number)
L = 1	(Number of intersections is one to start)

MSKRT = 0 (No skirting yet)

MVOL = 0 (No interior volume yet)

JCNT = 0 (Initialize a counter).

The DO 5 Loop clears the two storage arrays used by program TRACK at the start of a ray.

Statement 10

Call G1 to start or continue the ray from XB in the WB direction starting in region IR. G1 returns with IRPRIM new region number, XP point of contact, S1, distance through region IR.

TR(L) = SL	The TR array will contain the line-of-sight distance from contact to contact.
KLSURF = LSURF+7	Add seven to what might be a negative LSURF number, (1-6) indicating the ray is leaving that surface number.

The line "ITR(L) =", preserves in the ITR ARRAY, indexed by L number, the following information: surface number, body number, next region, this region. These data can be retrieved by program OPENK and is done in CALC.

Check L $\geq$ 500 and stop if this occurs, because only room for 500 intersections per ray is allocated.

Statement 41

If the next body entered is an RPP, the region of interest has been gone through and IRPRIM = 0 which terminates the ray.

Statement 42

LMAX = L (number of last contact).

The DO 43 Loop calculates the distance D2 from the last point XP to the center XBS. S1 is subtracted because this is the distance from last contact to the outside of the configuration.

Some printing for debugging at this point is optional on SSW6. Control is then transferred to Statement 60 and all shooting of rays and data recording are complete.

Statement 45

IR = IRPRIM	Set 'this region' indicator to be the next region to be encountered, in preparation for continuing the ray.
KHIT = KHIT+1	A running count of number of components hit along this ray.

Compute "D1" the distance from the attack plane to first contact if L = 1.

Statement 46

This is the calculation for D1, the distance between XBS and XP, using the standard distance formula.

Statement 19

The value of IDENT, a special number added to the description of each region and unpacked previously from its storage place in MAGIC. If IDENT = 0, there is no special material in this region and go to 35.

If IDENT is negative, it can only be a space. (-1) is the only negative legal identification number.

If IDENT is positive, Statement 20 tests for IDENT = 10 which indicates armor in the region. A flag so indicating, MARMR = 1, is set and control goes to 35.

At Statement 20, IDENT -10 negative indicates a space and control goes to 31.

Statement 31

If IDENT = 1, the space is interior volume and MVOL = 1. If IDENT = 1, the space is exterior volume and MVOL is not set.

Statement 48

L = L+1 count another intersection.

GO TO 10

Continue the ray, all data for the current intersection have been processed.

Statment 60

At tnc point when Statement 60 is reached in TRACK, the following has occurred:

1. A ray from some point in GRID square (II, J) called XB has been transversed in the WB direction until the outside of the geometry was reached.

2. At each intersection with the ray and a region, both the ITR and TR arrays were used to store at index L (intersection number) the data describing the intersection.

3. D1, D2 distances to first and last components were computed and KHIT is the number of vehicle components hit.

4. We now are ready to punch the data associated with this ray, i.e., ID card and DATA cards.

If (LMAX-1) = 0 there was no intersection with the vehicle geometry and TRACK terminates by going to 85.

At this point, program SETUP does the following:

CALL SETUP (AAA, J, K, N)

IP(J) = Integer portion of AAA

IP(K) = 'N' decimal digits of AAA where "IP" is in Common.

Function ISIGN gives the Hollerith blank for positive numbers and minus for negative numbers.

This suffices for an explanation of all variables up to IP(14), 15, 16, 17 which are set equal to the integer variables which indicate the presence of skirting, target armor, and interior volume. If present, target is always 1.

Statement 62

Subtract one from KHIT to make it the correct value (don't count last contact with outside).

The DO 330 Loop resets the temporary storage place of the card to zero prior to punching of a "Data" card.

Statement 70 (+L = 0)

Initialize L and TRAVEL to zero to start ray results.

TRAVEL = TR(1) distance to first intersection.

The DO 80 Loop sets up logic for LMAX/2 data cards. In fact, there may be less if some spaces are encountered.

L = L+1: begin work on a new intersection.

Test L > LMAX: if not, continue, if yes, terminate at 85.

Statement 71

Call CALC which will compute:

1. NIR - Region identification (vehicle component)
2. SIT - Line-of-sight distance
3. ANGLE - Angle of normal with ray (from XB in WB)
4. SN - Normal distance through region
5. NTYPE - Type of space follows component "NIR", zero if none.
6. SSPACE - Line-of-sight distance through space, zero if none.

CALC will update L if a space is encountered, and computes all of the above each time it is called.

Statement 101

If SSPACE was not zero, set JCNT = JCNT+1 (to keep track of how many spaces were hit).

The DO 79 Loop resets IP to zero and processes another data card. The process continues until no data cards remained to be punched. TRACK then returns to GRID.

Subroutine TROPIC (WP)

The routine will calculate isotropic direction cosines WP(1), WP(2), WP(3). Each call to TROPIC will deliver a different set of direction cosines. The procedure is outlined below.

1. Pick two random numbers X_1, X_2 such that $0 \leq X_1, X_2 \leq 1.0$
2. Continue picking sets of X_1, X_2 until $X_1^2+X_2^2 \leq 1.0$
3. Compute:

$$\cos\phi = \frac{X_1^2-X_2^2}{X_1^2+X_2^2}$$

$$\sin\phi = \frac{2.0\ X_1 \cdot X_2}{X_1^2+X_2^2}$$

The $\cos\phi$ and $\sin\phi$ are the sine and cosine of a random angle ϕ.

4. Compute:

$$CS = 2.0*X_3-1.0$$

where X_3 is a random number $0 \leq X_3 \leq 1.0$,

and CS is the cosine of a random angle.

5. The direction cosines are:

$$WP(1) = 1-CS^2 \cdot \sin\Phi$$

$$WP(2) = 1-CS^2 \cdot \cos\Phi$$

$$WP(3) = CS$$

Subroutine UN1 (LOC, I3)

This routine unpacks the item I3 from a single computer word packed as shown

	15	15	15 bits
Not used	I1	I2	I3

The items I1 and I2 are also packed, but are not unpacked by this routine.

The unpacking is done by the following statements:

```
I = MASTER (LOC)
I3 = I-(I/32768)*32768.
```

Subroutine UN3 (LW, J1, J2, J3)

This routine unpacks the three items J1, J2, and J3 which are stored in a single computer word in the array MASTER at location LW.

The three items are packed using 15 binary bits per item. The packed word appears as:

	15	15	15
Not used	J1	J2	J3

The individual items are extracted by multiplication and division.

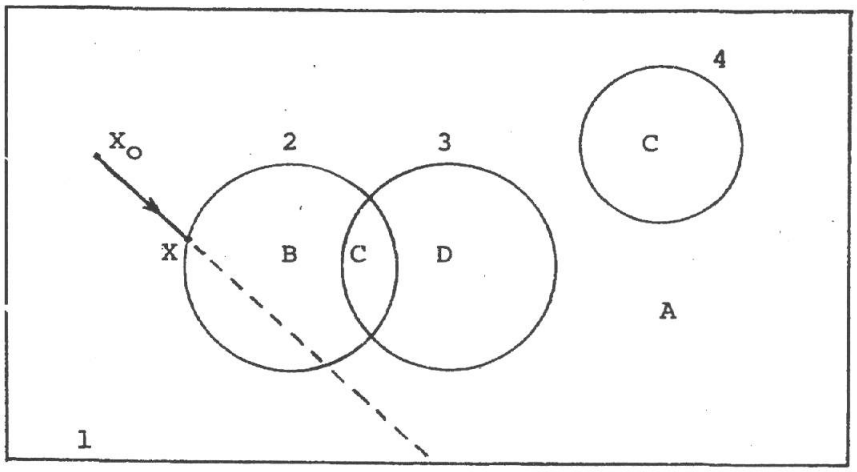
Subroutine WOWI (JREG, LSURF, NEX, LTRUE)*

The logic employed in subroutine WOWI is, in essence, the heart of the combinatorial geometry method. WOWI's only function is to answer the following question. Given a point X and a region number JREG, does the point lie within the region? The following is a detailed description of the method used by WOWI to make this decision. The flow chart appearing after this discussion closely follows the actual coding in WOWI.

Probably, the clearest way to describe WOWI is to follow the routine through a simple example. Consider the following geometry, consisting of three spheres (labeled

* The arguments LSURF and NEX are not used in the present version of the code but appear in the call statement to facilitate future expansion of the code.

bodies 2, 3, and 4) enclosed in a box (labeled body 1). For clarity, the regions have been designated by letters A, B, C, and D, although they are numbered in the actual code. Note that the discontinuity of region C is perfectly legal. The starting point and current position of a ray are labeled X_o and X, respectively.



Before proceeding with the description of WOWI, let us list two vital blocks of information, both contained in the MASTER-ASTER array. The first block contains the region descriptions of each region in the form of a logical combination of bodies.

Region	Bodies
A	+1-2-3-4
B	+2-3

Region	Bodies
C	OR(+2+3) OR(+4)
D	+3-2

The meaning of these equations is discussed in Section 2.1.

The second piece of information needed by the code is the entering and leaving table for each body. These tell the code which regions a ray might be in if it enters a given body and which regions the ray might go into if it leaves a given body. In this example, the table would appear as follows.

Body	Entering	Leaving
1	A	Escapes from geometry
2	C or B	A or C or D
3	C or D	A or B or C
4	C	A or C

Assume that the ray has been traced from its starting point X_o to a point X on the surface of body 2. G1 computes DIST, the total distance traveled along this path ($=X-X_o$) and investigates the entering and leaving table for body 2. It finds that a ray entering body 2 can be either in region C or in region B.

Region C is investigated first by setting JREG = C and calling WOWI. WOWI locates the region description of C in the MASTER-ASTER array and tests whether the point X satisfies the logical equation for C. This is done in the follow-

ing way. Two distances, RIN and ROUT, are computed for each body mentioned in the description. These are the distances between X_0 and the point where the ray enters (RIN) or leaves (ROUT) the body. In this example, RIN and ROUT are computed for bodies 2, 3, and 4 since these appear in the description of C. Note, however, that the ray does not actually intersect bodies 3 and 4. In this case, ROUT is set equal to -10^{50} for these bodies. We also note that for body 2, RIN = DIST, since point X is where the ray enters the sphere. ROUT is equal to RIN plus the path length of the ray through the sphere.

WOWI now determines if point X satisfies the description of C by testing each body in C against the following rules:

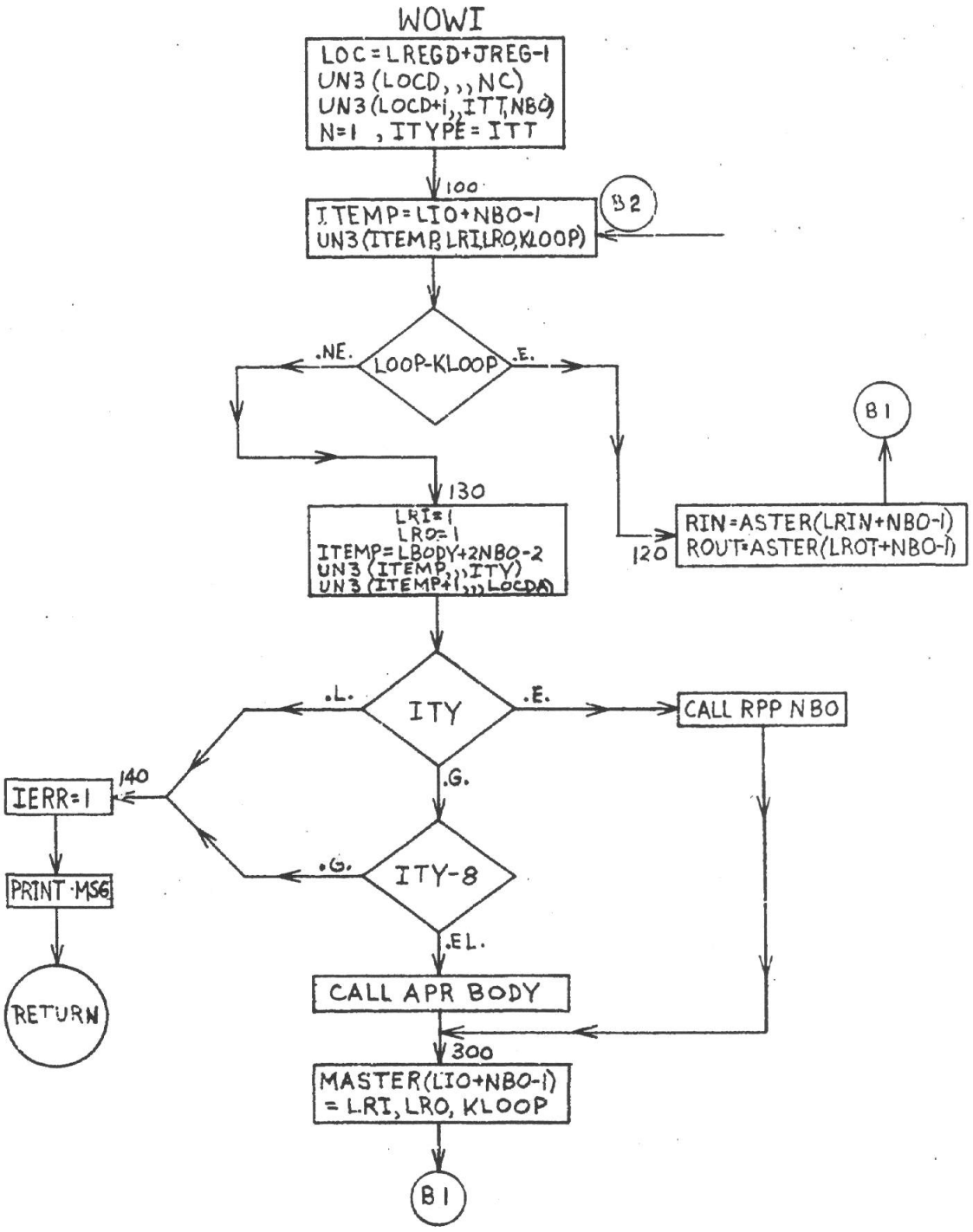
1. A (+) operator is valid if ...
   ROUT > 0 and RIN ≤ DIST < ROUT
2. A (-) operator is valid if ...
   ROUT = -10^{50} or DIST <RIN or DIST ≥ ROUT
3. An (OR) operator is valid only if every (+) and (-) within the (OR) statement is valid.
4. A region description containing one or more (OR) statements is satisfied if any one of the (OR) statements is valid.
5. A region description containing no (OR) statement is satisfied only if every (+)

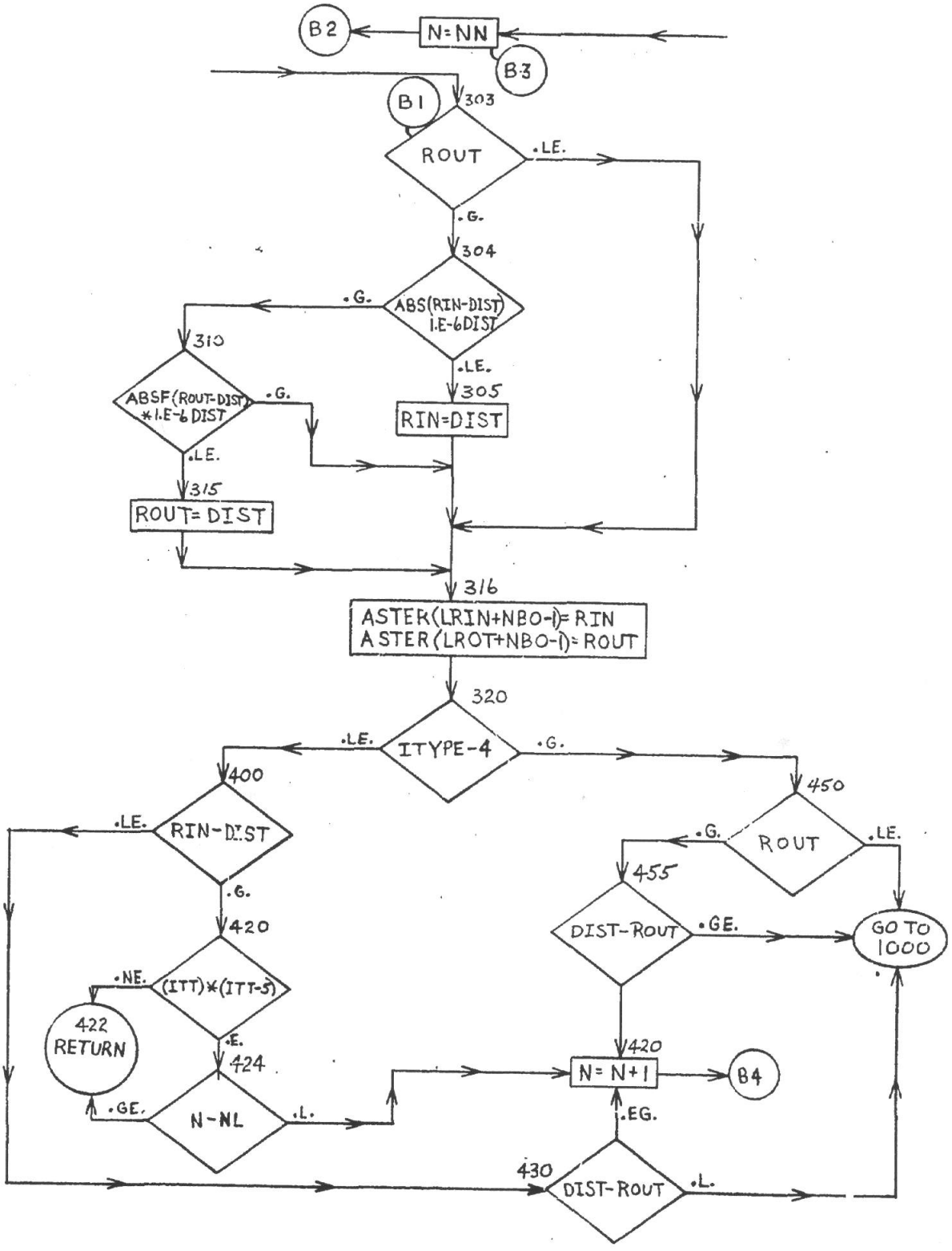
and (-) operator is valid.

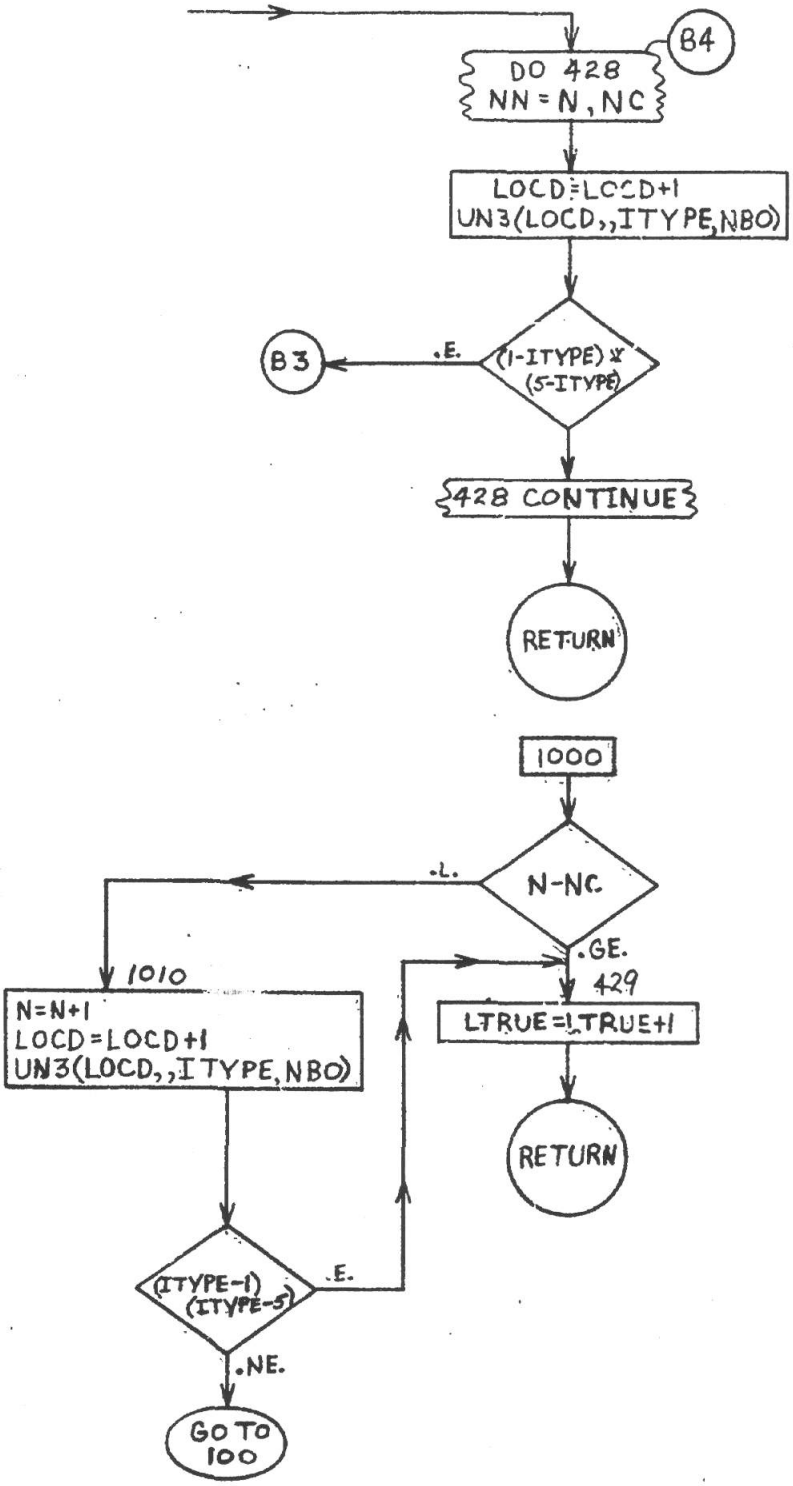
6. A sufficient condition for point X to be in region JREG, is that the region description of JREG be satisfied (two regions cannot be satisfied for the same point).

WOWI now applies these rules to the first (OR) statement for region C, namely OR(+2+3). The +2 is valid from rule 1, since DIST = RIN and ROUT is greater than zero. The +3, however, is not valid since the ray misses body 3 and ROUT = -10^{50}. Thus, the first (OR) statement is invalid from rule 3. The second (OR) statement, OR(+4), is also not valid since ROUT = -10^{50} for body 4. Therefore, from rule 4, the region description of C has not been satisfied and the point X is not in C. WOWI sets LTRUE = 0 and returns to G1. The value of zero informs G1 that region C has not been satisfied. G1 now sets JREG = B and calls WOWI again.

The region description for B is now investigated. The +2 is valid as seen above and the -3 is also valid since, for body 3, ROUT = -10^{50}. Therefore, the description of B is satisfied by rule 5. WOWI sets LTRUE = 1, informing G1 that the point X is in region B. This completes WOWI's task until a new point X has been computed by G1.


WOWI
LOC=LREGD+JREG-1
UN3(LOCD,,,NC)
UN3(LOCD+1,,,ITT,NBO)
N=1 , ITYPE=ITT
100
ITEMP=LIO+NBO-1
UN3(ITEMP,LRI,LRO,KLOOP)
B2
.NE.
LOOP-KLOOP
.E.
B1
130
LRI=1
LRO=1
ITEMP=LBODY+2NBO-2
UN3(ITEMP,,,ITY)
UN3(ITEMP+1,,,LOCDA)
RIN=ASTER(LRIN+NBO-1)
ROUT=ASTER(LROT+NBO-1)
120
.L.
ITY
.E.
CALL RPP NBO
.G.
140
IERR=1
.G.
ITY-8
PRINT MSG
.EL.
RETURN
CALL APR BODY
300
MASTER(LIO+NBO-1)
=LRI, LRO, KLOOP
B1


B2
N=NN
B3
B1
303
ROUT
.LE.
.G.
304
ABS(RIN-DIST) .LE. 1.E-6*DIST
.G.
310
ABSF(ROUT-DIST) .LE. 1.E-6*DIST
.G.
.LE.
305
RIN=DIST
315
ROUT=DIST
316
ASTER(LRIN+NBO-1)=RIN
ASTER(LROT+NBO-1)=ROUT
320
ITYPE-4
.LE.
.G.
400
RIN-DIST
.LE.
.G.
420
(ITT)*(ITT-5)
.NE.
.E.
422
RETURN
424
N-NL
.GE.
.L.
450
ROUT
.G.
.LE.
455
DIST-ROUT
.GE.
GO TO 1000
420
N=N+1
B4
.EG.
430
DIST-ROUT
.L.


B4
DO 428
NN = N, NC
LOCD=LOCD+1
UN3(LOCD,,ITYPE,NBO)
B3
.E.
(1-ITYPE)*(5-ITYPE)
428 CONTINUE
RETURN
1000
.L.
N-NC.
.GE.
429
1010
N=N+1
LOCD=LOCD+1
UN3(LOCD,,ITYPE,NBO)
LTRUE=LTRUE+1
RETURN
(ITYPE-1)(ITYPE-5)
.E.
.NE.
GO TO 100

Subroutine XDIST (XA, XB)

This routine computes the distance between point XA and point XB. Both XA and XB are arrays of length 3.

The routine is a function subprogram which computes the distance between two points using the familiar distance formula shown below.

$$\text{Distance}^2 = (XA(1)-XB(1))^2 + (XA(2)-XB(2))^2 + (XA(3)-XB(3))^2$$

where XA(1), XA(2), XA(3) are the X, Y, Z coordinates of the point.

This routine is used by the 'DCOSP' subroutine as part of the direction cosine calculation. The routine is also used in subroutine CALC as part of the normal distance computations.

3.3.2 Description of Output

The output consists of a repetition of input, and computed results. The items of output are listed in order of appearance.

1. Date Line. The date as read in on the date card which preceded the input deck.
2. Time Line. The time is printed in seconds and is an elapsed time indicator.
3. Comment Line. The comment card which allows the user to title the problem.

4. Parameters. The input parameters are listed.

5. Scale Factor. The scaling number used to scale the geometry.

6. RPP Data. The unscaled RPP data as read into the computer.

7. Triplet Data. Repeated as in the input deck.

8. Scalar Data. Repeated as in the input deck.

9. Body Data. The unscaled body data given as input, the bodies have been ordinarily numbered.

10. Variable Location Line. This line gives the location in the MASTER array of positioning constants.

11. Region Combination Data. This is a repetition of the input region data.

12. Total Room for Geometry Data. This is the number of words used in the MASTER array to contain the geometry.

13. Entering and Leaving Tables. The entering and leaving table as calculated for each region, and its location in the MASTER array are reprinted here (see Section 3.3.1, the WOWI routine).

14. MASTER-ASTER Array.* This is a printed copy

* This appears only if "IPRIN" in Column 60 on the parameter input card was nonzero.

in both integer and floating form of the MASTER array. There are three entries per line separated by a dollar sign.

15. Region Code Identifiers. This is the printed copy of the input which specifies region identifier and space type.

16. Number of Aspect Angles. Repeated from input. Indicates how many different attack planes are to be processed.

17. GRID Specifications. This is a reprinting of input to the GRID program, azimuth and elevation are printed twice, first in degrees and then in radians.

18. Ray Results. This is the output from the TRACK program and has three formats.

    a. A line beginning RAY contains cell number X-Y position, coordinates of the ray origin point, and the ray direction cosines.

    b. Identification card*

    c. Data cards.*

---

* Detailed formats of the identification and data cards are given on the following pages.

IDENTIFICATION CARD

1.	Col 1 - Col 6	X* coordinate of grid square (inches).
2.	Col 8 - Col 13	Y coordinate of grid square (inches).
3.	Col 14 - Col 16	A two-digit uniform random number. (Column 14 is always blank.)
4.	Col 18 - Col 24	The perpendicular distance from the first component contact to the X, Y plane (inches).
5.	Col 26 - Col 32	The perpendicular distance from the last component contact to the X, Y plane (inches).
6.	Col 33 - Col 34	Indicates if skirting is present in the grid square. (A one indicates yes, a zero indicates no.)
7.	Col 35 - Col 36	Indicates if there is a target hit in the grid square.
8.	Col 37 - Col 38	Indicates if there is armor shell present (turret or hull).
9.	Col 39 - Col 40	Indicates if there is interior volume present.
10.	Col 41 - Col 43	The total number of components encountered along the line-of-sight of the penetrator.
11.	Col 45 - Col 47	X divided by 4. (X being the coordinate of the grid square.)
12.	Col 48 - Col 50	Y divided by 4. (Y being the coordinate of the grid square.)

(Columns 7, 17, 25, 44 and 51 through 80 are left blank.)

* Note that "X" and "Y" refer to the virtual horizontal and vertical on the attack plane.

DATA CARD

1.	Col 1 - Col 4	A vehicle component.
2.	Col 5 - Col 11	The line-of-sight distance through the component (inches).
3.	Col 12 - Col 18	The normal thickness of the component (inches).
4.	Col 19 - Col 24	The angle of obliquity with respect to the normal to the component (degrees).
5.	Col 25 - Col 27	A type of space following the component.
6.	Col 28 - Col 34	The line-of-sight distance through the space (inches).
7.	Col 35 - Col 38	Same as 1.
8.	Col 39 - Col 45	Same as 2.
9.	Col 46 - Col 52	Same as 3.
10.	Col 53 - Col 58	Same as 4.
11.	Col 59 - Col 61	Same as 5.
12.	Col 62 - Col 68	Same as 6.
13.	Col 70 - Col 72	X divided by 4. (X is a coordinate of the grid square.)
14.	Col 73 - Col 75	Y divided by 4. (Y is a coordinate of the grid square.)
15.	Col 77	Vehicle designation.
16.	Col 78	Azimuthal angle of attack. (A 3 indicates 30°.)
17.	Col 79 - Col 80	The cumulative sum of components encountered along the line-of-sight of the penetrator.

(Columns 69 and 76 are left blank.)

3.3.3 Glossary of Important Variable Names

The following pages contain a listing of the important variables used throughout the MAGIC program. For each variable name the "common block," if any, and a description of the variable's usage is given. In addition, for each common block a list of subroutines using the common block is given.

*LIST OF IMPORTANT VARIABLES FOR MAGIC

Variable Name	Labeled Common	Definition
Dummy	Blank common	An array of 250 used to get proper location for unpacking the MASTER-ASTER array
ASTER	Blank common	Contains floating point numbers of the main array
XB	Param	X, Y, Z coordinates of ray origin
WB	Param	Direction cosines of ray
E	Param	Not used*
IR	Param	Region number tracking from
T	Param	Not used
IDET	Param	Not used
F	Param	Not used
NHIST	Param	Not used
WC	Param	Not used
SP	Param	Not used
WP	Param	Not used
LBASE	Geometry	Starting location of the aster, master array usually one (1)
RIN	Geometry	Distance to entering body intersection
ROUT	Geometry	Distance to leaving body intersection
LRI	Geometry	Entering surface number
LRO	Geometry	Leaving surface number

* Some variables appearing in Common are used only in the nuclear code, SAM-C.

Variable Name	Labeled Common	Definition
PINF	Geometry	Value of 1.E+50 (used to represent infinity)
IERR	Geometry	Flag to indicate an error in the geometry input, also used for number of errors in G1
DIST	Geometry	Total distance traveled from ray origin
NRPP	UNCGEM	Number of regular rectangular parallelepipeds
NTRIP	UNCGEM	Number of triplets
NSCAL	UNCGEM	Number of scalars
NBODY	UNCGEM	Number of bodies
NRMAX	UNCGEM	Maximum number of regions
LTRIP	UNCGEM	Location of triplet data
LSCAL	UNCGEM	Location of scalar data
LREGD	UNCGEM	Location of region data
LDATA	UNCGEM	Address of data (temporary)
LROUT	UNCGEM	Location of ROUT surface number
LRIN	UNCGEM	Location of RIN surface number
LIO	UNCGEM	Address in ASTER of temporary data for G1
LOCDA	UNCGEM	Location of data in MASTER array
I15	UNCGEM	Has the value of 2 raised to the 15th power (used to pack and unpack the middle word in the MASTER array)

Variable Name	Labeled Common	Definition
I30	UNCGEM	Has the value of 2 raised to the 30th power (used to pack and unpack data in the MASTER array)
LBODY	UNCGEM	Location of body data
NASC	UNCGEM	Current body number (-1 to start ray)
KLOOP	UNCGEM	Internal ray counter
XS	TEMPOR	Temporary storage 3 cells
X	TEMPOR	Coordinates of a point of an RPP
IX	TEMPOR	Converted integer numbers
IT	TEMPOR	Temporary storage
IA	TEMPOR	Temporary storage
IN	TEMPOR	Temporary storage
LSURF	LSU	Surface number of body surface hit negative means leaving
D1	GTRACK	Distance from center grid square to first vehicle component
D2	GTRACK	Distance to last component
KHIT	GTRACK	Number of components hit on one ray
TR	GTRACK	Storage array for recording ray contacts
XBS	GTRACK	Original position of ray on grid square
IRSTART	GTRACK	Starting region of grid plane (at ENGTH)

Variable Name	Labeled Common	Definition
IENC	GTRACK	Region in which geometry is enclosed (same as IRSTART)
ITR	GTRACK	Same as TR
CA	GTRACK	Cosine angle A
CE	GTRACK	Cosine angle E
SA	GTRACK	Sine angle A
SE	GTRACK	Sine angle E
NIR	CALC	Next region
SIN	CALC	Line-of-sight distance through a component
ANGLE	CALC	Oblique angle between normal and ray
NTYPE	CALC	Type of component
SSPACE	CALC	Line-of-sight distance through a space
L	CALC	Internal intersection counter
XS	CALC	Same as XB, used by TRACK and CALC
WS	CALC	Same as WE, used by TRACK and CALC
TRAVEL	CALC	Distance along ray so far, used by TRACK
SN	CALC	Normal thickness
V	CALC	Position on grid plane X direction
H	CALC	Position on grid plane Y direction
LIRFO	WALT	Location of component identification codes in MASTER array

Variable Name	Labeled Common	Definition
IP	SETUX	Output array for punch cards, used in TRACK.

LIST OF ROUTINES WHICH CONTAIN EACH COMMON

BLANK COMMON

MAGIC
UN3
UN1
CALC
GENI
RPPIN
S
ALBERT
G1
WOWI
RPP
RPP2
BOX
SPH
RAW
RCC
ELL
REC
TRC
ARB

COMMON/SCALE

MAGIC
GENI
RPPIN
SEE3
ALBERT

COMMON/LSU

G1
TRACK

COMMON/CALC

GIRD
TRACK
CALC

COMMON/TEMPOR

GENI

COMMON/PARAM

MAGIC
GRID
TRACK
CALC
GENI
S
G1
WOWI
RPP
RPP2
BOX
SPH
RAW
RCC
ELL
REC
TRC
ARB

COMMON/WALT

MAGIC
GRID
TRACK
CALC
G1

COMMON/METRY

MAGIC
GRID
TRACK
CALC
GENI
RPPIN
S
ALBERT
G1
WOWI

COMMON/SETUX

SETUP
TRACK

COMMON/METRY

RPP
RPP2
BOX
SPH
RAW
RCC
ELL
REC
TRC
ARB

COMMON/GTRACK

OPENK
GRID
TRACK
CALC

COMMON/UNCGEM

MAGIC
GRID
TRACK
CALC
GENI
RPPIN
S
ALBERT
G1
WOWI
RPP
RPP2
BOX
SPH
RAW
RCC
ELL
REC
TRC
ARB

3.4 PROGRAM IMPLEMENTATION

The MAGIC program operates in two modes. The first is a normal mode in which a geometric description will be read in and processed and aspect angle data produced. This mode will also create a magnetic tape containing the processed geometric data. The second mode, called a production mode, will read the magnetic tape, instead of the actual description, and then perform aspect angle calculations. Thus the normal mode consists of input processing and aspect angle processing. The production mode is just aspect angle processing.

Section 3.4.1 will discuss the operating procedure for the CDC-6600. Tape utilization will be discussed in Section 3.4.2.

3.4.1 Operating Instructions

The following list will detail the system control cards that must precede the data cards.

JOB CARD (The job card is described in detail in the systems manual.)

REQUEST TYYY.

REWIND (TYYY)

COPYBF (TYYY, MAGIC)

REWIND (MAGIC)

REQUEST TXXX, TAPE 4.

REWIND (TAPE 4)

MAGIC

EXIT

REWIND (TAPE 4)

REWIND (TYYY)

RETURN (TAPE 4)

RETURN (TYYY)

EOR (CARD)

Please note:

YYY should be a Cims Tape Number assigned to the tape which contains the binary record of MAGIC.

XXX should be the Cims Tape Number assigned to the tape which will or does contain the processed geometry.

All control cards begin in card Column 1.

Assume that the input data have been prepared according to the formats specified in Section 3.2.3. It is necessary to punch only one new card. This card, a date card, will determine which mode is to be used. If a normal operation is desired, punch the date in Columns 12-20 and leave the rest of the card blank. If production mode is desired, punch a "2" in Column 10 and the date in Columns 12-20 and leave the rest of the card blank. (Date should be "04/07/67" for April 7, 1967.)

Having punched the appropriate date card, if using the normal mode all input data is required. If using the production mode, only the region identifier and aspect data cards are needed. In both cases an end-of-file card is the

last card. Figs. 3.3 and 3.4 illustrate the two deck configurations.

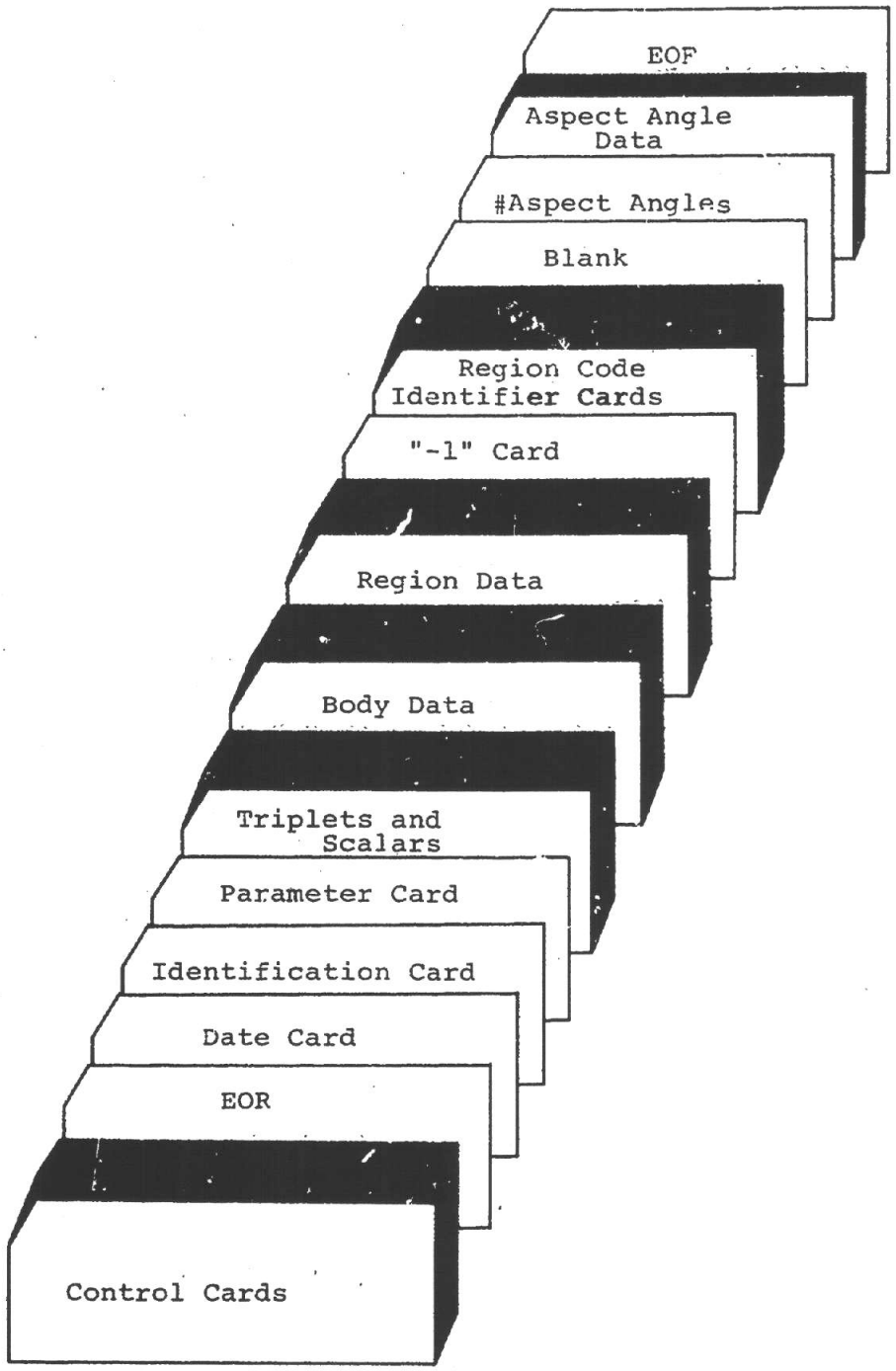


Fig. 3.3 - Normal Mode Deck Configuration

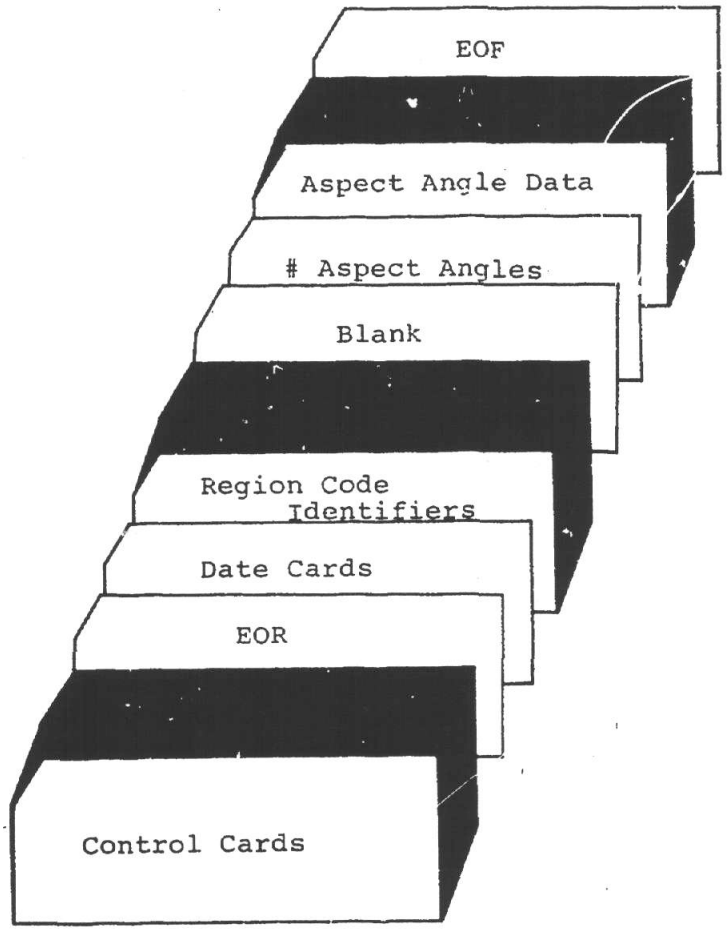


Fig. 3.4 - Production Mode Deck Configuration

3.4.2 Tape Utilization

The MAGIC program requires a "program" tape containing the binary program. This tape is created by compiling the MAGIC symbolic deck.

A tape which will contain processed geometry data is also required.

3.4.3 Error Messages

Routine	Message	Explanation
MAGIC	TERMINATION ON GEOMETRY INPUT ERROR	There should be at least one previous error message, fix what is wrong, try again
TRACK	TRACK ERROR	More than 500 components encountered on one ray, change program (big job).
CALC	BAD ITYPE IN CALC RETURN TO CALC	Indicates either error in routine OPENK or dimension of MASTER-serious error
CALC	STMNT 304 STOP	A TRC with two equal radii was encountered. Make it a RCC. Try again.
CALC	TRACK ERROR	An illegal surface number in a RAW. Serious error indicates bit failure probably.
GENI	ITYPE DID NOT MATCH AN ITY	The body name on last card is spelled incorrectly.

Routine	Message	Explanation
GENI	STMNT 290 GENI	Triplet input attempted for an ARB, this is illegal, use other form.
GENI	ERROR IN REGION INPUT XX	Indicates that region XX incorrect, probably a body number is mispunched.
GENI	ILLEGAL OPERATOR IN ABOVE CARD	Pertains to region description.
GENI	STMNT 490 STOP	Due to previous error.
GENI	STMNT 555 STOP	Due to previous error.
RPPIN	ERROR IN DESCRIPTION	The RPP just printed is not in the min max format required.
FLOCON	ERROR IN FLOCON NO DECIMAL POINT	The body just printed needs decimal points (even on zeros).
FLOCON	ERROR IN FLOCON NO EXPONENT	An E format input number on the last printed card has no exponent.
DIGCON	ERROR IN SUBDIGCON	An integer has an illegal character (nonnumeric or blank).
G1	ERROR IN G1 AT 140	Bad surface number unpacked - serious error could be bits dropped or added.
G1	ERROR IN G1 AT 150 (REGION #)	No bodies are entered or left which are mentioned in this region. Some error of omission in region data probably.

Routine	Message	Explanation
G1	ERROR IN G1 AT 640	None of the regions mentioned in the entering or leaving table are entered. An error of omission in region data.

3.4.4 Sense Switch Options

A debugging feature of the program is available under the control of sense switch 6. If sense switch 6 is down, a printout of intermediate results will occur. The printout will be intermingled with actual aspect angle outputs. Detailed discussions of the printouts will be found in the discussions of TRACK and CALC in Section 3.3.1.

3.5 SAMPLE PROBLEM

A sample problem illustrating the usage of the MAGIC program is given on the following pages. The illustration consists of a sample printout of computer results and a series of figures describing the geometry.

The vehicle being described consists of an outer shell, composed of six arbitrary polyhedrons, and a variety of bodies within the vehicle. Fig. 3.5 consists of three views of the vehicle (front, side, top). Fig. 3.6 shows the layout of the outer shell and an interior "bulkhead." Fig. 3.7 shows the organization of the interior configuration. The numeric identifiers in the figures denote the body numbers used in defining the problem.

The computer printout is annotated for illustration purposes. A more detailed description of the output is given in Section 3.3.2.

For the purposes of brevity, only the first 87 words of the MASTER array have been given in the sample output. In addition, some ray outputs have been deleted from the printout.

The azimuthal and elevation angles in the problem are both zero. Thus the rays are fired from the positive X-coordinate to the negative X-coordinate (as shown in position (1) of Fig. 3.1).

The following describes the annotated items of the computer output:

Item 1	The date card.
Item 2	The internal clock time at start of processing
Item 3	Printout of the input parameter card.
Item 4	The RPP data.
Item 5	The triplet and scalar data.
Item 6	The body data. Note that the left-most digits (as circled) are sequential body numbers as calculated by the program.
Item 7	Intermediate storage allocation data.
Item 8	Region description data. The parenthesis are printed for readability and do not appear on the input cards.
Item 9	The total room, in the MASTER array, for this problem.

Item 10 The entering and leaving tables. These tables are calculated from the region description data. The left-most number is the body number. The second and third numbers are the first and last locations, in the MASTER array, containing the possible regions. The rest of the numbers are the possible regions.

Item 11 A listing of the MASTER array. Three words per line are printed. Each word of the array is printed twice. First, as three unpacked integers, and then as a floating point number.

Item 12 The identification and space codes for each region.

Item 13 The attack plane data.

Item 14 The ray data. Note that the first rays did not hit the vehicle and no ray data cards were produced.

Item 15 The horizontal and vertical attack plane coordinates.

Item 16 The X-Y-Z coordinates of the starting point of the ray.

Item 17 The direction cosines of the ray.

Item 18 The ray data cards as described in Section 3.3.2.

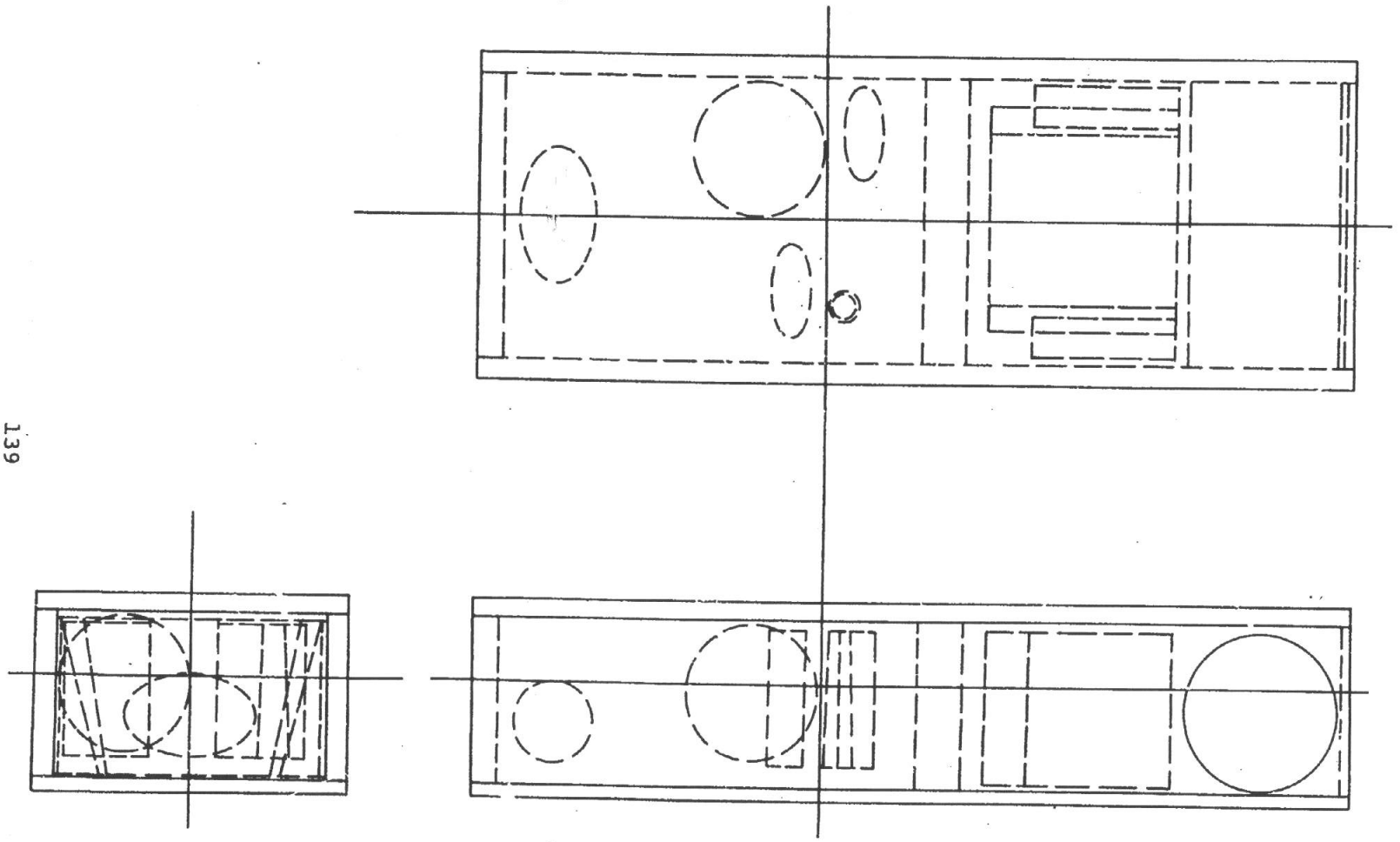


Fig. 3.5 - Three Views of Sample Vehicle

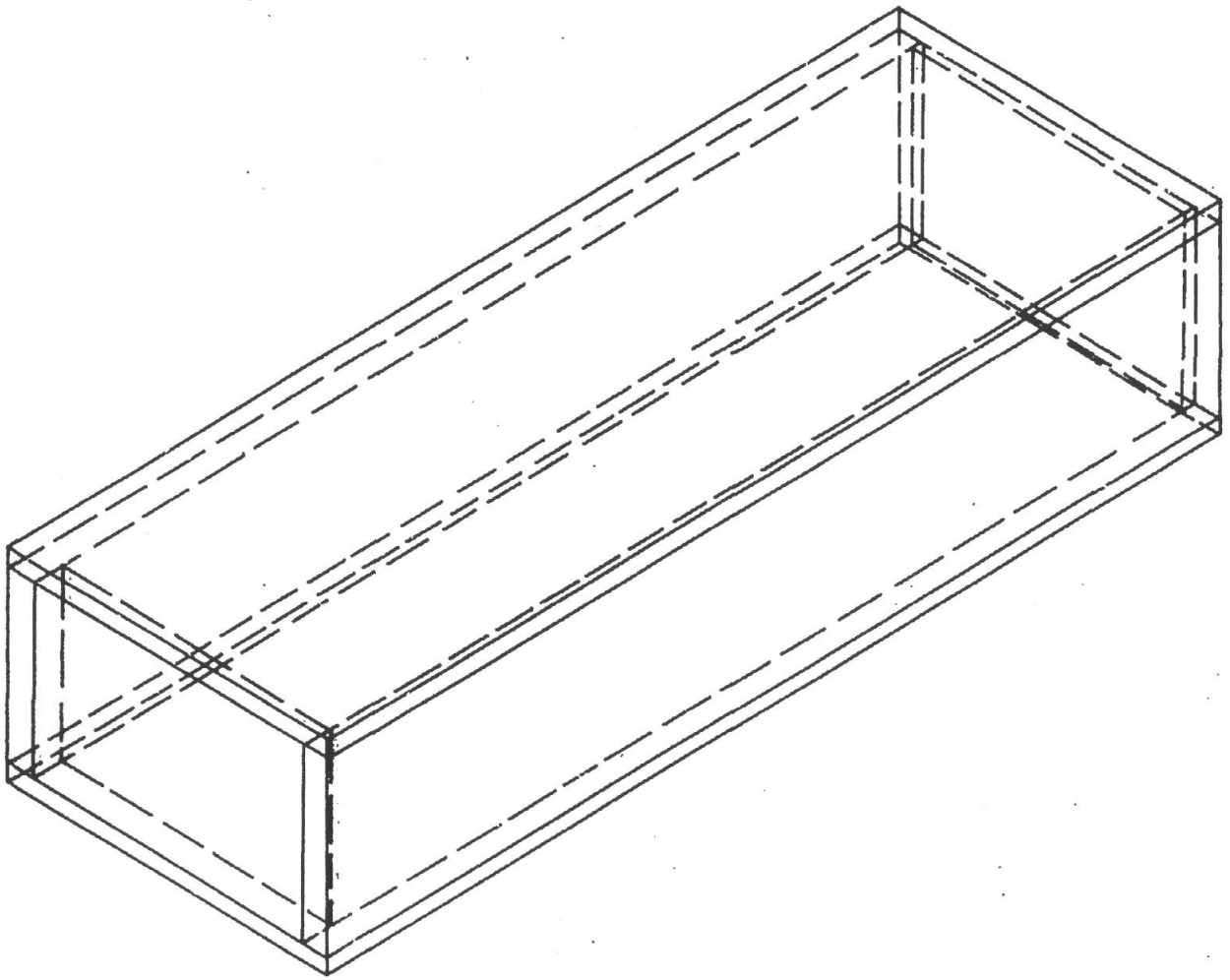


Fig. 3.6 - Outer Shell of Sample Vehicle

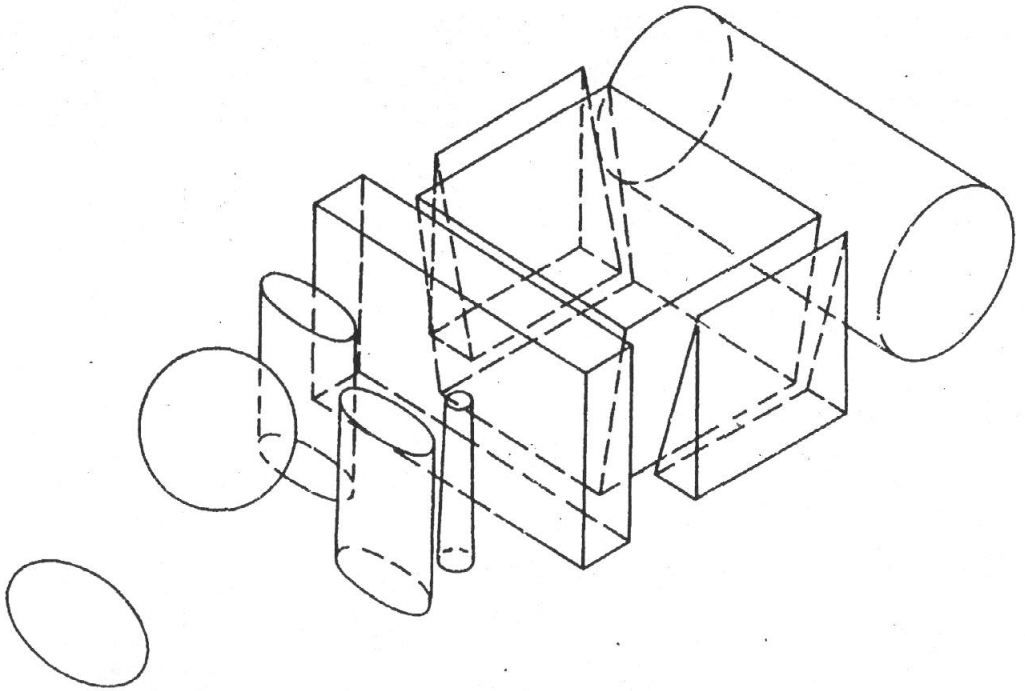


Fig. 3.7 - Interior Configuration of Sample Vehicle

THE DATE IS 07/03/67 ← (1)

THE TIME IS NOW 16:58:14.541E-01 ← (2) THIS IS A READING ONLY

TEST PROBLEM
NO. OF RECTANGULAR PARALLELPIPIPED 1
NO. OF TRIPLETS 1
NO. OF SCALARS 1
NO. OF BODIES 18
MAX. NO. OF REGIONS 15
SCALE FACTOR IS, 1.00000000E+00 ← (3)

RECTANGULAR PARALLELPIPIPED INPUT ← (4)
1 -2.54000000E+03 2.54000000E+03 -2.54000000E+03 2.54000000E+03 -2.54000000E+03 2.54000000E+03

TRIPLET DATA
1 0. -0. -0. ← (5)

SCALAR DATA
1 0.

DESCRIPTION OF BODIES

2 ARB 2
8.00000000E+01 3.60000000E+01 -2.30000000E+01 8.00000000E+01 -3.60000000E+01 -2.30000000E+01
-1.20000E+02 -3.60000E+01 -2.30000E+01 -1.20000E+02 3.60000E+01 -2.30000E+01
8.00000E+01 3.60000E+01 -2.60000E+01 8.00000E+01 -3.60000E+01 -2.60000E+01
-1.20000E+02 -3.60000E+01 -2.60000E+01 -1.20000E+02 3.60000E+01 -2.60000E+01
1 2 3 4 1 5 6 2 1 4 8 5 7 8 5 6 7 8 4 3 7 3 2 6

3 ARB 3
8.00000000E+01 -3.10000000E+01 1.40000000E+01 8.00000000E+01 -3.60000000E+01 1.40000000E+01
-1.20000E+02 -3.60000E+01 1.40000E+01 -1.20000E+02 -3.10000E+01 1.40000E+01
8.00000E+01 -3.10000E+01 -2.30000E+01 8.00000E+01 -3.60000E+01 -2.30000E+01
-1.20000E+02 -3.60000E+01 -2.30000E+01 -1.20000E+02 -3.10000E+01 -2.30000E+01
1 2 3 4 1 5 6 2 1 4 8 5 7 8 5 6 7 8 4 3 7 3 2 6

4 ARB 4
8.00000000E+01 3.60000000E+01 1.40000000E+01 8.00000000E+01 3.10000000E+01 1.40000000E+01
-1.20000E+02 3.10000E+01 1.40000E+01 -1.20000E+02 3.60000E+01 1.40000E+01
8.00000E+01 3.60000E+01 -2.30000E+01 8.00000E+01 3.10000E+01 -2.30000E+01
-1.20000E+02 3.10000E+01 -2.30000E+01 -1.20000E+02 3.60000E+01 -2.30000E+01
1 2 3 4 1 5 6 2 1 4 8 5 7 8 5 6 7 8 4 3 7 3 2 6

5 ARB 5
8.00000000E+01 3.10000000E+01 1.40000000E+01 8.00000000E+01 -3.10000000E+01 1.40000000E+01
7.40000E+01 -3.10000E+01 1.40000E+01 7.40000E+01 3.10000E+01 1.40000E+01
8.00000E+01 3.10000E+01 -2.30000E+01 8.00000E+01 -3.10000E+01 -2.30000E+01
7.40000E+01 -3.10000E+01 -2.30000E+01 7.40000E+01 3.10000E+01 -2.30000E+01
1 2 3 4 1 5 6 2 1 4 8 5 7 8 5 6 7 8 4 3 7 3 2 6

6 ARB 6
-1.18000000E+02 3.10000000E+01 1.40000000E+01 -1.18000000E+02 -3.10000000E+01 1.40000000E+01
-1.20000E+02 -3.10000E+01 1.40000E+01 -1.20000E+02 3.10000E+01 1.40000E+01
-1.18000E+02 3.10000E+01 -2.30000E+01 -1.18000E+02 -3.10000E+01 -2.30000E+01
-1.20000E+02 -3.10000E+01 -2.30000E+01 -1.20000E+02 3.10000E+01 -2.30000E+01
1 2 3 4 1 5 6 2 1 4 8 5 7 8 5 6 7 8 4 3 7 3 2 6

7 ARB 7
7.40000000E+01 3.10000000E+01 1.40000000E+01 7.40000000E+01 -3.10000000E+01 1.40000000E+01
-1.18000E+02 -3.10000E+01 1.40000E+01 -1.18000E+02 3.10000E+01 1.40000E+01
7.40000E+01 3.10000E+01 -2.30000E+01 7.40000E+01 -3.10000E+01 -2.30000E+01
-1.18000E+02 -3.10000E+01 -2.30000E+01 -1.18000E+02 3.10000E+01 -2.30000E+01
1 2 3 4 1 5 6 2 1 4 8 5 7 8 5 6 7 8 4 3 7 3 2 6

8 RCC 6
-1.00000000E+02 3.10000000E+01 +5.00000000E+00 0. -6.20000000E+01 0.
1.70000E+01

9 RAA 9
-8.00000000E+01 3.00000000E+01 -2.20000000E+01 0. 0. 3.40000000E+01
0. -9.00000E+00 0. 3.30000E+01 0. 0.

10 RAA 10
-8.00000000E+01 -3.00000000E+01 -2.20000000E+01 0. 0. 3.40000000E+01
0. 9.00000E+00 0. 3.30000E+01 0. 0.

11 ARB 11
-8.00000000E+01 1.90000000E+01 -2.20000000E+01 -8.00000000E+01 -1.90000000E+01 -2.20000000E+01
-8.00000E+01 -2.50000E+01 1.20000E+01 -8.00000E+01 2.50000E+01 1.20000E+01

← (6)

```
                                -3.70000E+01   1.90000E+01   -2.20000E+01   -3.70000E+01   -1.90000E+01   -2.20000E+01
                                -3.70000E+01   -2.50000E+01   1.20000E+01   -3.70000E+01   2.50000E+01   1.20000E+01
                                1 2 3 4   5 6 7 8   2 3 7 8   1 4 8 5   8 4 3 7   1 2 6 5
     121      BOX   14          -3.20000000E+01  3.10000000E+01 -2.30000000E+01 -8.60000000E+01  0.            0.
                                 0.            -6.20000E+01   0.             0.             0.            3.70000E+01
     131      BOX   15          -3.20000000E+01  3.10000000E+01 -2.30000000E+01  1.00000000E+00  0.            0.
                                 0.            -5.20000E+01   0.             0.             0.            3.70000E+01
     141      REC   14           8.00000000E+00  1.60000000E+01 -1.80000000E+01  0.             0.            3.0000000E+01
                                 0.             1.00000E+01   0.             4.00000E+00    0.            0.
     151      REC   15          -8.00000000E+00 -1.90000000E+01 -1.80000000E+01  0.             0.            3.00000000E+01
                                 0.             1.00000E+01   0.             4.00000E+00    0.            0.
     161      SPH   16           1.60000000E+01 -1.50000000E+01 -2.00000000E+00  1.50000000E+01
     171      ELL   17           6.12000000E+01  1.20000000E+01 -9.00000000E+00  6.10000000E+01 -1.20000000E+01 -9.00000000E+00
                                 3.00000E+01
     181      TRC   18          -4.00000000E+00  1.90000000E+01 -1.80000000E+01  0.             0.            3.00000000E+01
                                 3.20000E+00    2.20000E+00
     191      ARB    1           8.00000000E+01  3.60000000E+01  1.80000000E+01  8.00000000E+01 -3.60000000E+01  1.80000000E+01
                                -1.20000E+02   -3.60000E+01   1.80000E+01   -1.20000E+02   3.60000E+01   1.80000E+01
                                 8.00000E+01   3.60000E+01   1.40000E+01   8.00000E+01   -3.60000E+01   1.40000E+01
                                -1.20000E+02   -3.60000E+01   1.40000E+01   -1.20000E+02   3.60000E+01   1.40000E+01
                                1 2 3 4   1 5 6 2   1 4 8 5   7 8 5 6   7 8 4 3   7 3 2 6

 TRIPS=     2999R  LOCAL=     29997  LOOPY=          9  LREGUE=        326
```

(7)

REGION COMBINATION DATA

```
143
  1 (  R   19) (    -7) (   -12) (   -13) (  R    2) (    -7) (   -12) (   -13) (  R    3)
    (      -7) (   -12) (   -13) (    -8) (  R    4) (    -7) (   -12) (   -13) (    -8)
    (  R    5) (    -7) (  R    6) (    -7) (   -12) (    -0) (    -0) (    -0) (    -0)
  2 (      8) (    -3) (    -4) (    -0) (    -0) (    -0) (    -0) (    -0) (    -0)
  3 (  R   9) (  R   10) (    -0) (    -0) (    -0) (    -0) (    -0) (    -0) (    -0)
  4 (     11) (    -0) (    -0) (    -0) (    -0) (    -0) (    -0) (    -0) (    -0)
  5 (     12) (    -9) (   -10) (   -11) (    -8) (    -3) (    -4) (   -19) (    -2)
    (     -6) (   -13) (    -0) (    -0) (    -0) (    -0) (    -0) (    -0) (    -0)
  6 (     13) (    -3) (    -4) (   -12) (   -19) (    -2) (    -0) (    -0) (    -0)
  7 (     14) (    -0) (    -0) (    -0) (    -0) (    -0) (    -0) (    -0) (    -0)
  8 (     15) (    -0) (    -0) (    -0) (    -0) (    -0) (    -0) (    -0) (    -0)
  9 (     16) (    -0) (    -0) (    -0) (    -0) (    -0) (    -0) (    -0) (    -0)
 10 (     17) (    -0) (    -0) (    -0) (    -0) (    -0) (    -0) (    -0) (    -0)
 11 (     18) (    -0) (    -0) (    -0) (    -0) (    -0) (    -0) (    -0) (    -0)
 12 (      7) (   -13) (   -12) (   -19) (    -2) (    -3) (    -4) (    -5) (   -14)
    (    -15) (   -16) (   -17) (   -18) (    -6) (    -0) (    -0) (    -0) (    -0)
 13 (     11) (    -2) (    -3) (    -4) (    -5) (    -6) (   -19) (    -0) (    -0)
```

(8)

```
TOTAL ROOM FOR GEOMETRY DATA   340
```

(9)

```
ENTER    1   429   429    13
LEAVE    1   411   410     1
ENTER    2   460   460     1
LEAVE    2   411   415     1     5     6    12    13
ENTER    3   461   461     1
LEAVE    3   416   421     1     2     5     6    12    13
ENTER    4   462   462     1
LEAVE    4   422   427     1     2     5     6    12    13
ENTER    5   463   463     1
LEAVE    5   428   430     1    12    13
ENTER    6   464   464     1
LEAVE    6   431   434     1     5    12    13
ENTER    7   465   466     1    12
LEAVE    7   436   435     1
```

(10)

```
ENTER     8   467   468     1     2
LEAVE     8   436   437     1     5
ENTER     9   469   469     3
LEAVE     9   438   439     3     5
ENTER    10   470   470     3
LEAVE    10   440   441     3     5
ENTER    11   471   471     4
LEAVE    11   442   442     5
ENTER    12   472   473     1     5
LEAVE    12   443   445     1     6    12
ENTER    13   474   475     1     6
LEAVE    13   446   448     1     5    12
ENTER    14   476   476     7
LEAVE    14   449   449    12
ENTER    15   477   477     8
LEAVE    15   450   450    12
ENTER    16   478   478     9
LEAVE    16   451   451    12
ENTER    17   479   479    10
LEAVE    17   452   452    12
ENTER     6   480   480    11
LEAVE    18   453   453    12
ENTER    19   481   481     1
LEAVE    19   454   458     1     5     6    12    13
```

BEGIN EZGEOM OUTPUT

```
   1     9     0     7  0.          $     2     9     0     8  0.          $     3     9     0     7  0.          $
   4     9     0     8  0.          $     5     9     0     7  0.          $     6     9     0     8  0.          $
   7  1279 32767 32767 -2.5400E+03  $     8  3148     0     0  2.5400E+03  $     9   459   411     0  0.          $
  10     1     0     0  0.          $    11   460   411     8  0.          $    12     1     5    47  0.          $
  13   461   416     8  0.          $    14     1     6    53  0.          $    15   462   423     8  0.          $
  16     1     6    59  0.          $    17   463   428     8  0.          $    18     1     3    65  0.          $
  19   464   431     8  0.          $    20     1     4    71  0.          $    21   465   435     8  0.          $
  22     2     1    77  0.          $    23   467   436     3  0.          $    24     2     2    83  0.          $
  25   469   438     7  0.          $    26     1     2    84  0.          $    27   470   440     7  0.          $
  28     1     2    86  0.          $    29   471   442     8  0.          $    30     1     1    88  0.          $
  31   472   443     1  0.          $    32     2     3    94  0.          $    33   474   446     1  0.          $
  34     2     3    96  0.          $    35   476   449     4  0.          $    36     1     1    98  0.          $
  37   477   450     4  0.          $    38     1     1   100  0.          $    39   478   451     2  0.          $
  40     1     1   102  0.          $    41   479   452     6  0.          $    42     1     1   103  0.          $
  43   480   453     5  0.          $    44     1     1   104  0.          $    45   481   454     8  0.          $
  46     1     5   106  0.          $    47     1   318   319  0.          $    48     1   314   315  0.          $
  49     1   310   311  0.          $    50     1   306   307  0.          $    51     1   302   303  0.          $
  52     1   310   299  0.          $    53     1   295   296  0.          $    54     1   291   292  0.          $
  55     1   287   288  0.          $    56     1   283   284  0.          $    57     1   279   280  0.          $
  58     1   275   276  0.          $    59     1   295   296  0.          $    60     1   291   292  0.          $
  61     1   275   288  0.          $    62     1   283   284  0.          $    63     1   279   280  0.          $
  64     1   287   276  0.          $    65     1   271   272  0.          $    66     1   267   268  0.          $
  67     1   263   264  0.          $    68     1   259   260  0.          $    69     1   255   256  0.          $
  70     1   263   252  0.          $    71     1   248   249  0.          $    72     1   247   268  0.          $
  73     1   256   244  0.          $    74     1   240   241  0.          $    75     1   239   256  0.          $
  76     1   256   236  0.          $    77     1   232   233  0.          $    78     1   231   268  0.          $
  79     1   227   228  0.          $    80     1   223   224  0.          $    81     1   222   256  0.          $
  82     1   227   219  0           $    83   216   213   212  0.          $    84   209   206   203  0.          $
  85     0     0   200  0.          $    86   197   206   194  0.          $    87     0     0   200  0.          $
```

424	0	0	5	0.	$	425	0	0	6	0.	$	426	0	0	12	0.	$
427	0	0	13	0.	$	428	0	0	1	0.	$	429	0	0	12	0.	$
430	0	0	13	0.	$	431	0	0	1	0.	$	432	0	0	5	0.	$
433	0	0	12	0.	$	434	0	0	13	0.	$	435	0	0	1	0.	$
436	0	0	1	0.	$	437	0	0	5	0.	$	438	0	0	3	0.	$
439	0	0	5	0.	$	440	0	0	3	0.	$	441	0	0	5	0.	$
442	0	0	5	0.	$	443	0	0	1	0.	$	444	0	0	6	0.	$
445	0	0	12	0.	$	446	0	0	1	0.	$	447	0	0	5	0.	$
448	0	0	12	0.	$	449	0	0	12	0.	$	450	0	0	12	0.	$
451	0	0	12	0.	$	452	0	0	12	0.	$	453	0	0	12	0.	$
454	0	0	1	0.	$	455	0	0	5	0.	$	456	0	0	6	0.	$
457	0	0	12	0.	$	458	0	0	13	0.	$	459	0	0	13	0.	$
460	0	0	1	0.	$	461	0	0	1	0.	$	462	0	0	1	0.	$
463	0	0	1	0.	$	464	0	0	1	0.	$	465	0	0	1	0.	$
466	0	0	12	0.	$	467	0	0	1	0.	$	468	0	0	2	0.	$
469	0	0	3	0.	$	470	0	0	3	0.	$	471	0	0	4	0.	$
472	0	0	1	0.	$	473	0	0	5	0.	$	474	0	0	1	0.	$
475	0	0	6	0.	$	476	0	0	7	0.	$	477	0	0	8	0.	$
478	0	0	9	0.	$	479	0	0	10	0.	$	480	0	0	11	0.	$
481	0	0	1	0.	$	482	0	0	0	0.	$	483	0	0	0	0.	$

REGION	CODE	TYPE	ALPHA
1	101	20	
2	703	-0	
3	220	-0	
4	761	-0	
5	998	2	
6	333	-0	
7	41	-0	
8	1	-0	
9	7	-0	
10	472	-0	
11	17	-0	
12	999	2	
13	987	1	
-0	-0	-0	

(12)

NO. OF ASPECT ANGLES IS 1

THE TIME IS NOW 2.0[illegible]32/869E+00 THIS IS A READING ONLY

CENTER VALUES FOR TANK ASSUMED TO BE 0. 0. 0.

(13)

NX 16 NY 11 KSTART 13 KENC 13

AZIMUTH 0. ELEVATION 0. DIST 2.00000000E+02 PI

AZIMUTH 0. ELEVATION 0. DIST 2.00000000E+02 PI 3.14159000E+00

STARTING FROM GRID SQUARE NUMBER 1

RAY	1	1	3.5[illegible]E+01	1.56[illegible]E+01	2.0000E+02	3.5000E+01	1.8600E+01	-1.0000E+00	-0. -0.
RAY	1	2	3.[illegible]E+01	2.10[illegible]E+01	2.0000E+02	3.1000E+01	2.1000E+01	-1.0000E+00	-0. -0.
RAY	1	3	[illegible].[illegible]E+01	2.14[illegible]E+01	2.0000E+02	2.6600E+01	2.1400E+01	-1.0000E+00	-0. -0.
RAY	1	4	[illegible].[illegible]E+01	2.14[illegible]E+01	2.0000E+02	2.3400E+01	2.1400E+01	-1.0000E+00	-0. -0. ← (14)
RAY	1	5	[illegible].[illegible]E+01	2.14[illegible]E+01	2.0000E+02	1.9400E+01	2.1400E+01	-1.0000E+00	-0. -0.

```
RAY          1          6  1.7400E+01  1.9400E+01  2.0000E+02  1.7400E+01  1.9400E+01 -1.0000E+00 -0.          -0.
RAY          1          7  1.0200E+01  1.9000E+01  2.0000E+02  1.0200E+01  1.9000E+01 -1.0000E+00 -0.          -0.
RAY          1          8  7.0000E+00  2.0200E+01  2.0000E+02  7.0001E+00  2.0200E+01 -1.0000E+00 -0.          -0.
RAY          1          9  5.8000E+00  2.0600E+01  2.0000E+02  5.7999E+00  2.0600E+01 -1.0000E+00 -0.          -0.
RAY          1         10 -1.4000E+00  1.8200E+01  2.0000E+02 -1.3999E+00  1.8200E+01 -1.0000E+00 -0.          -0.
RAY          1         11 -3.8000E+00  2.1800E+01  2.0000E+02 -3.8000E+00  2.1800E+01 -1.0000E+00 -0.          -0.
RAY          1         12 -6.6000E+00  2.1800E+01  2.0000E+02 -6.6001E+00  2.1800E+01 -1.0000E+00 -0.          -0.
RAY          1         13 -1.1000E+01  2.1400E+01  2.0000E+02 -1.1000E+01  2.1400E+01 -1.0000E+00 -0.          -0.
RAY          1         14 -1.5800E+01  2.0600E+01  2.0000E+02 -1.5800E+01  2.0600E+01 -1.0000E+00 -0.          -0.
RAY          1         15 -1.8200E+01  2.0200E+01  2.0000E+02 -1.8200E+01  2.0200E+01 -1.0000E+00 -0.          -0.
RAY          1         16 -2.5400E+01  2.1800E+01  2.0000E+02 -2.5400E+01  2.1800E+01 -1.0000E+00 -0.          -0.
RAY          1         17 -2.8600E+01  1.8200E+01  2.0000E+02 -2.8600E+01  1.8200E+01 -1.0000E+00 -0.          -0.
RAY          1         18 -3.0600E+01  2.0600E+01  2.0000E+02 -3.0600F+01  2.0600E+01 -1.0000F+00 -0.          -0.
RAY          2          1  3.7000E+01  1.5000E+01  2.0000E+02  3.7000F+01  1.5000E+01 -1.0000E+00 -0.          -0.
RAY          2          2  3.0600E+01  1.5000E+01  2.0000E+02  3.0600E+01  1.5000E+01 -1.0000E+00 -0.          -0.
 30.6   15.0 12   80. 0 -120. 0 0 1 1 0  1   5  4
101 200. 0 200. 0    0.0  9   0. 0    0   0. 0   0. 0   0.0 0   0. 0   8  4 00 1
RAY          2          3  2.7800E+01  1.5000E+01  2.0000E+02  2.7800F+01  1.5000E+01 -1.0000E+00 -0.          -0.
 27.8   15.0 42   80. 0 -120. 0 0 1 1 0  1   7  4
101 200. 0 200. 0    0.0  9   0. 0    0   0. 0   0. 0   0.0 0   0. 0   7  4 00 1
RAY          2          4  2.5400E+01  1.6600E+01  2.0000E+02  2.5400E+01  1.6600E+01 -1.0000E+00 -0.          -0.
 25.4   16.6 86   80. 0 -120. 0 0 1 1 0  1   6  4
101 200. 0 200. 0    0.0  9   0. 0    0   0. 0   0. 0   0.0 0   0. 0   6  4 00 1
RAY          2          5  2.1800E+01  1.5800E+01  2.0000E+02  2.1800E+01  1.5800E+01 -1.0000E+00 -0.          -0.
 21.8   15.8 94   80. 0 -120. 0 0 1 1 0  1   5  4
101 200. 0 200. 0    0.0  9   0. 0    0   0. 0   0. 0   0.0 0   0. 0   5  4 00 1
RAY          2          6  1.5400E+01  1.7000E+01  2.0000E+02  1.5400E+01  1.7000E+01 -1.0000E+00 -0.          -0.
 15.4   17.0 37   80. 0 -120. 0 0 1 1 0  1   4  4
101 200. 0 200. 0    0.0  9   0. 0    0   0. 0   0. 0   0.0 0   0. 0   4  4 00 1
RAY          2          7  1.0600E+01  1.6600E+01  2.0000E+02  1.0600F+01  1.6600E+01 -1.0000E+00 -0.          -0.
 10.6   16.6 16   80. 0 -120. 0 0 1 1 0  1   3  4
101 200. 0 200. 0    0.0  9   0. 0    0   0. 0   0. 0   0.0 0   0. 0   3  4 00 1
RAY          2          8  8.6000E+00  1.4600E+01  2.0000E+02  8.6000F+00  1.4600E+01 -1.0000E+00 -0.          -0.
  8.6   14.6 61   80. 0 -120. 0 0 1 1 0  1   2  4
101 200. 0 200. 0    0.0  9   0. 0    0   0. 0   0. 0   0.0 0   0. 0   2  4 00 1
RAY          2          9  5.8000E+00  1.4200E+01  2.0000E+02  5.7999E+00  1.4200E+01 -1.0000E+00 -0.          -0.
  5.8   14.2 90   80. 0 -120. 0 0 1 1 0  1   1  4
101 200. 0 200. 0    0.0  9   0. 0    0   0. 0   0. 0   0.0 0   0. 0   1  4 00 1
RAY          2         10  1.0000E+00  1.6200E+01  2.0000E+02  1.0001E+00  1.6200E+01 -1.0000E+00 -0.          -0.
  1.0   16.2 75   80. 0 -120. 0 0 1 1 0  1   0  4
101 200. 0 200. 0    0.0  9   0. 0    0   0. 0   0. 0   0.0 0   0. 0   0  4 00 1
RAY          2         11 -2.2000E+00  1.4600E+01  2.0000E+02 -2.2001F+00  1.4600E+01 -1.0000E+00 -0.          -0.


  2.2   14.6 91   80. 0 -120. 0 0 1 1 0  1 - 1  4
101 200. 0 200. 0    0.0  9  -0. 0    0   0. 0   0. 0   0.0 0   0. 0 - 1  4 00 1
RAY          2         12 -9.4000E+00  1.5800E+01  2.0000E+02 -9.3999F+00  1.5800E+01 -1.0000E+00 -0.          -0.


  9.4   15.8 14   80. 0 -120. 0 0 1 1 0  1 - 2  4
101 200. 0 200. 0    0.0  9   0. 0    0   0. 0   0. 0   0.0 0   0. 0 - 2  4 00 1
RAY          2         13 -1.1000E+01  1.6200E+01  2.0000E+02 -1.1000F+01  1.6200E+01 -1.0000F+00 -0.          -0.


 11.1   16.2 75   80. 0 -120. 0 0 1 1 0  1 - 3  4
101 200. 0 200. 0    0.0  9   0. 0    0   0. 0   0. 0   0.0 0   0. 0 - 3  4 00 1
```

```
RAY          2        14 -1.4490E+01  1.7400E+01  2.0000E+02 -1.4200E+01  1.7400E+01 -1.0000E+00 -0.          -0.

 14.2   17.4 98   80. 0 -120. 0 0 1 1 0  1 - 4  4
101 200. 0 200. 0    0.0  9   0. 0   0   0. 0   0. 0   0.0  0   0. 0 - 4  4 00 1
RAY          2        15 -1.9400E+01  1.5800E+01  2.0000E+02 -1.9400E+01  1.5800E+01 -1.0000E+00 -0.          -0.

 19.4   15.8 64   80. 0 -120. 0 0 1 1 0  1 - 5  4
101 200. 0 200. 0    0.0  9   0. 0   0   0. 0   0. 0   0.0  0   0. 0 - 5  4 00 1
RAY          2        16 -2.4200E+01  1.7400E+01  2.0000E+02 -2.4200E+01  1.7400E+01 -1.0000E+00 -0.          -0.

 24.2   17.4 48   80. 0 -120. 0 0 1 1 0  1 - 6  4
101 200. 0 200. 0    0.0  9   0. 0   0   0. 0   0. 0   0.0  0   0. 0 - 6  4 00 1
RAY          2        17 -2.7000E+01  1.4600E+01  2.0000E+02 -2.7000E+01  1.4600E+01 -1.0000E+00 -0.          -0.

 27.0   14.6 71   80. 0 -120. 0 0 1 1 0  1 - 7  4
101 200. 0 200. 0    0.0  9   0. 0   0   0. 0   0. 0   0.0  0   0. 0 - 7  4 00 1
RAY          2        18 -3.2200E+01  1.7400E+01  2.0000E+02 -3.2200E+01  1.7400E+01 -1.0000E+00 -0.          -0.

 32.2   17.4 48   80. 0 -120. 0 0 1 1 0  1 - 8  4
101 200. 0 200. 0    0.0  9   0. 0   0   0. 0   0. 0   0.0  0   0. 0 - 8  4 00 1
RAY          3         1  3.5800E+01  1.0200E+01  2.0000E+02  3.5800E+01  1.0200E+01 -1.0000E+00 -0.          -0.
 35.8   10.2 40   80. 0 -120. 0 0 1 1 0  1  9  3
101 200. 0 200. 0    0.0  9   0. 0   0   0. 0   0. 0   0.0  0   0. 0  9  3 00 1
RAY          3         2  3.1000E+01  1.1400E+01  2.0000E+02  3.1000E+01  1.1400E+01 -1.0000E+00 -0.          -0.
 31.0   11.4 23   80. 0 -120. 0 0 1 1 1  4  6  3
101   6. 0   6. 0    0.0  2 105. 0 333   1. 0   1. 0   0.0  5  63.52   8  3 00 2
703   8.95  34. 0   74.7  5  13.52 101    2. 0   2. 0   0.0  9   0. 0   8  3 00 4
RAY          3         3  2.7800E+01  1.3400E+01  2.0000E+02  2.7800E+01  1.3400E+01 -1.0000E+00 -0.          -0.
 27.8   13.4 48   80. 0 -120. 0 0 1 1 1  5  7  3
101   6. 0   6. 0    0.0  2 105. 0 333   1. 0   1. 0   0.0  5  86. 0   7  3 00 2
101   2. 0   2. 0    0.0  9   0. 0   0   0. 0   0. 0   0.0  0   0. 0   7  3 00 3
RAY          3         4  2.2200E+01  1.1800E+01  2.0000E+02  2.2200E+01  1.1800E+01 -1.0000E+00 -0.          -0.
 22.2   11.8  4   80. 0 -120. 0 0 1 1 1  6  6  3
101   6. 0   6. 0    0.0  2  82.86  41   6.26   7.13  17.5  2  35.86   6  3 00 2
333   1. 0   1. 0    0.0  5   5. 0 701   43. 0   43. 0   0.0  5  17.40   6  3 00 4
703   5.20  34. 0   81.2  5  15.40 101    2. 0   2. 0   0.0  9   0. 0   6  3 00 6
RAY          3         5  1.8200E+01  1.0200E+01  2.0000E+02  1.8200E+01  1.0200E+01 -1.0000E+00 -0.          -0.
 18.2   10.2  0   80. 0 -120. 0 0 1 1 1  7  5  3
101   6. 0   6. 0    0.0  2  82.10  41   7.80   7.89   5.1  2   5.67   5  3 00 2
 17   4.86   5.13  18.3  2  24.57 333    1. 0   1. 0   0.0  5   5. 0   5  3 00 4
701  43. 0  43. 0    0.0  5  12.39 703   15.23  34. 0  63.4  5  10.39   5  3 00 6
101   2. 0   2. 0    0.0  9   0. 0   0   0. 0   0. 0   0.0  0   0. 0   5  3 00 7
RAY          3         6  1.7400E+01  1.0200E+01  2.0000E+02  1.7400E+01  1.0200E+01 -1.0000E+00 -0.          -0.
 17.4   10.2 80   80. 0 -120. 0 0 1 1 1  7  4  3
101   6. 0   6. 0    0.0  2  82. 8  41   7.92   7.95   3.2  2   4. 4   4  3 00 2
 17   4. 8   5.13  38.7  2  25. 0 333    1. 0   1. 0   0.0  5   5. 0   4  3 00 4
701  43. 0  43. 0    0.0  5  12.39 703   15.23  34. 0  63.4  5  10.39   4  3 00 6
101   2. 0   2. 0    0.0  9   0. 0   0   0. 0   0. 0   0.0  0   0. 0   4  3 00 7
RAY          3         7  1.3800E+01  1.3410E+01  2.0000E+02  1.3800E+01  1.3400E+01 -1.0000E+00 -0.          -0.
 13.8   13.4 98   80. 0 -120. 0 0 1 1 1  5  3  3
101   6. 0   6. 0    0.0  2 105. 0 333   1. 0   1. 0   0.0  5  86. 0   3  3 00 2
```

```
101   2. 0   2. 0    0.0  9   0. 0   0   0. 0   0. 0   0.0  0   0. 0   3- 5 00 5
RAY         11          8  9.4000E+00 -1.9400E+01  2.0000E+02  9.4000E+00 -1.9400E+01 -1.0000E+00 -0.          -0.
  9.4 - 19.4 86   80. 0 -120. 0 0 1 1 1  5   2- 5
101   6. 0   6. 0    0.0  2 105. 0 333   1. 0   1. 0   0.0  5   5. 0   2- 5 00 2
701  43. 0  43. 0    0.0  5  10.96 703  18. 7  34. 0  57.8  5   8.96   2- 5 00 4
101   2. 0   2. 0    0.0  9   0. 0   0   0. 0   0. 0   0.0  0   0. 0   2- 5 00 5
RAY         11          9  5.8000E+00 -2.1400E+01  2.0000E+02  5.8001E+00 -2.1400E+01 -1.0000E+00 -0.          -0.
  5.8 - 21.4 91   80. 0 -120. 0 0 1 1 1  5   1- 5
101   6. 0   6. 0    0.0  2 105. 0 333   1. 0   1. 0   0.0  5   5. 0   1- 5 00 2
701  43. 0  43. 0    0.0  5  15.52 703   8.95  34. 0  74.7  5  13.52   1- 5 00 4
101   2. 0   2. 0    0.0  9   0. 0   0   0. 0   0. 0   0.0  0   0. 0   1- 5 00 5
RAY         11         10  1.8000E+00 -1.8600E+01  2.0000E+02  1.7999E+00 -1.8600E+01 -1.0000E+00 -0.          -0.
  1.8 - 18.6 98   80. 0 -120. 0 0 1 1 1  5   0- 5
101   6. 0   6. 0    0.0  2 105. 0 333   1. 0   1. 0   0.0  5   5. 0   0- 5 00 2
701  43. 0  43. 0    0.0  5   9.80 703  20.40  34. 0  53.1  5   7.80   0- 5 00 4
101   2. 0   2. 0    0.0  9   0. 0   0   0. 0   0. 0   0.0  0   0. 0   0- 5 00 5
RAY         11         11 -3.4000E+00 -1.9400E+01  2.0000E+02 -3.4000E+00 -1.9400E+01 -1.0000E+00 -0.          -0.

  3.4 - 19.4 66   80. 0 -120. 0 0 1 1 1  5 - 1- 5
101   6. 0   6. 0    0.0  2 105. 0 333   1. 0   1. 0   0.0  5   5. 0 - 1- 5 00 2
701  43. 0  43. 0    0.0  5  10.96 703  18. 7  34. 0  57.8  5   8.96 - 1- 5 00 4
101   2. 0   2. 0    0.0  9   0. 0   0   0. 0   0. 0   0.0  0   0. 0 - 1- 5 00 5
RAY         11         12 -6.2000E+00 -2.1800E+01  2.0000E+02 -6.2001E+00 -2.1800E+01 -1.0000E+00 -0.          -0.

  6.2 - 21.8 90   80. 0 -120. 0 0 1 1 1  5 - 2- 5
101   6. 0   6. 0    0.0  2 105. 0 333   1. 0   1. 0   0.0  5   5. 0 - 2- 5 00 2
701  43. 0  43. 0    0.0  5  17.40 703   5.20  34. 0  81.2  5  15.40 - 2- 5 00 4
101   2. 0   2. 0    0.0  9   0. 0   0   0. 0   0. 0   0.0  0   0. 0 - 2- 5 00 5
RAY         11         13 -1.1000E+01 -2.0600E+01  2.0000E+02 -1.1000E+01 -2.0600E+01 -1.0000E+00 -0.          -0.

 11.0 - 20.6 73   80. 0 -120. 0 0 1 1 1  5 - 3- 5
101   6. 0   6. 0    0.0  2 105. 0 333   1. 0   1. 0   0.0  5   5. 0 - 3- 5 00 2
701  43. 0  43. 0    0.0  5  13.24 703  13.51  34. 0  66.5  5  11.24 - 3- 5 00 4
101   2. 0   2. 0    0.0  9   0. 0   0   0. 0   0. 0   0.0  0   0. 0 - 3- 5 00 5
RAY         11         14 -1.5000E+01 -2.1000E+01  2.0000E+02 -1.5000E+01 -2.1000E+01 -1.0000E+00 -0.          -0.

 15.0 - 21.0 72   80. 0 -120. 0 0 1 1 1  5 - 4- 5
101   6. 0   6. 0    0.0  2 105. 0 333   1. 0   1. 0   0.0  5   5. 0 - 4- 5 00 2
701  43. 0  43. 0    0.0  5  14.26 703  11.49  34. 0  70.2  5  12.26 - 4- 5 00 4
101   2. 0   2. 0    0.0  9   0. 0   0   0. 0   0. 0   0.0  0   0. 0 - 4- 5 00 5
RAY         11         15 -2.1400E+01 -1.8600E+01  2.0000E+02 -2.1400E+01 -1.8600E+01 -1.0000E+00 -0.          -0.

 21.4 - 18.6 78   80. 0 -120. 0 0 1 1 1  4 - 5- 5
101   6. 0   6. 0    0.0  2 105. 0 333   1. 0   1. 0   0.0  5  57.80 - 5- 5 00 2
703  20.40  34. 0   53.1  5   7.80 101   2. 0   2. 0   0.0  0   0. 0 - 5- 5 00 4
RAY         11         16 -2.4200E+01 -2.1000E+01  2.0000E+02 -2.4200E+01 -2.1000E+01 -1.0000E+00 -0.          -0.

 24.2 - 21.0 42   80. 0 -120. 0 0 1 1 1  5 - 6- 5
101   6. 0   6. 0    0.0  2 105. 0 333   1. 0   1. 0   0.0  5  45. 0 - 6- 5 00 2
250  33. 0  33. 0    0.0  5  14.26 703  11.49  34. 0  70.2  5  12.26 - 6- 5 00 4
```

```
101    2. 0    2. 0    0:0  9    0. 0    0    0. 0    0. 0    0.0  0    0. 0 - 6- 5 00 5
RAY          11          17 -2.9600E+01 -2.0600E+01  2.0000E+02 -2.9800E+01 -2.0600E+01 -1.0000E+00 -0.          -0.


 29.8 - 20.6  3    80. 0 -120. 0 0 1 1 1  5 - 7- 5
101    6. 0    6. 0    0:0  2 105. 0 333    1. 0    1. 0    0.0  5  15. 0 - 7- 5 00 2
250   33. 0   33. 0    0:0  5  13.24 703   13.51   34. 0   66.5  5  11.24 - 7- 5 00 4
101    2. 0    2. 0    0:0  9    0. 0    0    0. 0    0. 0    0.0  0    0. 0 - 7- 5 00 5
RAY          11          18 -3.2200E+01 -1.8600E+01  2.0000E+02 -3.2200E+01 -1.8600E+01 -1.0000E+00 -0.          -0.


 32.2 - 18.6 48    80. 0 -120. 0 0 1 1 0  1 - 8- 5
101  200. 0  200. 0    0:0  9    0. 0    0    0. 0    0. 0    0.0  0    0. 0 - 8- 5 00 1
THE TIME IS NOW      1.505020E+01     THIS IS A READING ONLY
END OF CASE     1
```

4. DESCRIPTION OF THE SAM-C MONTE CARLO PROGRAM

INTRODUCTION

The SAM-C program described below is based in large measure on the UNC-SAM-2 Monte Carlo program prepared by United Nuclear Corporation (with MAGI as subcontractor) for Aberdeen Proving Ground.* The primary difference between UNC-SAM-2 and SAM-C is the geometrical description technique employed. SAM-C uses combinatorial geometry and is therefore capable of representing more complex assemblies. To use this geometry capability a number of modifications were made in the logic and storage requirements of UNC-SAM-2. In addition, a ray-tracing volume calculation routine was added since, for many of the shapes produced by the combinatorial geometry, it is impractical to determine the volume analytically. All the nuclear interaction routines of UNC-SAM-2 are unchanged.

---

* Troubetzkoy, E.S.: UNC-SAM-2: A FORTRAN Monte Carlo Program Treating Time-Dependent Neutron and Photon Transport through Matter, UNC-5151 (Sept.1966).

4.1 GENERAL DISCUSSION

SAM-C is a Monte Carlo program, written in FORTRAN, which calculates the time-dependent transport of neutrons or gamma rays through matter. It is composed of a series of independent routines which perform the following four basic functions:

1. Process cross-section data
2. Process geometry data
3. Perform the transport calculation
4. Edit the results.

Basically, the program requires as input a geometry specification (see Section 2), the elemental composition of each region, and a specification of the location and time-energy-angular distribution of the radiation source. The program selects individual particles from the given source distribution and tracks them through a series of interactions within the geometry until such time as the particle history is terminated. The tracking of a particle can be terminated for any of the following reasons.

1. The energy of the particle after an interaction falls below a specified "cutoff energy."
2. The elapsed time spent by the particle in traversing the geometry exceeds a specified "cutoff time."

3. The particle escapes from the geometry (crosses an external boundary).

4. The particle is "killed." This procedure will be explained in the section dealing with the importance sampling techniques employed in the program.

For each region traversed by a given particle, the code computes the flux per unit time per unit energy as a function of energy and time. The flux contribution for a given particle is defined as its expected total path length contribution in a region divided by the volume of the region. Individual particle flux contributions are accumulated so that the end result of the tracking process is the total flux in each region in a specified group of energy and time bins. At the users' option the problem also can be made time independent. The code has the additional capability of being able to compute fluxes at specified points within the geometry, as well as in finite regions. The use of this option is discussed later in Section 4.2.4.

The above description of the SAM-C program is, of course, a very simplified view of the computational procedure. The following sections provide a more detailed, although nonmathematical, description of each part of the computation.

4.2 INPUT REQUIREMENTS

4.2.1 Cross Sections and Region Compositions

Generally, the user of the program will have at his disposal an Element Data Tape (EDT) which contains, for every isotope in the problem, a set of energy-dependent interaction cross sections. The user must then specify each of the material compositions appearing in the problem. A composition is defined in terms of atomic concentrations (in units of 10^{24} atoms/cm^3) of each isotope in the composition. These may be calculated from the expression:

10^{24} x Avogadro's number x mass density/atomic weight. For compounds or mixtures the concentration of each component must be specified. In addition, each composition must be identified by a composition number.

This input is processed in conjunction with the EDT by the BAND routine, which generates an Organized Date Tape (ODT). The ODT contains total macroscopic cross sections and absorption and scattering probabilities for each composition. During the tracking process this information is used to determine:

1. The probability that a particle has an interaction in a region of given composition.
2. The element with which the particle interacts.
3. The type of interaction (absorption, elastic scattering, etc.) occurring.

4. The energy and direction of the particle after interaction.

4.2.2 Geometry Input

The specification of the geometry has been treated in Section 2, where the point was made that the same geometry input can be used in both MAGIC and SAM-C. It should be noted, therefore, that if both codes are to be applied to a given vehicle (or any geometry, for that matter) important regions for the nuclear calculation (i.e., air, ground) should appear in the geometry specification.

4.2.3 Importance Sampling

A. General

Importance sampling or "weighting" provides the user with a powerful method of controlling the direction and/or energy of particles in the problem. The purpose of a particular problem, for example, may be to calculate the fast-neutron flux in a given region within the geometry. Under normal circumstances, the probability of a source neutron reaching that region at high energy may be quite small, requiring a vast number of source neutrons to be tracked before an adequate statistical estimate of the flux is obtained. However, with proper particle weighting the code can be made to concentrate only on those fast neutrons having the best chance of reaching the chosen region. Conversely, the code will spend little time tracking neutrons which are either traveling in the wrong direction or are at relatively low

energy.

The program determines the relative importance of a particle from a parameter called the weight. The total weight (W) of a particle is, in turn, determined from a combination of three quantities called region weight (W_R), angular weight (W_Ω), and energy weight (W_E), where $W = W_R \times W_\Omega \times W_E$. Values of W_R, W_Ω, and W_E are given as input. The following brief discussion should provide the user with a better insight into how these weights are actually used by the code.

A quantity F is assigned to each particle. The value of F is 1.0 for a source particle. The code calculates the probability that the particle will reach the boundary of the source region along its flight path without collision. This value is called F". The probability that a collision takes place in the region is then 1-F". The code picks a random number and with probability 1-F" creates a collision point inside the source region at a point picked from an exponential probability distribution. The source particle history is not terminated but continues to the boundary of the region. After going through the collision mechanics, a new particle (latent) will be started at the collision point after the original source particle has been completely tracked.

Suppose that the source particle is leaving region 1 where the weight is W_1 and entering region 2 where the weight is W_2. At the boundary, the ratio of weights W_1/W_2 is multiplied by F" and the particle is given a starting value of $F = F" W_1/W_2$ in region 2. The probability of the particle reaching the next boundary of region 2 uncollided is calculated and multiplied by F to obtain a new value of F". In region 2 a number of collision points approximately equal to F-F" will be produced. Notice that if W_2 is small compared to W_1, a large number of latent particles will be produced along the track of the source particle in region 2. Conversely if W_2 is large compared to W_1, the probability is high that no latents will be produced since both F and F" will be small compared to one. In fact, a parameter F_z is an input parameter to the program. If F, on entry into a new region, is lower than F_z, a random number between zero and one is picked. If the number is greater than F the history is terminated. If it is lower than F the history is continued with F set equal to one.

Thus by establishing weight sets properly, increased numbers of collisions can be forced to occur in important regions, and in addition, the original source particles will continue to propagate through the geometry.

Without going into detail it can be stated that a small value of F_z minimizes the number of kills (increases problem running time). A large value maximizes the kills (decreases

running time per history) but increases the variance (or % accuracy) of the answers, requiring more source particles to be run. The optimum value of F_Z will generally lie in the range from 0.01 to 0.1.

In order to facilitate input preparation, the three components of the total weight will now be discussed separately.

B. Region Weights

A region weight (W_R) must be specified for every region in the problem. Ordinarily, these weights are set up so that they gradually decrease as a particle proceeds from the source toward a region in which the flux is desired. Weights should gradually increase in regions which are located progressively further from the "important" regions. On the input forms the user must specify all values of W_R to be used in the problem. The order in which these values are entered determines their region weight number (i.e., the first value of W_R is assigned weight #1, the second value is weight #2, etc.). Then for each region, the weight number to be used in that region must be specified.

C. Angular Weights

By using angular weighting, it is possible to specify preferred directions for a particle, independent of the region location of the particle. The user first specifies the direction cosines (with respect to the coordinate axes of the problem) of one or more aiming angles. These vectors

serve as "zero directions" about which angular weights will be given. Each aiming angle is assigned a number. Next, a set of angular bins is specified between 0° and 180°, with the bin boundaries given in terms of their cosines. Thus, if one desires to specify four bins of equal angle, the cosines of 0°, 45°, 90°, 135°, and 180° should be entered. Then, one or more sets of angular weight values are given. For each set, a weight value (W_Ω) is specified for each angular bin. Each set is also assigned a number. Finally, for each region, the aiming angle number and the angular weight set number must be given. To illustrate how the code uses this information, assume that a given region has been assigned aiming angle #1 and angular weight set #2. A particle enters the region and the code determines that the particle is traveling at an angle θ with respect to aiming angle #1. The code then determines which angular bin encompasses θ, goes to angular weight set #2, and finds the value of W_Ω in that bin.

In general, as the particle direction (angular bin) becomes more important, the value of W_Ω assigned to that bin should decrease.

D. Energy Weighting

The use of energy weighting enables the user to instruct the code as to which particle energies are most important in a given problem. A set of energy bins is first given, where the bin boundaries are listed in decreasing

order. Then, one or more energy weight sets are specified, with each set being assigned a number. For each set an energy weight value W_E must be given for each energy bin. The energy weight set number corresponding to each region is then given. Assume, for example, that a particle of energy E is in a region which has been assigned weight set #1. The code first locates the energy bin which emcompasses E, refers to weight set #1, and determines the value of W_E which was given for that bin. In establishing the energy weights, the more important energies should generally have smaller W_E values than the less important energies.

E. Application of Weights to Tracking

As noted earlier, the total particle weight is the product of $W_R \times W_\Omega \times W_E$. The particle weight is used to determine the number of collisions that a particle will produce given that it has a specified energy and direction in a given physical region. By an appropriate choice of aiming angle and angular weights, particles heading downward can be caused to have more collisions than particles heading upward in the same region. Thus more computing time will be spent on the "important" downward directed particles and their descendents than on the "less important" upward directed particles.

F. Treatment of "Latent" Particles

If a collision does occur the program calculates the energy and direction of the particle emerging from the collision. The collided particle is stored in a latent storage table and will be picked up and followed as though it were a source particle at a later time. When it is started out as a real particle, it is assigned an F value equal to the ratio of the weight of the particle which went into collision to the weight of the particle emerging from collision. In general, these will be different due to differences in energy and direction of travel.

The program stores the information concerning latents in a table which can hold up to 100 latents. Prior to storage a test is made to see if the F of the latent exceeds the input value of F_z. If so, it is stored. If not, a Russian roulette calculation is performed, as previously discussed, and the latent is either eliminated or has its F set equal to 1.0.

If more than 100 latents are generated by a source particle, the program has a "squeeze" routine which reduces the number of latents in a statistically valid way.

Although the use of importance sampling may appear to be a rather complicated procedure, the user will generally find that after gaining a little experience with the code the process becomes relatively straightforward and easily applied. Certainly, the time spent in learning how to

properly apply this technique will be well worth it in the long run, since it enables complex, deep-penetration problems to be run in a reasonable amount of machine time. Appendix A contains a discussion of the theory of importance sampling with an example of how a set of weights is established.

4.2.4 Flux-at-a-Point

In certain problems it may be desired to calculate the flux at a particular point in the geometry. Since, during the ordinary tracking process, no particle can be expected to pass through a given point, SAM-C incorporates an independent method of estimating the flux at one or more specified points.

The program actually uses two independent computational methods: flux-at-a-point (FAP) and a statistical estimation method (SEM). FAP is applicable to detector points in a scattering medium but not in the immediate vicinity of the source region. SEM is applied to detectors outside the immediate vicinity of either the source region or a scattering material. The user must specify the total number of detector points to be evaluated and the number of points where the FAP method is to be used. The X, Y, Z coordinates of each point are then listed with the FAP detectors listed first, followed by the SEM detectors.

The choice of method to be employed is left to the user, but, in general, SEM is more rapid if the detector is located in a vacuum (or very low-density material) and not close to the source. FAP is preferred for detectors within a scattering material but not near the source. Neither method is recommended for a point near the source region, and in this case, the point should be approximated by a relatively small three-dimensional region.

4.2.5 Source Specification

The specification of the initial particle source provides the user with several options. These options are described briefly below.

A. Spatial Distribution

Sources may be generated in any number of regions. For each source region the "power density" (particles/volume) must be given. The user has the option of normalizing the problem to a unit source or to the total input power density. If the number of source regions is zero, the program will use an external source tape (one which has been generated from a previous problem).

B. Angular Distribution

Sources may be either isotropic or monodirectional but the same angular distribution must be used in all source regions. (It should be noted, however, that a source may be generated in a finite cone by specifying an isotropic distribution and using angular weights to kill particles

which are generated outside the desired cone.)

C. Energy Distribution

The code has built into it the Cranberg fission neutron spectrum. If this option is selected, no energy spectrum input is required. If an arbitrary spectrum is desired, the input must contain the desired energy mesh and the integrated source above each energy point (i.e., a table of E vs $\int_E^\infty$ S(E) dE is required).

D. Time Distribution

If a time-dependent problem is to be run, the user must supply a table of time values and the integrated source up to each time (i.e., t vs $\int_0^t$ S(t) dt). This input should be deleted from time-independent problems.

4.2.6 Time Dependence

SAM-C enables the user to compute particle fluxes as a function of time as well as energy and position. The user selects any desired time bin structure for the problem and enters the bin limits in consecutive order on the input forms, starting with the highest bin. Output fluxes will be given in this bin structure in the edit. A time-cutoff must also be specified, which instructs the code to cease tracking all particles which have "aged" beyond this cutoff. During the tracking process the code computes the flight time of a particle between collision points from its velocity (or energy). Interactions are assumed to occur instantaneously. By accumulating the flight times for each

particle, the code is capable of storing particle fluxes in the proper output time bin.

4.2.7 Output Energy Mesh

During tracking, the code stores fluxes in each region in a set of energy output bins specified by the user. The number and width of these bins are arbitrary, but they may not extend beyond the energy range for which cross-section data are available. The bin limits must be given consecutively in the input, starting with the highest energy. The upper and lower bin limits must be preceded by minus signs. The reason for this will be explained shortly. Care should be taken to insure that the upper energy bin limit is equal to or greater than the highest source energy to be generated in the problem. A cutoff energy is also specified, which instructs the code to cease tracking any particle which degrades below this energy. The user should be certain that the lowest energy bin limit is equal to or lower than the cutoff energy. In essence, there must be a bin available to store every possible energy in the problem.

Some calculations may require more computer storage than is available. This situation can be alleviated by using the "supergroup" option provided by the code. This option divides the overall energy range into smaller groups (called supergroups) and the code then treats each of these groups separately. In this manner, only the cross-section

data for that group currently being treated are stored in the memory and only those particles having energies in that group are tracked. When a particle degrades to a lower supergroup, its parameters are stored and its tracking is resumed only after all higher groups have been completed. The supergroup structure is defined by the user by placing a minus sign before those output energies he wishes to designate as supergroup limits. If this option is not desired, only the upper and lower energy bin limits require minus signs. This instructs the code to treat all energies as part of a single supergroup.

4.2.8 Response Functions

SAM-C provides response function options which allow the user to automatically transform particle fluxes into any desired flux-dependent quantity (dose, heat deposition, etc.). Assume, for example, that the dose is required in several regions. The user supplies, as input, a flux-to-dose conversion factor as a function of energy and the numbers of the regions where the dose is desired. For each region, the code multiplies the flux $\phi(E)$ in each energy bin by the corresponding conversion factor $C(E)$ and integrates over energy. Thus

$$\text{DOSE} = \sum_{0}^{\text{EMAX}} \left\{ \phi(E) \times C(E) \times dE \right\} .$$

Two calculation options are provided as described on the following page.

1. Built-In Response

This option allows for a single response calculation for all flux regions. The calculation is performed before the flux edit.

2. Multiple Response

This option allows for the calculation of several responses through the use of the aggregate tape. At the completion of the Monte-Carlo problem the flux results will be stored, by aggregate, on the aggregate tape. This tape may be processed at a later time for as many responses as desired.

4.2.9 Transmission and Escape Regions

A transmission region has the property such that when a particle enters it, the tracking of that particle is terminated and all of its parameters (X, Y, Z coordinates, energy, etc.) are stored on a magnetic tape called the interaction tape. This tape then can be used to generate a source tape composed of particles entering the transmission region. In general, a transmission region is used when it is desired to run a problem in two steps. This is usually done for very deep penetrations or for unusual geometric configurations (such as ducts) where it may be more economical to run the problem in stages. The designation of a transmission region is, however, optional. Note that the program is capable of treating up to 10 different transmission regions.

An escape region is one in which all particles that enter are killed. It is ordinarily used to define the outer limits of the geometry (i.e., the complete geometry is enclosed in a large region which is designated as the escape region).

4.2.10 Scoring Regions

A scoring region is one in which the flux contribution is computed for each particle which passes through it. In a nonscoring region no such computation is made, so that the output edit provides fluxes only in those regions designated in the input as scoring regions.

In most problems it is desired to know the flux in every region separately, in which case each region in the problem would be defined as a scoring region with a different number. In some problems, however, two or more regions may be completely symmetric with respect to the source, in which case the fluxes in these symmetric regions could be combined without any loss of information, and, in fact, an improvement in the accuracy will be obtained. Each of these regions then would be designated by the same scoring region number. In still other problems it may be unnecessary to know the fluxes in certain regions. These should then be given scoring region number zero, which tabs them as nonscoring.

Since fluxes are only stored and printed out for scoring regions, it is possible to reduce both the size of the edit and the core storage requirements by reducing the number of different scoring regions. It should be remembered, however, that once a problem is run it is impossible to recapture any flux information in nonscoring regions.

4.2.11 Number of Histories and Statistical Groups

The user must designate the total number of source particles (histories) to be run in the problem. Although a greater number of histories will improve the accuracy of the answers, it will also increase the problem running time. The user must therefore strike a balance between the tolerable errors in the answers and the cost of running the problem. In complicated problems it is usually wise to run a test problem of 100 to 200 histories to get a "feel" for whether particles are reaching the desired regions. If they are not, the fault probably lies in incorrect importance sampling and the weights should be adjusted. If the test problem appears to have run"well," then the number of histories can be increased by perhaps a factor of about 10. After some experience, the user can generally determine the correct number of histories to run in a particular problem.

In running the problem the total number of histories is divided into aggregates called statistical groups. This is done in order to compute the variance (or standard deviation) of the fluxes. All particles (and their latents)

within a group are tracked before another group is treated. Fluxes are computed and stored on tape separately for each group. The size of the statistical group is constant in a given problem and must be specified in the input. The group size is not critical but should be small compared to the total number of histories to be run. About 20 histories per statistical group has generally been found to be adequate.

4.2.12 Volume Computation

To evaluate the flux, the track length in a region is divided by the volume of the region. Provision has been made to input volumes of regions if they are known. It often happens however that regions described by the combinatorial geometry technique have such complex shapes that an analytic volume computation is not practical. To determine the volume of such regions, a routine is included to perform a ray-tracing numerical integration calculation of the volume.

A grid plane is set up such that rays normal to the plane penetrate the geometry in question. Rays are passed through the plane with a fixed spacing ΔX and ΔY. The track length through any region is calculated and set equal to Z_n where n is the ray number. The volume, then, of a region is $\Delta X \, \Delta Y \, \Sigma_n \, Z_n$. The accuracy of the volume computation is dependent on the ΔX and ΔY spacings. For vehicles such as the M-60, accuracies of a few percent have been obtained with spacings of approximately one inch.

4.2.13 Secondary Production (GASP)

A. Interaction Scoring

SAM-C provides the user with a method of calculating the production and transport of secondary particles coming out of primary source particle interactions (e.g., γ-rays from neutron capture or inelastic scattering and secondary neutrons from fission or (n,2n) reactions). During the tracking of primary source neutrons, all interactions which are capable of producing secondaries are stored on the interaction tape (of Section 4.2.14). The stored data are the coordinates of the collision point, the energy and weight of the primary neutron, and the type of interaction. This tape then can be processed via the GASP program, which generates a source tape for use in a subsequent SAM-C problem. It should be noted, however, that GASP is run as a separate problem and the input to it should not be included with the SAM-C input package. The following discussion describes the basic input required to run GASP.

B. GASP (Gamma Secondary Production)

The first group of input specifies the number of elements in the configuration, number of regions in the problem, maximum number of primary neutron histories to be processed (this may exceed the number of source particles run in the SAM-C problem, in which case GASP will process all primary histories), maximum number of secondary particles to be created, and the number of energy bins for an

interaction tape edit. If the latter quantity is non-zero, the user must supply a list of primary neutron energy bins and the code will provide the number of absorptions and inelastic scatterings in each bin for each region. If the user wishes to edit the interaction tape but does not wish to generate a secondary source tape, a zero should be entered for the number of secondary particles to be created.

Assuming that a source tape is desired, the following input is required for <u>each element in the configuration.</u>

1. Atomic weight.
2. Number of different capture gamma energies produced per neutron capture.
3. Number of corresponding primary neutron energies. This quantity permits the user to change the capture gamma spectrum as a function of neutron energy.
4. Number of different inelastic scattering gamma-ray energies.
5. Number of corresponding primary neutron energies. This can be used to change the inelastic gamma spectrum as a function of neutron energy.
6. Capture gamma energy values.
7. Corresponding neutron energy values.

8. Number of gammas produced at each gamma energy for each neutron energy bin (i.e., starting with neutron bin #1, enter the number of gammas produced at each energy in the gamma spectrum, then repeat for neutron bin #2, etc.).

9. Items 6, 7, and 8 are then repeated for inelastic gammas.

After items 1 through 9 have been entered for each element, the user must specify the region and energy weights for secondary particles. The format is given in Section 4.2.14. Note that no angular weights are permitted in GASP. In setting up the secondary particle weights for GASP it also should be noted that the number of secondaries produced will be modified by the ratio of primary neutron weight to secondary gamma weight. Assume, for example, that the user has specified that one gamma ray is produced in a particular capture reaction. If the weight of the captured neutron was 1.0 and the secondary gamma weight is given as 0.5, the code will actually generate two gamma rays. This should be of no concern since the normalization to the proper energy spectrum is done correctly by the code. This method of generating secondaries can be used to advantage since it allows the user to suppress secondary production in certain regions (by making the region weight in GASP very large), while enhancing it in other regions.

The final group of input is used if an edit of the source tape is desired. The energy bin structure of secondary particles is specified and the edit provides the number of secondaries generated in each energy bin in each region.

The input requirements for the secondary SAM-C problem are similar to those for the primary problem. The statistical group size for the secondary problem should be the same as, or a multiple of, that of the primary problem.

4.2.14 Card Input Formats

Definitions of all input quantities are discussed below in their correct order.

A. TUNC Input

TUNC is the main program of the SAM-C system and the input consists of two cards.

Card 1

Any 80 Hollerith characters (usually serves as a title card).

Card 2

IBAND, NBAND (Format 2I10)

IBAND = 0 An organized data tape (ODT) will be generated and edited (BEDIT) and the geometry input will be added to the ODT.

IBAND = 1 An existing ODT will be edited and the geometry input will be added to the ODT.

IBAND = 2 An organized data tape (ODT) will be generated and the geometry input will be added to the ODT.

IBAND = 3 — The geometry input will be added to an existing ODT.

IBAND = 4 — An existing ODT also containing the geometry input will be used.

NBAND — The number of energy bands in which the cross sections are to be processed. During tracking, the cross sections in only one band are in the computer at any one time. An attempt should be made to keep the total number of cross sections roughly constant in each band.

B. BAND Input

The BAND routine organizes the cross sections and generates an ODT.

Card 1 - Identification Card (Formats I10, 2E15.5, 3I5)

Problem Number - any integer

Upper and lower energies - may be left blank.

Number of compositions - enter the total number of distinct compositions.

NG - enter 0 for a neutron problem or 1 for a gamma-ray problem.

NBAND - enter the same number as in the TUNC input.

Card 2 - Cross Section Band Limits (Format 5E14.5)

Enter the energy limits (in ev) of each band starting with the highest energy and proceeding to the lowest energy in the problem. Use as many cards as necessary.

Card 3 - Composition Identification (Format 2I10)

Composition number - an integer starting with 1 for the first composition.

Number of elements - enter the number of discrete elements or isotopes (i.e., U^{235} and U^{238} would count as two elements) in the composition.

Card 4 - Element Cards (one card per element) (Formats I5, E15.5)

ID - an integer which identifies each element. Make sure that this integer agrees with the identification number of the element on the element data tape. The identification numbers on this tape are usually, but not necessarily, equal to the integral atomic weights of the elements.

Concentration - enter the atomic concentration (10^{24} atoms/cm^3) of each element in the composition.

Cards 3 and 4 are repeated for each composition.

C. GENI (Geometry) Input

The format for the geometry input is discussed in Sections 2 and 3.2 of this report.

D. INPUTD Input

The INPUTD routine reads and stores the remaining input for a problem.

Card 1 - General Information (Formats 3I10, 8I5)

NSTART	The number of real time seconds of running time before terminating and editing.
NSTOP	Number of the last history to be treated.
NSTAT	Number of histories per statistical group.
NRMAX	Number of regions in the geometry.
NG	Enter 0 for a neutron problem or 1 for a gamma problem.
NT	Number of output time bins (enter 0 for a time-independent problem).
NOUT	Number of output energy bins.

NUMSC	Number of flux scoring regions.
NRWL	Number of distinct region weights.
IREX	The escape region number.
IRT(1)	The first transmission region number (leave blank if no transmission regions are desired).

Note that the above 11 items appear on a single card, the first three are Format I10 and the last eight are Format I5.

Card 1.1 - Transmission-Interaction Information (Format 14I5)

IRT(2) IRT(3) IRT(4) IRT(5) IRT(6) IRT(7) IRT(8) IRT(9) IRT(10)	Additional transmission region numbers. Leave blank if only one or if no transmission region.
IWA	Enter a "1" if absorptions are to be recorded on tape, otherwise leave blank.
IWI	Enter a "1" if inelastics are to be recorded on tape, otherwise leave blank.
IWE	Enter a "1" if elastics are to be recorded on tape, otherwise leave blank.
IWO	A multiplicity factor. Each transmission will be recorded on tape "IWO" times. If IWO is zero, no transmissions will be recorded.

Card 2 - Number of Point Detectors (Format 2I10)

NDET	Total number of point detectors.
NDFAP	Number of detectors for which flux-at-a-point is to be used. Statistical estimation will be used for the remaining detectors.

Card 3 - Point Detector Coordinates (Format 3E14.6)

XAD, YAD, ZAD	The X, Y, Z coordinates of each detector point. One card per detector. Enter the flux-at-a-point detectors first, followed by the statistical estimation detectors.

Card 4 - Cutoff Information (Format 5E14.5)

ECUT	Low energy cutoff (ev). Tracking of a particle is terminated if its energy degrades below ECUT.
ETH	Thermal energy if a thermal group is required. ETH must be within the energy limits of the problem.
TCUT	Leave blank.
FZ	See discussion in Section 4.1.
EHIGH	High energy cutoff (ev). This should be less than or equal to the highest energy for which cross sections are available.

Card Set 5 - Output Energy Bins (Format 5E14.5)

These cards give the boundaries of the output energy bins (ev) used for the flux edit. There should be five entries per card with a total of (NCUT + 1) entries. The energies should be listed from high to low with the lowest energy equal to or less than ECUT. The first and last entries should be negative. If supergroups are used, any number of intermediate energies may also be negative.

Card Set 6 - Output Time Bins (Format 5E14.5)

These cards give the boundaries of the output time bins. There should be five entries per card with a total of (NT + 1) entries. However, if NT = 0, omit Card Set 6

entirely. Times should be entered from high to low with the first entry equal to or greater than TCUT and the last entry must equal 0.

Card Set 7 - Region Weights (Format 5E14.5)

These cards give all of the region weights needed in the problem. They are entered five to a card with a total of NRWL entries. The order in which the weights are entered is irrelevant, but does determine the region weight numbers (i.e., entry one is weight #1, etc.).

Card 8 - Region Specifications (Format 6I5)

Use one card per region with a total of NRMAX cards. The first card applies to region 1, the second to region 2, etc.

ISC	Scoring region number in which the fluxes in this geometric region are to be stored. Several regions may be assigned the same ISC number. If ISC = 0, fluxes will not be stored.
NREG	Number of the composition to be found in this region. These numbers are defined by the BAND input.
IRW	Region weight number assigned to this region. A weight number is given by its position in the list of region weights.
IEW	Energy weight set number assigned to this region. If IEW = 0 there is no energy weighting in this region.
IAN	Aiming angle number assigned to this region. If IAN = 0 there is no angular weighting in this region.
IANG	Angular weight set number assigned to this region.

Card 9 - Energy Weight Specification (Format 2I10)

NEWL	Number of energy bins for energy weighting.
NEW	Number of distinct energy weight sets. If NEWL and NEW = 0, the problem contains no energy weighting and Cards 10 and 11 are omitted.

Card Set 10 - Bin Limits for Energy Weights (Format 5E14.5)

Enter the boundaries (ev) of the energy bins to be used for energy weighting. There should be five entries per card with a total of (NEWL + 1) entries. The energies should be entered in decreasing order. The lowest bin limit should be less than or equal to ECUT.

Card Set 11 - Energy Weight Sets (Format 5E14.5)

The energy weight value in each of the above energy bins should be entered. One or more sets of energy weights may be entered. Each set should contain NEWL entries and a new card should be used to start each set. There should be a total of NEW sets. The order in which the sets are entered determines the energy weight set numbers (the first set is weight set #1, etc.).

Card 12 - Angular Weight Specifications (Format 3I10)

NAIML	Number of distinct aiming angles.
NUMANL	Number of angular bins for angular weighting.
NUMANG	Number of distinct angular weight sets.

If the problem contains no angular weighting, enter 0 for the above three quantities and omit Cards 13, 14, and 15.

Card Set 13 - Aiming Angles (Format 3E14.5)

Enter the direction cosines of each aiming angle with respect to the X, Y, Z coordinates. Use a total of NAIML cards.

Card 14 - Bin Limits for Angular Weights (Format 5E14.5)

Enter the boundaries of the angular bins to be used for angular weighting. Boundaries are given in terms of the cosines of the angles, with the first entry equal to 1.0 and the last entry equal to -1.0. There would be a total of (NUMANL + 1) entries.

Card Set 15 - Angular Weight Sets (Format 5E14.5)

The angular weight value in each of the above angular bins should be entered. One or more sets of angular weights should be entered. Each set should contain NUMANL entries and a new card should be used to start each set. These should be a total of NUMANG sets. The order in which the sets are entered determines the angular weight set numbers.

Card 16 - Source Specifications (Format 3I10)

NSR	Number of different source regions in the problem. If NSR = 0, an external source tape is used and no further source input is required.
IFLAG	Number of energies used to define the source spectrum. If IFLAG = 0, a built-in Cranberg fission spectrum will be used for a neutron problem.
ISW	If ISW = 0, fluxes will normalize to one source particle. If ISW = 1, fluxes will be normalized to the total source power as given on the next card.

Card 17 - Source Regions (Formats I10, E20.10, I10)

One card is required for each source region with a total of NSR such cards.

ISR	Geometrical region number.
P	Power density in the region (source particles per unit volume).
ISO	If ISO = 0, the source particles will be emitted isotropically. If ISO = 1, the source will be monodirectional, with the direction specified on Card 21. The value of ISO (0 or 1) must be the same for all source regions.

Card 18 - Spectrum Description (Format 2E20.8)

These cards give the integrated source spectrum. Each card contains an energy (ev) and the integrated source above that energy. The first card contains the upper energy $\int S(E)dE$, of the source with the integral equal to 0. The last card gives the lower energy of the source with the integral equal to 1.0. The lower energy of the source should be less than or equal to ECUT, but greater than or equal to the lowest output flux energy bin limit.

Card 19 - Time Specifications (Format I10)

NOT	Number of time values used to specify the time distribution. Cards 19 and 20 should be omitted for a time-independent problem.

Card Set 20 - Time Distribution (Format 2E20.10)

Each of these cards should contain a time value and the corresponding integrated source up to that time $\int S(t)dt$.

These cards are, therefore, analogous to Card 18, except that time replaces energy. The first entry should correspond to the longest time with the integral equal to 1.0. The last entry is for time equal 0 with the integral equal to 0.

Card 21 - Monodirectional Source (Format 3E14.5)

If ISO = 1, the direction cosines of the monodirectional source with respect to the X, Y, Z coordinate axes should be entered. If ISO = 0, omit this card. Do not use angular weighting in the source regions if a monodirectional source is specified.

Card 22 - Volume Parameters (Format 2I10)

IR	The region number of the vertex point XV. If IR = 0, then all volumes will be read in and no computations will be made.
IPUN = 0	Do not punch out computed volumes.
IPUN ≠ 0	Punch out computed volumes.

Card 23 - The X, Y, Z Coordinates of the XV Point (Format 3E20.8)

Card 24 - The X, Y, Z Coordinates of the XT Point (Format 3E20.8)

Card 25 - The X, Y, Z Coordinates of the XO Point (Format 3E20.8)

Card 26 - The X, Y, Z Coordinates of the XA Point (Format 3E20.8)

The above four cards describe the vertices of the box within which the volumes will be computed.

Card 27 - DOD, DT, (Format 2E20.8)

DOD is the increment on the line (XV, XO). DT is the increment on the line (XV, XT). The two increments define the scanning mesh for the numerical integration.

Note that cards 23 through 27 are to be omitted if IR = 0.

Card 28 - Pre-Computed Volumes (Formats I10, E20.8)

IR1 is the region number of the pre-computed volume. VR is the pre-computed volume.

Repeat Card 28 for all regions having pre-computed volumes.

The last card of this set must contain an IR1 greater than the number of regions.

Card 29 - Built-in Response Data (Format 5E14.5)

The built-in response data are as described in Section 4.2.8. Enter the C(E) data in the same order as the output energy bins. Enter NOUT - 1 numbers. If no built-in response is desired, enter the corresponding number of blank cards.

E. GASP Input

The GASP routine is used to edit the interaction tape produced in a primary neutron problem and to generate and edit a secondary gamma-ray or neutron source tape.

Card 1 - General Specifications (Format 6I5)

NELE	Number of elements present in the configuration.
IRMAX	Number of geometric regions in the configuration.
IHCUT	Number of primary histories to be processed.
IGCUT	Number of secondary particles to be generated.
NOUT	Number of energy bins for editing the interaction tape. If NOUT = 0, the edit is bypassed and Card 2 should be omitted.
IPRIN = 0	Print GASP element data.
IPRIN = 1	Print only element parameter card of GASP element data.

Card 2 - Output Energies for Interaction Tape Edit (Format 5E14.5)

Enter the limits of the output energy bins in which the interaction tape is to be edited. The edit will provide the number and type of interactions occurring in each bin for each region. There should be a total of (NOUT + 1) entries in decreasing order. Cards 3 through 9 are to be omitted if IGCUT = 0. In this case no secondary source tape will be generated. If IGCUT is greater than 0, repeat Cards 3 through 9 for each element (i.e., these seven cards are specified in order for the first element, then go back to Card 3 and repeat the input for the second element, etc.).

Card 3 - Element Parameter Card (Format 5I5)

IATWT	Atomic weight (this must be the same number used to identify the element in the BAND input.
LA	Number of discrete capture gamma energies. Enter zero if no capture gammas are produced.
KA	Number of corresponding incident neutron energies. Enter zero for no capture gammas. If the capture gamma spectrum is independent of neutron energy, enter a 1.
LI	Number of inelastic gamma ray energies. Enter a zero if no inelastic gammas are produced.
KI	Number of corresponding incident neutron energies.

Card 4 - Capture Gamma Energies (Format 6E12.4)

Enter the energies in the capture gamma spectrum in increasing order. These should be a total of LA entries. Omit this card if LA = 0.

Card 5 - Captured Neutron Energies (Format 6E12.6)

Enter the upper bound of each neutron energy bin, in ascending order. The lower bound of the lowest energy bin is not entered and is assumed by the code to be 0. There should be a total of KA entries. Omit this card if KA = 0.

Card Set 6 - Capture Gamma Numbers (Format 6E12.6)

For each of the KA neutron energies, enter the number of gammas produced at each of the LA gamma energies. Start with the lowest neutron energy bin and enter each gamma number in increasing order of gamma energy. Start a new card

for each neutron energy bin. There should be a total of KA x LA entries. Omit this card if either Card 4 or Card 5 was omitted.

Card 7 - Inelastic Gamma Energies (Format 6E12.6)

Identical to Card 4 except that the entries are inelastic gamma energies instead of capture gamma energies. There should be a total of LI entries. Omit if LI = 0.

Card 8 - Inelastically Scattered Neutron Energies (Format 6E12.6)

Enter the boundaries of the inelastically scattered neutron energy bins, starting with the lower bound of the lowest bin and proceed in ascending order. There should be a total of KI entries. Omit if KI = 0.

Card Set 9 - Inelastic Gamma Numbers (Format 6E12.6)

Identical to Card 6 except that the entries are the number of inelastic gammas produced at each of the LI gamma energies. Start a new card for each neutron energy bin. Since the number of neutron bins equals (KI-1), the total number of entries must equal LI x (KI-1).

Repeat Cards 3 through 9 for each element.

Cards 10 through 15 - Region and Energy Weights

These cards specify the weights for the secondary particles. The input format is identical to that used in the INPUTD input. Note that angular weighting is not permitted so that IAN and IANG are left blank in Card 12. Omit these six cards if IGCUT = 0.

The following two cards are used to obtain an edit of the source tape being generated. If no source tape was generated (IGCUT = 0) they may also be used to edit an existing source tape.

Card 16 - Number of Energy Bins for Source Tape Edit (Format I5

NOUT	Number of output energy bins. Within each of these energy bins, the edit will provide the number of secondary gammas generated in each region.

Card 17 - Energies for Source Tape Edit (Format 5E14.5)

Enter the limits of the output energy bins for the source tape edit in decreasing order. There should be a total of (NOUT + 1) entries.

4.3 ORGANIZATION OF THE SAM-C PROGRAM

The SAM-C program consists of two main programs and their associated subroutines.

The first main program is TUNC. TUNC reads and processes all input. The input is distributed among various common blocks and in an array labeled MASTER or ASTER. The two arrays are equivalenced and the different names are used to reference fixed point or floating point data. The MASTER-ASTER array contains the bulk of the input, i.e., cross sections, geometry, and Monte Carlo track length scores. A diagram detailing the allocation of storage in the MASTER-ASTER array is shown in Appendix D. A flow chart of the TUNC program is shown in the discussion of the TUNC routine.

The second main program is called MONTE. MONTE controls the actual Monte Carlo calculation and the flux-at-a-point calculation. The routine controls the reading of cross-section bands and the writing on tape of the answer arrays. A detailed discussion is given in the description of the MONTE program.

4.3.1 Description of Routines

This section gives a brief description of every routine in the program. The routines which have an asterisk (*) superscript are of major importance. The descriptions are given in alphabetical order. The brief descriptions are followed by detailed descriptions of the important routines.

Note that the following routines are used by the SAM-C program and the MAGIC program.

BOX	SPH
DIGCON	TROPIC
FLOCON	UN1
G1	UN2
S	WOWI
SEE3	

The above routines are described in Section 3.3.1.

Routine	Description
ALBERT	An input processing routine to read in arbitrary polyhedron (ARB) data. The routine will check for errors and also put the data in the MASTER array.
ARB	A tracking routine to calculate distance to first intersection (RIN) and distance to last intersection (ROUT) for the ARB figure.

Routine	Description
BAND*	The cross-section processing routine. The routine reads in composition data. Element data for the required elements is read from an element data tape (EDT). The element and composition data are used to calculate macroscopic total cross sections. Composition, element, and total cross-section data are placed in the MASTER array and on the organized data tape (ODT).
BEDIT	A routine to edit the (ODT) in a readable form.
BOX	A tracking routine to calculate distance to first intersection (RIN) and distance to last intersection (ROUT) for the BOX figure.
CARLO*	The Monte Carlo calculation routine. The routine controls importance sampling, collision mechanics, and scoring.
CLOC	This routine will print the time as computed by the computer internal clock.
DCOSP	Given two position vectors, XA and XB, the routine will compute direction cosines WA from XA to XB.
DIGCON	A routine to read Hollerith data and convert to integer data. The routine is called by GENI and is part of the input processing procedure.
DIREC	Using a direction vector $\underset{\sim}{W}$ as a polar axis, a direction $\underset{\sim}{W}'$ is generated the cosines of whose polar angle is CSTHT and with random azimuthal angle. Thus, new direction cosines W' are computed such that $\underset{\sim}{W} \cdot \underset{\sim}{W}' = \text{CSTHT}$. The routine is used in computing the new direction after a scattering.

Routine	Description
DR*	The collision mechanics routine. The routine is used to calculate macroscopic total cross sections as a function of energy. The routine also determines energy and angle after scattering.
ELL	A tracking routine to calculate distance to first intersection (RIN) and distance to last intersection (ROUT) for the ellipsoid of revolution.
EDIT	The flux edit routine. The routine prints flux answers as a function of energy and regions, and dose answers as a function of region. The routine also prints flux-at-a point results.
FAP*	The flux-at-a-point calculation routine. The routine calculates flux-at-a-point scores as a function of energy and detector point. The scores are stored in the MASTER array and printed out by the EDIT routine.
FLOCON	A routine to read Hollerith data and convert to floating point data. The routine is called by GENI and is part of the input processing procedure.
GE	A routine called by FAP to compute intermediate results to be used in the flux-at-a-point scores.
GENI*	The major geometry input processing routine. The routine reads geometry data, checks for errors, and puts the data into the MASTER array in the form required by the tracking routines.
GI*	The main geometry tracking routine. Given a position $\underline{X}$, a direction $\underline{W}$, and a region IR the routine will calculate the distance "S" from the point $\underline{X}$ to the next region in the direction $\underline{W}$. The routine also determines IR' the next region to be encountered.

Routine	Description
INPUTD*	The Monte Carlo input routine. The routine reads Monte Carlo input data and stores it in the MASTER array for use by the calculation routines. Region specification, importance sampling data, output energy meshes, and flux-at-a-point data are handled by this routine.
NOPR	A routine to return the current status of the computer internal clock.
MONTE*	The control routine for the Monte Carlo calculations. The routine manipulates the organized data tape (ODT) and the supergroup tapes. The routine also outputs on tape the scores for each statistical aggregate for later use by the edit routines.
PACK	A routine to pack the position, direction, energy and weight data for a particle into seven computer words. The seven-word records are used for latent storage and for output onto the interaction/transmission tape.
PEDIT	The control program for the edit routines. The routine sets up arrays for use as storage for the edit routines.
PICK	The control routine for supergroup latents. This routine stores particles whose energies are not included in the supergroup currently being processed. These particles are stored by PICK in either the central memory or on tapes depending on the amount of core available for the particular problem.
RAW	A tracking routine to calculate distance to first intersection (RIN) and distance to last intersection (ROUT) for the right angle wedge.
REC	A tracking routine to calculate distance to first intersection (RIN) and distance to last intersection (ROUT) for the right elliptic cylinder.

Routine	Description
REACOM	This routine reads all of the common blocks from tape. At the point of writing all the input has been processed and distributed among the common blocks. All common blocks are written on tape and will be read in by subsequent Monte Carlo routines using REACOM.
RCC	A tracking routine to calculate distance to first intersection (RIN) and distance to last intersection (ROUT) for the right circular cylinder.
RPP	A tracking routine to calculate distance to first intersection (RIN) and distance to last intersection (ROUT) for the rectangular parallelepiped.
RPPIN	An input processing routine to process the RPP data and store it in the MASTER array.
RPP2	Given that a ray has intersected an RPP, the routine will determine which RPP lies on the other side.
SEEK	Given a vector ET with elements monotonically decreasing and a variable E, this routine will search through the ET and determine the bin containing E.
S	A routine used in extracting RPP data from the MASTER array.
SOUCAL*	The input processor for the source information required by the Monte Carlo routines. The routine processes energy and time spectrum data, and source region data. The data are stored in the MASTER array.
SOUPIC*	A routine to generate initial source particles. The routine uses the source information processed by SOUCAL and delivers position, energy, time, direction, and weight data for source particles.

Routine	Description
SPH	A tracking routine to calculate distance to first intersection (RIN) and distance to last intersection (ROUT) for the sphere figure.
SUBED	The main edit routine. The routine reads the answer arrays from the statistical aggregate tape and prepares it for editing by the EDIT routine. Dose responses are also calculated in this program.
STAT	A routine to calculate standard deviations. The routine accepts data from the EDIT routine and computes standard deviations.
TALLY	A summary routine. The routine prints a one line summary of results for each statistical aggregate. Quantities such as number of collisions, absorptions, degradations, births, and deaths are printed for each aggregate.
TERP	A linear interpolation routine.
TRALA	A tracking routine used by FAP, the main flux-at-a-point routine. The routine computes distances from an initial point to a final point.
TRC	A tracking routine to calculate distance to first contact (RIN) and the distance to last contact (ROUT) for the truncated right cylinder.
TROPIC	A routine to generate a vector of direction cosines from an isotropic direction.
TUNC*	The main control program for the entire Monte Carlo program. The routine controls the cross-section processing routines, BAND and BEDIT, the geometry input routine, GENI, and the Monte Carlo input routine, INPUTD.

Routine	Description
UNPR	A routine to unpack two computer words containing six integer variables and to deliver the six integer variables stored in the common block labeled REGPAR.
UN1	A routine to unpack the integer part of a word containing a packed floating point and an integer variable.
UN3	A routine to unpack a computer word containing three integer variables.
UNPACK	A routine to unpack a seven-word vector containing position, direction, energy, and weight information.
VELF	The velocity computation routine. The routine is used only for time-dependent problems and computes velocity as a function of energy.
VOLUM	The volume computation routine. Region volumes are computed by numerical integration.
WOWI*	The routine that determines which region a ray will enter next. The routine is called by "G1".
WRICOM	The routine which writes all common blocks on tape after all input has been processed. The routine is called by TUNC.
WRT14	A routine to write seven-word records onto tape. The routine is called whenever a transmission or interaction is to be put on tape.
XDIST	Given XA and XB, two position vectors, the routine computes the geometric distance between XA, XB.

Subroutine BAND

The purpose of BAND is to arrange the cross-section data in an organized fashion for the particular problem in process.

The GENDA and GENPRO programs are used to generate an element data tape (EDT) which contains microscopic cross sections and probability tables for all nuclides of interest. This EDT can be considered as a library tape for any subsequent Monte Carlo problems.

For a specific problem, however, the system is divided into regions which contain given nuclide compositions. Each composition may be described by a set of nulcides and their respective concentrations. When a particle enters collision at some point within a region of given composition, the Monte Carlo program has to select the nuclide in the composition with which the particle collides.

The probability that a particle with energy E will interact with nuclide k of the composition R is given by

$$P(k,R) = C(k,R)\sigma_T(k,E) \Big/ \mu_{T,R}(E)$$

where $\mu_{T,R}(E)$ is the total macroscopic cross section of the composition, and $C(k,R)$ is the concentration of nuclide k in region R.

$$C(k,R) = \frac{\rho_k}{A_k} N_o v_{k,R}$$

where ρ_k = density of the nuclide, k

A_k = its atomic weight

No = 0.06023×10^{25}, Avogadro's number

$v_{k,R}$ = volume fraction of the nuclide.

The BAND program searches and reads from the EDT all information concerning the nuclides which appear in the various compositions specified in the input. It computes the total macroscopic cross sections of each composition at each final energy, $\mu_{T,R}(E_j^G)$, using the concentration data supplied as input to the program. Finally, it stores all data concerning the nuclides and the compositions of the problem in a prescribed order.

Hence, the role of BAND is, above all, the organization of the cross-section and probability data read from the EDT. In addition, it computes and stores the $\mu_{T,R}(E_j^G)$'s.

$$\mu_{T,R}(E_j^G) = \sum_{k's\ in\ R} C(k,R)\,\sigma_T(k,E_j^G).$$

The output energy mesh for BAND is the same as that of the EDT. Although the present GENPRO output is at a fixed lethargy mesh, BAND and subsequent programs can accept an

arbitrary energy mesh.

The output of BAND is arranged in energy bands. The number NBAND of bands, and the (NBAND + 1) band limits are specified on input. The ODT consists of all cross sections and probability tables for the high-energy band for all nuclides and compositions, followed by all cross sections and probability tables for the next energy band for all nuclides and compositions, etc., down to the lowest energy.

Subroutine BEDIT

BEDIT edits the ODT generated by BAND. The edit is by energy band.

First Line

BAND-FLAG, sequential number of the band being edited.

Second Line

NO. of BANDS, total number of bands in problem.

Third Line

Is a heading for the table that follows.

JEN1 is the location of the first word of the energy mesh table; JEN2 is the location of the next to the last word of the energy mesh table.

Next Heading

REGION	Composition number
LOCMU	The location of the first word of the macroscopic total cross-section table for that composition
LOCCON	The location of the first word of the table giving the concentrations of different nuclides.

This is followed by a table for all compositions present in the problem.

Next Heading

CONC	Concentration of nuclide
LOC	Location of the first word of microscopic cross-section data for the nuclide
ATWT	Atomic weight of the nuclide.

This is followed by a table for all nuclides in each composition present in the problem.

This is followed by a printout for each nuclide present in the problem.

Heading

LOELEV	Location of first word of table giving discrete energy levels
ATWT	Atomic weight of nuclide
LPLEV	Number of discrete energy levels
LCHI	Length of CHI-table
LENN	Length of ENN-table

We will mention here that the angular distribution (E, $\cos\theta$) is divided into N equally probable intervals of $\cos\theta$. The N + 1 boundaries of $\cos\theta$ define N + 1 values of $\chi = (1 + \cos\theta)/2$, which are stored in a CHI table for the energy E. The length of the CHI-table is N + 1.

Similarly the spectrum of inelastic neutrons P(E,E′) is divided into M equally probable intervals of E′, and the M + 1 boundaries of E′ are stored in an ENN-table for the

energy E. M + 1 is the length of the ENN table.

Next Heading

E	Total cross section, in barns
SIGMA	If CHI = 1, elastic scattering is isotropic in the CM system. If CHI = 2, elastic scattering is with hydrogen (special routine). If = 4, elastic scattering isotropic in the lab system. If = 5, the Klein-Nishina formula is to be used (gamma rays only). If >5, the CHI is the location of the first word of the appropriate CHI-table.
SCATTER	Elastic scattering cross section, in barns
PLEV	If = 0, inelastic scattering is not via discrete levels. If >0, PLEV is the location of the first word of a table giving probabilities to excite the various levels.
ABSORB	Absorption cross section, in barns
ENN	If = 0, inelastic scattering does not give rise to a continuous spectrum of inelastic neutrons. If >0, ENN is the location of the first word of the appropriate ENN table.
L	Location of the total cross-section entry.

This is followed by a listing of all the tables referred to (CHI, ENN, PLEV, ELEV), together with their relative locations.

After the printout for all the elements comes the printout of macroscopic cross sections for all the compositions, in cm^{-1} vs E, in ev.

This is the end of the edit for neutron problems. For γ-rays, one has, in addition, a printout of $\Sigma_i C_i Z_i$ for each composition, where C_i is a concentration, and Z_i is an atomic number.

Subroutine CARLO

Source particles, either read from source tape or generated internally by the subroutine SOUPIC, as well as latents, are tested on energy. If the energy is below the lower bound of the supergroup currently treated, they are stored as latents. For energies within the supergroup, they are transmitted to the subroutine CARLO which tracks, distributes further collisions, if any, and scores answers.

Given the region number IR, the energy E, and the direction of flight WB, a total weight

$$W = W_{IR} \times W_E \times W_\Omega$$

is calculated and a new F given by the ratio F/W is treated as the total number of biased particles born simultaneously at XB, T, with an energy E, in the direction WB. The carry-along statistical weight W_C, can be considered as a normalization factor, to be transmitted to all future collided particles due to the present source particle. For internally generated particles the quantity F always will be unity. Externally generated particles will have an F different from unity if they were not picked from the properly biased source distribution. The quantity F is tested vs an input cutoff value FZ. If it is smaller, a game of chance

is played, and, with probability (1-F) the particle is killed; whereas with probability F the particle is kept with F set as unity. These two events are scored as either (FxW) "kills" or (1-F)xW "births".

If the particle survives the test, a subroutine DR is called, which provides the total macroscopic cross section μ in the region IR. A random number ξ_1 is picked for picking future collision points.

A geometry routine Gl(S_1, IR', X') is called to provide the distance S_1 to the first region boundary encountered from XB in the direction WB. IR' is the region number on the other side of the boundary, and X' is the point of intersection of the track with the boundary. If the region IR is a scoring region, the contribution to the flux is calculated and scored. Points where particles enter into collision are picked by computing

$$\xi' = \xi_1 - F\left(1 - e^{-\mu S_1}\right).$$

If $\xi' < 0$, a particle comes into collision in region IR at X = XB + S'WB where $S' = -1/\mu \log (1 - \xi_1/F)$; the interaction routine DR is called to pick a particle coming out of collision. The interaction is written on tape, and the particle coming out of collision (if any) is stored as a latent. ξ_1 is then increased by one, and control is returned to the part of the code which computes ξ' to attempt

to produce more collisions in region IR. When $\xi' < 0$, no more collisions are produced in region IR. F is then set to F x $e^{-\mu S_1}$ x W, where W is the weight. A test is made whether the next region is a transmission region (if it is, the coordinates X', IR', energy, time, etc., are written on tape) and whether it is the escape region. If it is not an escape region, the particle tracked is moved to the boundary by setting IR = IR', by computing the new weight W, and by setting a new F equal to F/W. If F <FZ, the particle is either killed or F is set equal to 1, both events being properly tallied. For F <FZ, ξ is set equal to ξ', the DR routine is called to obtain the new total cross section μ, and control is transferred to the part of the code which calls the geometry routine Gl (S_1, IR', X'), thus the tracking continues until either a kill occurs or the escape region is reached. In the latter case F "escapes" are tallied.

Subroutine DIREC (CSTHT, W, WP)

Using $\underline{W}$ as a polar axis, a direction $\underline{W}'$ is generated the cosines of whose polar angle is CSTHT and with a random azimuthal angle. Thus the new direction cosines W' are computed such that:

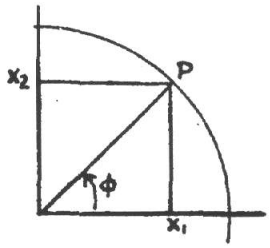
$$\underline{W} \cdot \underline{W}' = \text{CSTHT}$$

The routine will first generate the sine and cosine, $\sin\phi$, $\cos\phi$ of the random azimuthal angle. The procedure is as follows:

1. Pick two random numbers $0 \leq X_1, X_2 \leq 1.0$

   The point P lies in the quarter circle if

   $$X_1^2 + X_2^2 \leq 1.$$



2. Pick sets of X_1, X_2 until $X_1^2 + X_2^2 \leq 1$.
3. At this point the random angle is $0 \leq \phi \leq \pi$

   If we take 2ϕ as our random angle, the range becomes $(0, 2\pi)$ and

   $$\cos\phi = \frac{X_1^2 - X_2^2}{X_1^2 + X_2^2}$$

   $$\sin\phi = \frac{2 \cdot X_1 X_2}{X_1^2 + X_2^2}$$

4. Given $\sin\phi$ and $\cos\phi$ the new set of direction cosines is

   $$WP(1) = T2 \cdot \{W(1) \cdot W(3) \cdot \sin\phi + W(2) \cos\phi\} + W(1) \cdot CSTHT$$

   $$WP(2) = T2 \cdot \{W(2) \cdot W(3) \cdot \sin\phi - W(1) \cos\phi\} + W(2) \cdot CSTHT$$

   $$WP(3) = W(3) \cdot CSTHT - T_1 \cdot T_2 \sin\phi$$

Where $T_1 = \sqrt{W(1)^2 + W(2)^2}$

$$T_2 = \sqrt{\frac{1-CSTHT^2}{T_1}}$$

5. If $T_1 \leq 10^{-10}$, the new set is given by

$WP(1) = \sqrt{1-CSTHT^2} \cdot \cos\phi$

$WP(2) = \sqrt{1-CSTHT^2} \cdot \sin\phi$

$WP(3) = CSTHT \cdot W(3)$

Subroutine DR (N, NNREG)

This routine performs all of the collision mechanics. The routine is called by CARLO and FAP to retrieve total cross sections and to make collisions. Subroutine TRALA will call the routine to retrieve total cross sections.

The argument NNREG is the composition number and the argument N defines the calculation to be performed. The calculations performed for each value of N are given below.

N = 1

Find the energy interval containing the current energy (E). Subroutine SEEK is used here. Compute an interpolation factor (FACINT) to be used in calculating point values of cross sections at energy E. The remainder of the calculation for N = 1 is the same as for N = 2.

N = 2

Calculate the macroscopic total cross-section (FMUTOT) for composition NNREG. The calculation is done by linear interpolation between the two point values of the cross-section interval and corresponding energy interval containing E. The interpolation factor FACINT is used.

The DR routine is called with N = 1 only when a new energy is encountered by the CARLO or FAP routines. Given the energy interval and the interpolation factor, the N = 2 option is used when the composition number NNREG changes and E remains constant.

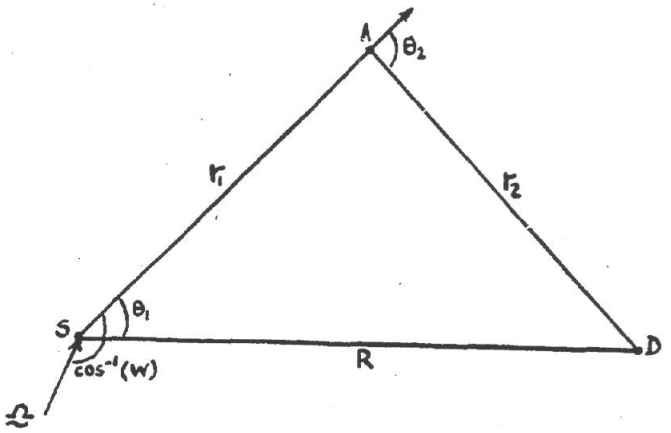
N = 3

This option performs the actual collision mechanics. The output will be a new energy (EPRIM), a new direction (WP), the cosine of the scattering angle (CSTHT), and an integer (NCDB) denoting the type of interaction. The types of interaction are listed below.

NCDB	INTERACTION
1	Scattering with hydrogen
2	Isotropic elastic scatter
3	Anisotropic elastic scattering
4	Scattering in the LAB system, no degradation
5	Elastic scattering for gammas
6	Absorption
7	Inelastic continuous scattering
8	Inelastic discrete scattering

Subroutine FAP

The method used to estimate flux-at-a-point involves obtaining by normal Monte Carlo methods a distribution of points where particles enter into collision and estimating the once-more-collided flux, at the point of interest, using a special distribution of the last collision point.



Assuming a collision point at S, a detector at D, as shown in the sketch, the expression for the once-more-collided flux is

$$\phi = \int g_S(W) \frac{e^{-\int_0^{r_1} \mu dS}}{4\pi r_1^2} \mu_A g_A(\cos\theta_2) \frac{e^{-\int_0^{r_2} \mu dS}}{4\pi r_2^2} \cdot dV$$

where $g_S(W)dW$ is the probability of scattering through an angle whose cosine is in the range (W, W+dW).

For source particles one has to estimate in addition the uncollided flux.

If the intermediate points A are picked from a distribution given by

$$\frac{R}{\pi^3} \quad \frac{dV}{r_1^2 r_2^2}$$

the estimator becomes

$$f_1 = \frac{\mu_s \pi}{16R} q_s q_A e^{-\int_0^{r_1} \mu ds} e^{-\int_0^{r_2} \mu ds} .$$

This is the method used in the present code. It turns out that there is a very simple way to pick intermediate points A from the distribution, if a proper choice of coordinate system is made.

Let us define the point A by the three angles θ_1, θ_2 and ϕ where θ_1 and θ_2, and ϕ are defined on the sketch and ϕ is an azimuth around SD. The distribution becomes:

$$\frac{R}{\pi^3} \quad \frac{r_1^2 \sin\theta_1 \, d\theta_1 \, d\phi \, \frac{\partial r_1}{\partial \theta_2} \, d\theta_2}{r_1^2 r_2^2} .$$

Using the relationship:

$$\frac{r_1}{\sin(\theta_2 - \epsilon_j)} = \frac{r_2}{\sin\theta_1} = \frac{R}{\sin\theta_2}$$

one has

$$r_2 = \frac{\sin\theta_1}{\sin\theta_2} R \qquad \frac{\partial r_1}{\partial \theta_2} = \frac{\sin\theta_1}{\sin^2\theta_2} R \ .$$

Substituting, one obtains the distribution

$$\frac{1}{\pi^3} \, d\theta_1 \, d\theta_2 \, d\phi$$

i.e., a random distribution of θ_1, θ_2, ϕ in the range

$$0 \le \theta_1 \le \theta_2 \le \pi$$
$$0 \le \phi < 2\pi \ .$$

Once θ_1 and θ_2 are picked, the cosine of the angle of scattering w is easily obtainable, and the score can be evaluated.

In the actual program, coding is provided for either the procedure outlined above, which gives a finite variance provided the detector is not in the immediate vicinity of the source region, or for a simple statistical estimation method, which gives finite variance if the detector is not in the immediate vicinity of either the source region or of a scattering material. In that case the quantity

$$q_s(\cos\theta) \, \frac{e^{-\int_0^R \mu \, dS}}{4\pi R^2}$$

is scored for each collision point S, where θ is the angle between the directions $\vec{\Omega}$ and $\vec{SD}$. The type of calculation to be performed is specified on input, by specifying the total number of detectors ND, and the number NDFAP of detectors ND, and the number NDFAP of detectors for which the first procedure is to be applied. The remaining (ND-NDFAP) detectors will be treated by statistical estimation.

If ND $\neq$ 0, the flux-at-a-point routine FAP is called for each source particle by the MONTE program, and for each collision point by the CARLO routine, except if inelastic scattering occurs giving an outgoing energy below the supergroup being currently treated. In that case the particle is stored as a latent, and MONTE will call the FAP routine with that latent when the proper supergroup will be treated.

Once called, the subroutine FAP examines whether the particle to be treated is a source particle, a particle coming out of inelastic scattering, a particle coming out of elastic scattering, or a latent from a previous FAP calculation. In the first three cases, the trigonometric functions of θ_1 and θ_2 picked according to the distribution are obtained. This is done rather efficiently by picking directly the sin and cos functions of two random angles, and then determining which corresponds to the smallest angle, and therefore which should be assigned to θ_1; the others are assigned to θ_2. θ_1 and θ_2 are kept the same for

all FAP detectors in the problem.

For source particles, the uncollided flux is computed (the source is assumed to be isotropic).

The contribution of the once-more-collided flux is computed as follows. The distances r_1 and r_2 are given by $r_2 = R \sin\theta_1/\sin\theta_2$ and $r_1 = R \sin\theta_1 - r_2\sin\theta_2$. A random azimuth around $\vec{SD}$ is picked. For both source particles and inelastic particles, $g_S = 1$, and the outgoing energy is as set by the main code (either energy of the source particle, or energy of the inelastic particle). (No energy-angular correlation is assumed in the code. The difference between lab and center-of-mass system is also ignored for inelastic scattering.) For particles coming out of elastic scattering, the angle of scattering is computed in the lab system. A subroutine GE is called. For neutrons, it performs the transformation to the CM system, looks up the probability g_S and computes the outgoing energy E. For gamma rays g_S and E are given by the Klein-Nishina formula. If this energy is below the supergroup currently treated, the par-energy is below the supergroup currently treated, the particle is stored as a latent of the first kind to be picked up and returned to this point of the coding at a later time in the calculation.

The subroutine DR is called, and the total cross section μ is made available. A subroutine TRALA is called, which tracks the particle a distance r_1 in the direction of flight, with an energy E, and computes $\lambda_1 = \int_0^{r_1} \mu \, dS$;

if the particle escapes, an 'infinite' λ_1 ($\lambda_1 = 10^{27}$) is returned, no score is made, and the next (if any) detector is treated. If the particle did not escape, the subroutine DR is called to determine the type of interaction occurring at the intermediate point. Absorption corresponds to no score. For inelastic particles, E is picked from the spectrum by the DR routine, and $g_A = 1$. For elastic scattering, the subroutine GE determines g_A and E. If the energy E is below the supergroup currently treated, the particle is stored as a latent of the second kind, to be picked up and returned to this point of the coding at a later time in the calculation. Finally, the subroutine TRALA provides

$$\lambda 2 = \int_0^{r_2} \mu ds$$

and the total contribution is calculated and scored, and the next detector (if any) is considered, unless the calculation dealt with a latent of either first or second kind, which applies only to a particular detector. When all FAP detectors have been considered, the statistical estimation detectors are treated by using the coding described above for the tracking from intermediate point to the detector, where the intermediate point is replaced by the original collision point.

Subroutine MONTE

The program MONTE is the program where all the tracking and collision mechanics are performed. The program arranges the calculations in supergroups. We have seen that the program BAND arranges cross sections in certain energy bands. The output energy bins are also arranged in certain output supergroups. Cross-section input corresponding to a single band can be stored in computer memory at any given time. Scores corresponding to a single output supergroup can be stored in computer memory at any given time. The two meshes (bands and output supergroups) do not necessarily coincide. A combined mesh defines a set of supergroups.

The program MONTE starts by reading in the highest energy cross-section band, and by arranging the memory for the highest energy output supergroup. The highest of the low-energy bounds of these energy ranges defines EBL, the low energy of the supergroup currently treated.

The program then calls the source-picking routine SOUPIC, and examines the energy E. If E<EBL, the particle is stored as a latent by calling the subroutine STORE, and SOUPIC is again called. When E>EBL, the subroutine CARLO is called. The subroutine CARLO tracks the particle, scores contributions to fluxes when needed, and distributes collisions along their track. Particles coming out of collision are also tracked if their energy is above EBL; particles coming out of collision with E<EBL are stored as latents by

calling STORE. Control is finally returned to MONTE, which proceeds to the next source particle, until a complete statistical aggregate of particles has been treated for the highest supergroup.

At this point, MONTE switches to the next supergroup by either reading a new band of cross-section data, or by writing out on tape the set of scores obtained and preparing the memory layout for the next supergroup, or both. It then proceeds to call a subroutine PICK, which picks latents from previous supergroups. If E<EBL, the particle is stored again as a latent by calling STORE. If E>EBL, CARLO is called. The procedure continues until all latents have been examined, at which point MONTE switches to the next supergroup until the low-energy cutoff is encountered. When this occurs, MONTE switches to the highest supergroup, and proceeds to treat the next statistical aggregate of particles. The calculation terminates when a history number exceeds the cutoff value NHIST specified on input. A "blank" interaction record, with NHIST = NHIST + 1, is written on the interaction tape and all tapes are rewound. Control is transferred to TUNC which calls the editing program PEDIT.

Subroutine PACK (X, WX, E, IR, T, IDET, F, NHIST, WC, J12345. P)

Subroutine PACK will pack all the items in the above argument list into seven computer words and store them into the variable P which has a dimension of 7. The items in

the list are described below.

X	A three-element array containing X,Y,Z, position coordinates
WX	A three-element array containing X,Y,Z direction cosines
E	Energy of the particle
IR	Region number of the particle
T	Time of flight if the problem is time dependent
IDET	A flux-at-a-point detector number or zero
F	An importance sampling parameter
NHIST	The history number for the particle
WC	An importance sampling weight
J12345	An indicator denoting the type of particle.

The correspondence between the individual items and the packed array P are shown as,

P(1) = X(1)	packed with WX(1)	48 bits X(1)	12 bits WX(1)
P(2) = X(2)	packed with WX(2)	48 bits X(2)	12 bits WX(2)
P(3) = X(3)	packed with WX(3)	48 bits X(3)	12 bits WX(3)
P(4) = E	packed with IR	45 bits E	15 bits IR
P(5) = T	packed with IDET	45 bits T	15 bits IDET
P(6) = F	packed with NHIST	45 bits F	15 bits NHIST
P(7) = WC	packed with J12345	45 bits WC	15 bits J12345

60 bit computer word

Note that the WX array is floating point. Thus, in order to compact it into 12 binary bits truncation is required. The actual FORTRAN statements to do the packing are given below.

```
J =     (WX(I)+1.0)*1024.
P(I) = (X(I).AND.777777777777770000B). OR.J
```

where I = 1, 2, 3.

The first statement puts the WX in the range 0,2. and truncates it to an integer in the range 0,2048 which will fit into 12 bits. The second statement does the actual packing.

Subroutine PICK

We have seen throughout the previous sections that particles degrading below the energy EBL, low-energy limit of the supergroup currently treated, were stored as latents by calling the subroutine PICK. They were later picked up by MONTE by calling the subroutine PICK. The subroutine INPUTD allocates the memory to data, scores, etc. The remaining memory is assigned to the subroutine PICK, to be used as a buffer for latents. One "end" of the buffer is assigned to "degraded" particles. This is the end of the buffer where particles are being stored. The other "end" of the buffer is assigned to "unsorted" particles, i.e., the particles to be picked. Associated with each end of the buffer is a magnetic tape to be used when the buffer overflows. There are two modes of operation. In one mode,

the top of the buffer is unsorted and the bottom is sorted. When the switch is made from one supergroup to the next, the "unsorted" part is empty, and the "degraded" part may have particles which become "unsorted" for the supergroup about to be treated. The designation of the buffers (and of the tapes) is therefore switched.

There is no set boundary between the two "ends" of the buffers. The number of particles in the "unsorted" buffer keeps decreasing, whereas the number of particles in the "degraded" buffer keeps increasing, and can increase faster than the other number decreases. Therefore, the two parts of the buffer can meet, causing an overflow of the buffer.

It is then determined which "end" of the buffer is longest, and a number of particles exactly equal to one-half the total length of the buffer are written from the longest "end" onto the corresponding magnetic tape.

When the "unsorted" buffer becomes empty, a test is made whether any "unsorted" particles are available on the corresponding magnetic tape. If none are available, the calculation has been completed for the current supergroup. If some are available, they are read into the buffer if room is available. If room is not available, it is made available by writing out part of the other buffer on the other tape; the length of the record written out from one "end" is equal to the length of the record to be read into the other "end".

The subroutine PICK deals with different kinds of latents. The quantities stored are: X, Ω, E, IR, T, I, F, NHIST, WC, and J12345, where X is the position, Ω the direction, E the energy, IR the region number, T the time, F the weight, NHIST the history number, and WC a normalization factor. I and J12345 are indices.

J12345 = 1 - identifies a source particle

= 2 - identifies a particle coming out of elastic scattering

= 3 - identifies a particle coming out of inelastic scattering

= 4 - identifies a latent of the second kind for FAP only

= 5 - identifies a latent of the first kind for FAP only.

(In other parts of the code, J12345 = 6 identifies a transmitted particle, J12345 = 7 an inelastic interaction, and J12345 = 8 an absorption.)

I is irrelevant (set to 0) for J12345 = 1, 2, 3 (and 6). For FAP latents (J12345 = 4, 5), I is set to IDET, the detector number for which the latent applies. In the description of interactions (J12345 = 7, 8), I is set to IATWT, a five-digit identifier of the element with which the interaction occurred.

When the program MONTE calls the subroutine PICK, it examines the J12345 obtained and calls the proper routines:

J12345 = 1, 3 - both FAP and CARLO

J12345 = 2 - CARLO only

= 4, 5 - FAP only.

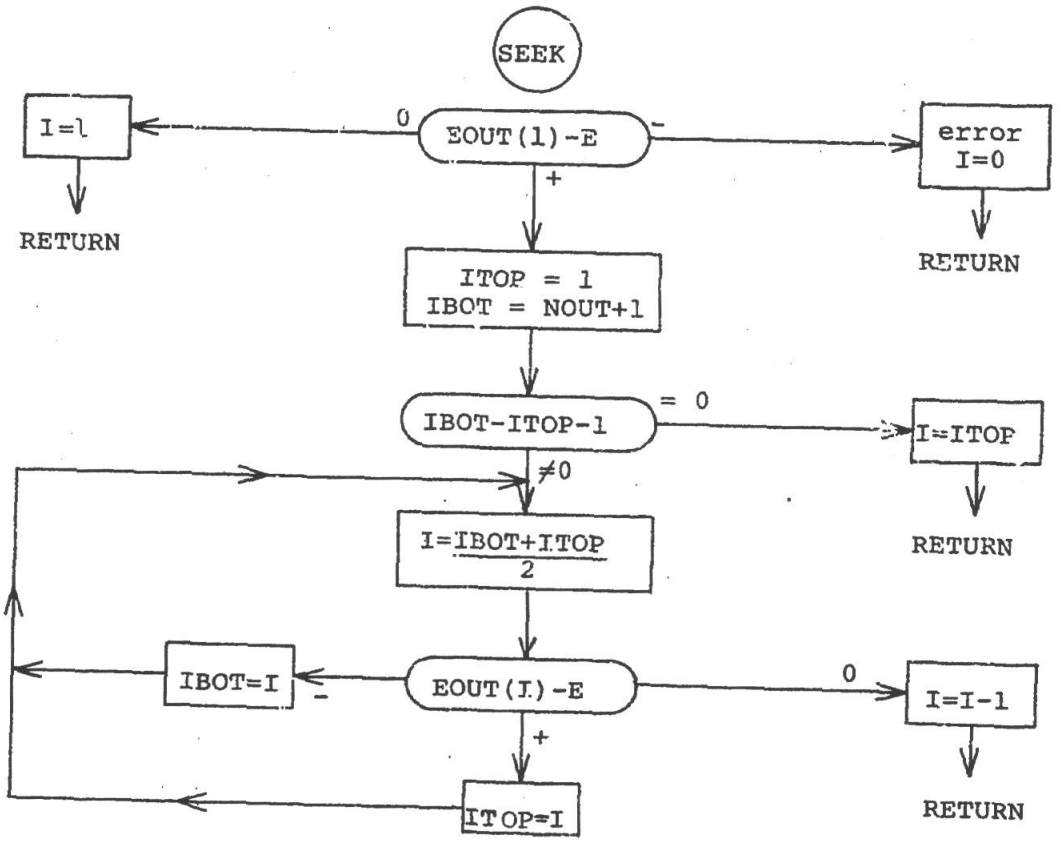
Subroutine SEEK (E, EOUT, NOUT, I)

Given the vector array EOUT, of length NOUT + 1, and the argument "E" the routine will perform a binary search and return "I" such that

EOUT(I)≥E>EOUT(I+1).

An error message is given if EOUT(1)<E.

A flow chart is given below.



Note: The EOUT array must be monotonic and decreasing.

Subroutine SOUCAL

SOUCAL will read source data and prepare tables for use by SOUPIC.

The first card contains:

NSR — Number of regions where the source extends. If this number is 0, an external source tape is expected, and no further input is required.

IFLAG — If IFLAG = 0, a built-in Cranberg fission spectrum will be used for the energy spectrum. If IFLAG >0, the spectrum is specified later in input. It will be defined by IFLAG entries.

ISW — Switch determining the normalization of problem. If it is 0, fluxes will be normalized to a single (unbiased) source particle. If it is 1, fluxes will be normalized to the "total power" of the source as defined below.

This is followed by (NSR) cards specifying the (NSR) source regions. Each of these cards gives:

ISR — Specification of a geometrical region number.

P — Power density in that region.

ISO — Flag indicating angular distribution:

ISO = 0, isotropic,

ISO = 1, monodirectional. If ISO ≠ 1 in any of the (NSR) regions, a monodirectional source will be assumed in all the (NSR) regions.

After this follow the (IFLAG) cards specifying the energy spectrum assumed to apply to all the source regions. This

is a table of E vs F(E) where

$$F(E) = \int_E^\infty S(E)\,dE \quad .$$

Linear interpolation is assumed on E vs log F(E) . The first entry must be for $E \geq E_{high}$, and the last for $E \leq E_{cut}$.

If there is time dependence (NT >0), the time dependence of the source must be specified. A card gives:

NOT — Number of cards. This card is followed by NOT cards giving

$$t \text{ vs } \int_0^t S_t(t)dt.$$

Linear interpolation is assumed between entries in the table.

Finally, if the source is monodirectional, the projections Ω_x, Ω_y, Ω_z of the direction must be given on the last card.

The subroutine SOUCAL reads in all this input, prints it back, and pre-computes tables to pick directly from the biased source distribution. The code first pre-computes a table of

$$SPEC(I) = \int_{E_I}^\infty S(E)dE,$$

where the E_I's take the values of E_{high}, of all the energy

boundaries where the energy weight changes, and E_{cut}. The code then runs through all the source regions, and, for each new energy-importance set encountered, pre-computes a table of

$$SPEC(I,J) = \int_{E_I}^{\infty} \frac{S(E)dE}{W_E(E)}$$

where J runs from 1 to the total number of different energy-importance sets encountered in the source regions; also for each new angular importance set encountered, a table

$$P(I,K) = \int_{\omega_i}^{1} \frac{d\omega}{W_\omega(\omega)}$$

where the ω_i's take the values of cos θ at which the angular weight changes, and K runs from 1 to the total number of different angular importance sets encountered. The different tables are renormalized, and both the modified and unmodified integrated source are computed in each source region. The former quantities are proportional to the probability with which particles should be picked in different source regions. A table SOUR(L) is built up, which gives the cumulative probability for a source to be picked in the M^{th} region for $M \geq L$.

Subroutine SOUPIC

This is a subroutine which picks particles from the biased source distribution.

If an external source tape is to be used, groups of 35 source particles are read from tape into a buffer, and returned one by one to the main code. The quantities describing a source particle are:

XB	Coordinates of the particle
IR	Region number where particle is born
WB	Direction of the particle
T	Time at which particle is born
E	Energy of the particle
NHIST	History number attached to the source particle
F	Statistical weight of the particle (usually set equal to unity).

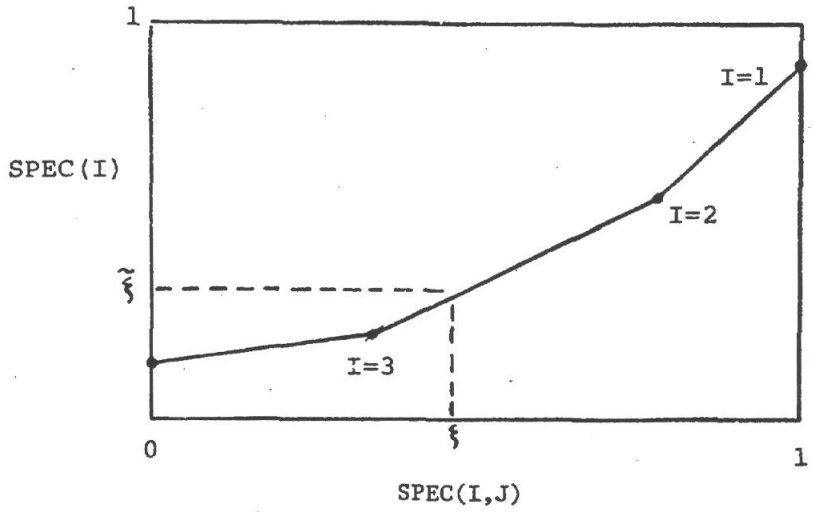
The procedure for internal source generation is outlined below.

A first random number ξ is compared to the table SOUR(L) (computed by SOUCAL). The smallest L for which SOUR(L) $>\xi$ determines the region IR = ISR(L) to be picked from. Standard techniques are used to pick coordinates of points uniformly distributed in a region.

The energy is picked next. A stratified random number (called CE in the code) is obtained (stratification is done for each statistical aggregate of source particles). A biased random number ξ is then obtained by interpolation in the SPEC vs SPEC(J) tables pre-computed by SOUCAL. (The J is determined by the region number.) Finally the energy is determined by solving the equation

$$\tilde{\xi} = \int_E^{\infty} S(E)\,dE$$

using semi-log interpolation.



The direction of WB of the source particle is determined as follows. If the source is isotropic and there is angular importance sampling, the cosine of the angle between the particular aiming angle and the direction is chosen by picking a random number, and interpolating between the angular mesh supplied on input vs the table P(I,K) precomputed by SOUCAL. (The K is determined by the region number.) A random azimuth is then picked, which completes the specification of the direction. In the absence of angular importance in the source region, standard techniques are used. The case of a monodirectional source also can be

be handled, provided there is no angular importance in the source region. Finally, if time dependence is to be determined, a time T is determined by another random number ξ and the solution of the equation

$$\xi = \int_0^T S_t(t)dt.$$

The quantities communicated to the main code are XB, IR, WB, T, E, NHIST, F, and W_c (where $W_c = 1$), and F is the weight in region IR at energy E in the direction WB.

Subroutine STAT (TOTSUM, SUMSQ, FLUX, SD, NHIST, V, DE)

Subroutine STAT is called by the EDIT routine and calculates FLUX, SD which are to be printed by the EDIT program. The calculation flow is shown below.

```
H = (NHIST/NSTAT)*NSTAT
```

where NHIST is total number of histories processed by the Monte Carlo program and NSTAT is the size of a statistical aggregate.

```
GH = FLOATF (NSTAT)/H
OMGH = 1.-GH, if OMGH = 0, set it 1.0
DEV = (SUMSQ-GH*TOTSUM**2)/OMGH
DEV = SQRTF (DEV)
```

at this point DEV is the standard deviation of TOTSUM.

```
SD = 100.*DEV/TOTSUM, SD = the "coefficient of
                           variation" or percentage error.
```

SD is set to 99.99 if TOTSUM = 0. or if SD > 99.99

FLUX = TOTSUM/(DE*H*V)

where DE is the ΔE of the energy bin, and V is the volume of the region.

Subroutine TALLY (J, NHIST, NCOL, NDEG, NABS, FKILL, BIRTH, ESCAP, NRMAX)

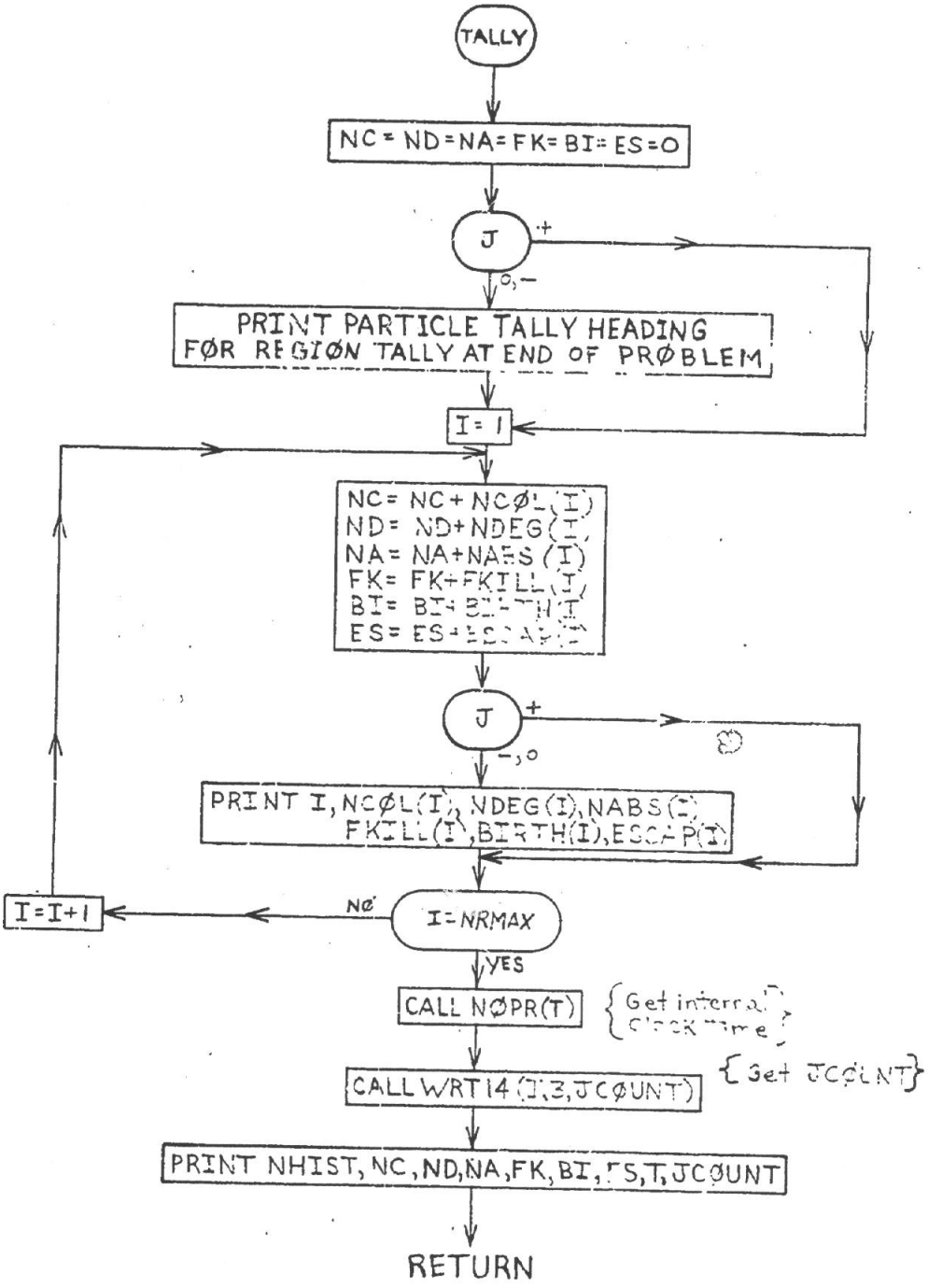
The routine is used to print a tally at the end of each statistical aggregate and to print a tally by region at the end of the problem.

At the end of each aggregate the following items are printed.

NC	Total number of collisions thus far
ND	Total number of degrades thus far
NA	Total number of absorptions thus far
FK	Total number of kills thus far
BI	Total number of births thus far
ES	Total number of escapes thus far
T	Elapsed time thus far
JCOUNT	Total number of particles on the interaction/transmission tape.

The same items, except for ES and T, are printed at the end of the problem as a function of region.

A flow chart of the routine is given.


TALLY
NC = ND = NA = FK = BI = ES = 0
J
+
0, −
PRINT PARTICLE TALLY HEADING
FØR REGIØN TALLY AT END OF PRØBLEM
I = 1
NC = NC + NCØL(I)
ND = ND + NDEG(I)
NA = NA + NABS(I)
FK = FK + FKILL(I)
BI = BI + BIRTH(I)
ES = ES + ESCAP(I)
J
+
−, 0
PRINT I, NCØL(I), NDEG(I), NABS(I),
FKILL(I), BIRTH(I), ESCAP(I)
I = I + 1
NØ
I = NRMAX
YES
CALL NØPR(T)
{Get interval clock time}
CALL WRT14 (I, 3, JCØUNT)
{Get JCØUNT}
PRINT NHIST, NC, ND, NA, FK, BI, ES, T, JCØUNT
RETURN

Subroutine TRALA (AMDIN, R)

Given an initial point XB and a direction WB, the routine will track from XB in the direction WB. The tracking will continue until the distance "R" measured from XB is reached.

During the tracking, the routine will form

$$AMDA = \Sigma \Delta R_i \mu_i$$

Where ΔR_i is the geometric thickness of region "i" and μ_i is the total macroscopic cross section of region i. The quantity AMDA is the number of mean free paths along the line segment from XB to XB+WBR.

Subroutine UNPACK (X, W, E, IR, T, IDET, F, NHIST, WC, J12345, P)

Subroutine UNPACK will unpack the seven words in the array "P" and distribute the data into the argument list. The correspondence between P and the individual items is shown in the discussion of subroutine PACK.

The FORTRAN statements used to unpack P(1), P(2), P(3) are described below.

X(1) = P(1)	Store the X part of the packed word
TEMP = P(I)	Retrieve the integer value of W
TEMP = TEMP.AND.7777B	
TEMP = ITEMP	Float the integer value
W(I) = TEMP/1024.-10 for I = 1, 2, 3	Re-normalize and adjust the range to -1, +1.

Subroutine UNPR (IEGT, IR)

The routine will unpack six integer variables stored in two words in the IEGT array.

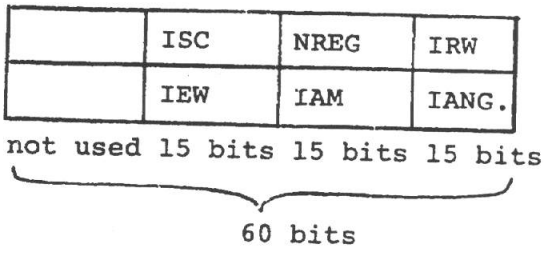
The two packed words are located as follows:

I = 2* (IR-1)

I1 = IEGT(I+1) the two packed words

I2 = IEGT(I+2)

The two packed words contain the six integer variables in the following format:



Function VELF (E)

This FORTRAN function computes velocity of neutrons or gammas as follows:

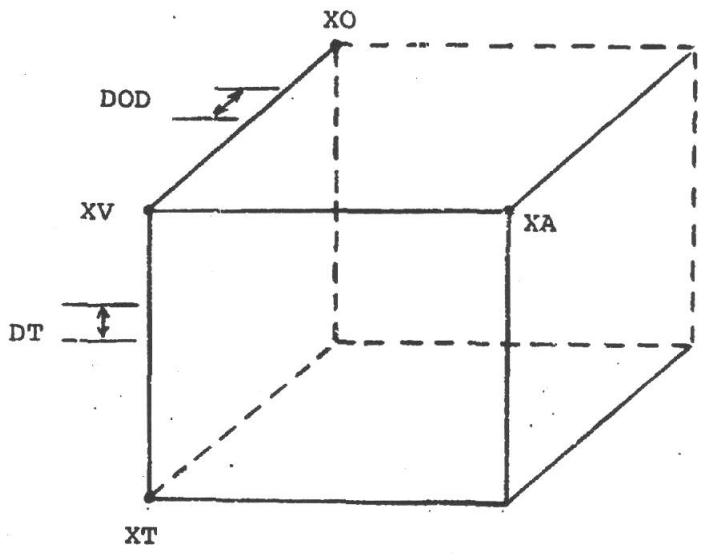
VELF = $1.38333 \times 10^{6} \cdot \sqrt{E}$ for NG = 0 (neutron)

VELF = 2.997928×10^{10} for NG = 1 (gamma)

The variable, NG, is found in Common.

Subroutine VOLUM

Input to this routine is the coordinate points that specify a box



XV, XT, XO, XA, then two additional input numbers, DOD and DT, specify two delta distances over which ray origins will increment.

The routine reads the input parameters as specified above and calculates the number of rays to be shot, starting between XV and XT at increments of DT. As each ray is shot, the distance through each region in the box is added to a distance counter for that region. When all rays between XV and XT have been treated, the code shifts each starting point a distance DOD in the direction from XV to XO and

repeats the above process until XO is reached by the point which started as XV. After shifting the starting points, rays are again fired in the direction from XV to XA and the region counters are incremented.

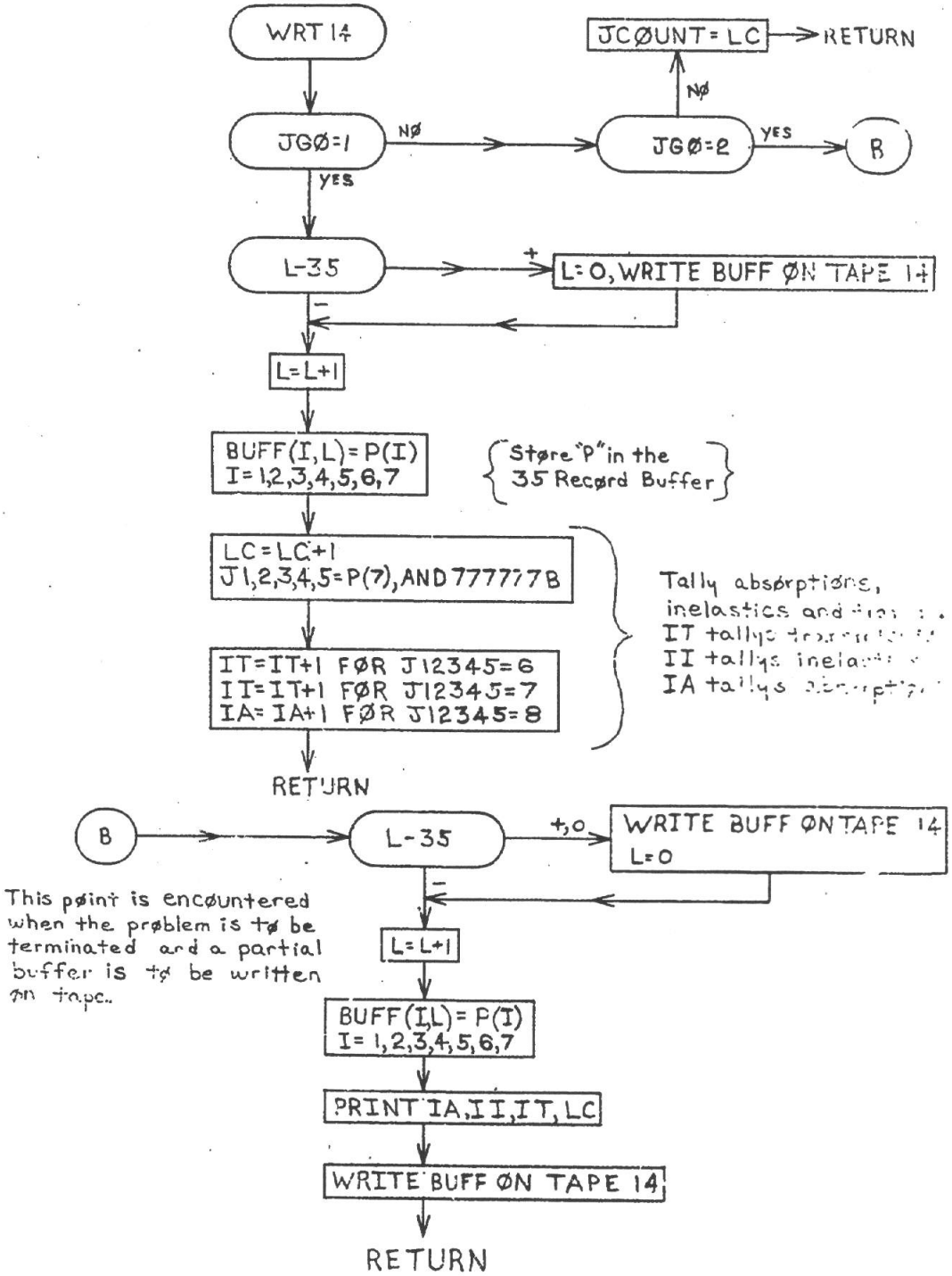
When all the rays are processed, each region distance counter is multiplied by the product DT x DOD, thus producing the volumes of the region within the box.

The routine has the capability of reading in pre-computed volumes and calculating a percent error. These pre-computed volumes should be punched with the region number in Column 10 and then the volume in Columns 11-30. A special card completes the volume input (see page 184).

Subroutine WRT14 (P, J, GO, JCOUNT)

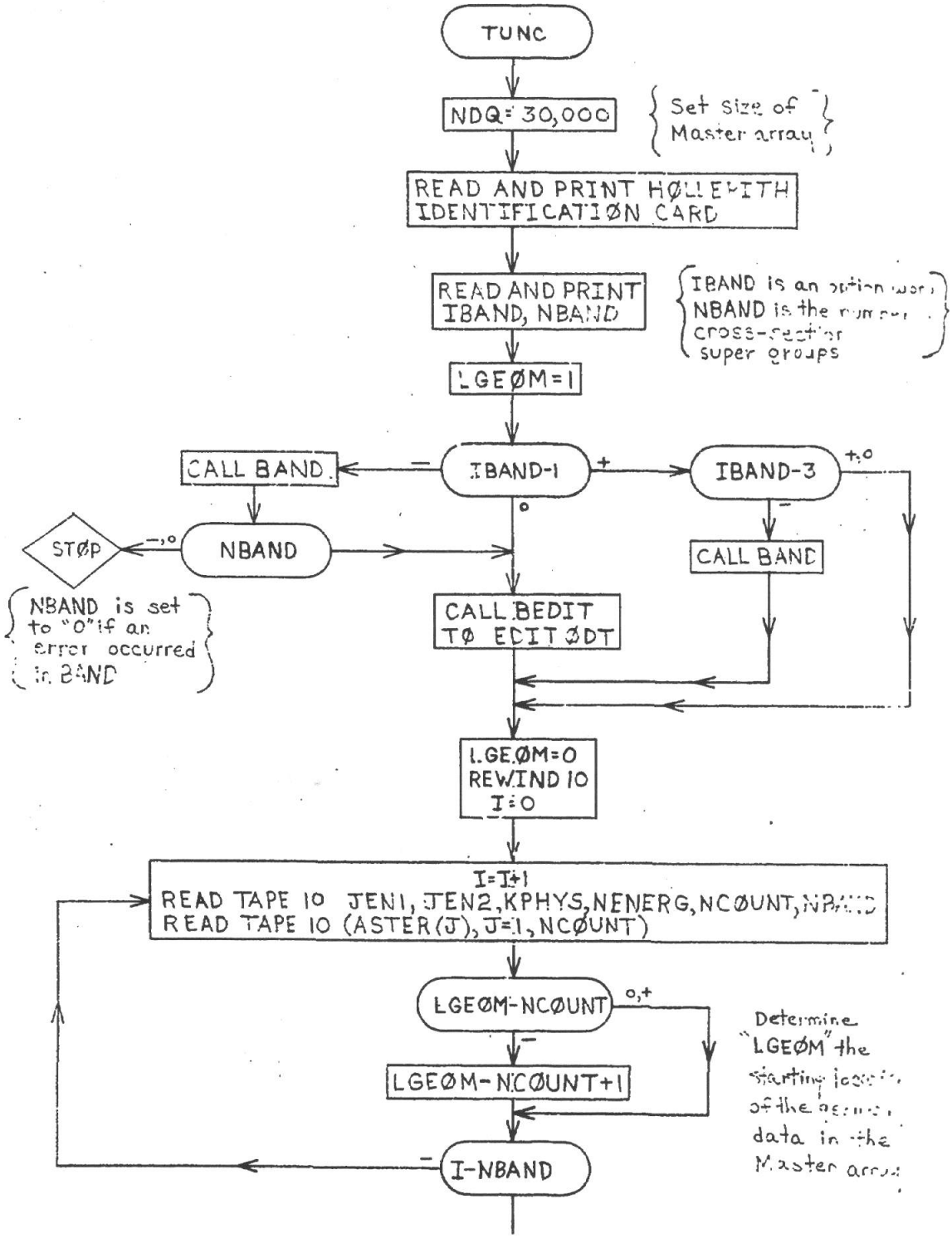
This routine accepts seven word records stored in the array P and writes them on tape 14 in units of 35 records each. Thus each unit on tape has 35 x 7 = 245 words. The seven word records are either transmissions or interactions. The integer part of P(7) denotes the type of record. The program will use the integer part of P(7) to tally transmissions, absorptions, and inelastics.

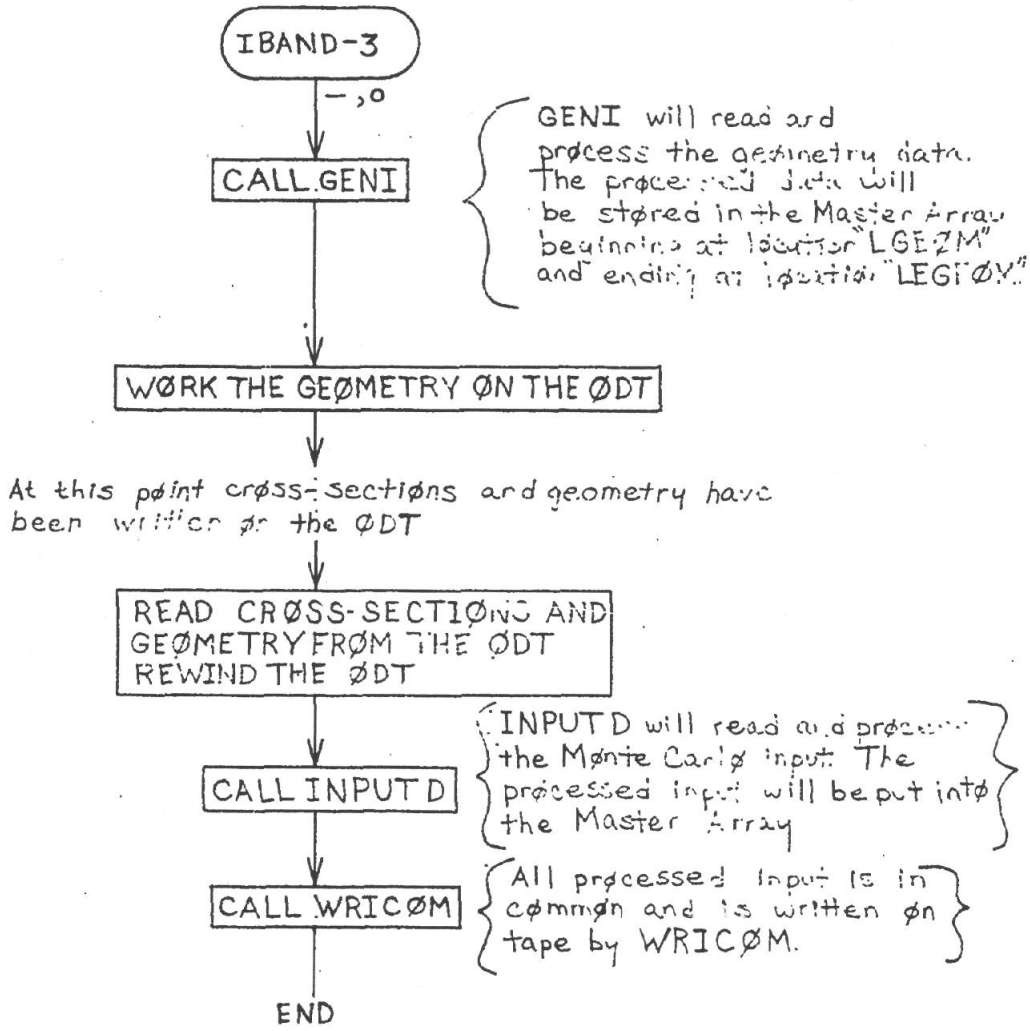
A descriptive flow chart follows.


WRT 14
JCØUNT = LC
RETURN
NØ
JGØ=1
NØ
JGØ=2
YES
B
YES
L-35
+
L=0, WRITE BUFF ØN TAPE 14
-
L=L+1
BUFF(I,L)=P(I)
I=1,2,3,4,5,6,7
Støre "P" in the 35 Recørd Buffer
LC=LC+1
J1,2,3,4,5=P(7),AND 777777B
IT=IT+1 FØR J12345=6
II=IT+1 FØR J12345=7
IA=IA+1 FØR J12345=8
Tally absørptiøns, inelastics and
IT tallys
II tallys inela
IA tallys
RETURN
B
L-35
+,0
WRITE BUFF ØN TAPE 14
L=0
-
This pøint is encøuntered when the prøblem is tø be terminated and a partial buffer is tø be written øn tape.
L=L+1
BUFF(I,L)=P(I)
I=1,2,3,4,5,6,7
PRINT IA,II,IT,LC
WRITE BUFF ØN TAPE 14
RETURN

Program TUNC

TUNC is the main control program and controls the entire calculation up to the actual Monte Carlo. A flow chart describing the functions of TUNC is given on the following pages.


TUNC
NDQ=30,000
Set size of Master array
READ AND PRINT HØLLERITH IDENTIFICATIØN CARD
READ AND PRINT IBAND, NBAND
IBAND is an option word
NBAND is the number of cross-section super groups
LGEØM=1
IBAND-1
CALL BAND
IBAND-3
STØP
NBAND
NBAND is set to "0" if an error occurred in BAND
CALL BEDIT TØ EDIT ØDT
CALL BAND
LGEØM=0
REWIND 10
I=0
I=I+1
READ TAPE 10 JEN1, JEN2, KPHYS, NENERG, NCØUNT, NBAND
READ TAPE 10 (ASTER(J), J=1, NCØUNT)
LGEØM-NCØUNT
LGEØM-NCØUNT+1
I-NBAND
Determine "LGEØM" the starting location of the group data in the Master array


IBAND-3
-,0
CALL GENI
GENI will read and process the geometry data. The processed data will be stored in the Master Array beginning at location "LGEOM" and ending at location "LEGTOV."
WORK THE GEOMETRY ON THE ODT
At this point cross-sections and geometry have been written on the ODT
READ CROSS-SECTIONS AND GEOMETRY FROM THE ODT REWIND THE ODT
CALL INPUTD
INPUTD will read and process the Monte Carlo input. The processed input will be put into the Master Array
CALL WRICOM
All processed input is in common and is written on tape by WRICOM.
END

4.3.2 Description of Output

The printed output will consist of four parts:

1. Cross-Section Output
2. Geometry Output
3. Monte Carlo Data
4. Answers.

The four parts will be discussed in detail below.

1. Cross-Section Output

The cross-section output is dependent on the input parameter, IBAND. The resulting output as a function of IBAND is:

IBAND = 0	A printout of all TUNC and BAND input (see Section 4.2). A printout of cross sections used by the problem. See the discussion of BEDIT for actual output formats.
IBAND = 1	A printout of the data on the existing ODT. See the discussion of BEDIT for actual output formats.
IBAND = 2	A printout of TUNC and BAND input (see Section 4.2).
IBAND = 3	A printout of TUNC input.
IBAND = 4	A printout of TUNC input.

2. Geometry Output

The geometry output is also dependent on the input parameter IBAND. The dependence is shown on the following page.

IBAND = 0 A complete printout of geometry input and the entering and leaving tables. See Section 3.3 for a description of formats.

IBAND = 1 Same as IBAND = 0

IBAND = 2 Same as IBAND = 0

IBAND = 3 Same as IBAND = 0

IBAND = 4 No geometry output will be printed.

3. Monte Carlo Data

This output section will consist of a complete printout of all Monte Carlo input. In addition, the core storage requirements for cross sections, geometry, and other inputs will be tabulated. The results of the VOLUM calculation will be printed. A detailed discussion of the volume output appears in the discussion of Subroutine VOLUM.

4. Answers

The program will print for each statistical aggregate the following items:

History number

Number of collisions

Number of degrades

Number of absorptions

Number of kills

Number of births

Number of escapes

The elapsed real time in seconds

The number of transmissions and interactions recorded on tape.

Note that all of the above items are cumulative and are printed as a single line for each aggregate.

When all the histories have been processed a printout giving the total number of absorption, inelastics, and transmissions will occur.

The previously discussed tallies as a function of aggregate will be repeated as a function of region. Thus, for each region the number of collisions, degrades, absorptions, etc., will be printed.

The dose, flux, and flux-at-a-point printouts will occur next. The doses are printed five regions per line of output. Both the flux and flux-at-a-point edits are printed five regions across the page and energy groups down the page. For each answer a percentage error is given. The volume of each flux region is also printed.

4.3.3. Glossary of Important Variable Names

The following is a brief outline of each common block in the program. A detailed description of each variable in common, in the order in which they appear in the TUNC program, is given on the following pages.

The "blank" common block contains the master storage array and frequently used input arrays.

Common REGPAR contains six parameters used in the Monte Carlo calculations.

Common INPUT contains non-subscripted input parameters.

Common PAREM contains position, direction, energy and importance sampling parameters.

Common COMPUT contains parameters computed from input.

Common CROSS contains cross-section parameters.

Common METRY contains geometry dependent input and computed parameters.

Common FAP contains flux-at-a-point parameters.

Common SOU contains source dependent parameters.

Common PUTAD contains data controlling the parameters to be written on the interaction, transmission tape.

"Blank" Common

EOUT(100)	An array containing the output energy bins for flux results. The array contains all bins for all output supergroups.
EWTAB(50)	The energy mesh for energy importance sampling.
ANGLE(50)	The c[illegible]ine mesh for angular importance sampling.
TTAB(50)	The time mesh for time dependent problems.

ASTER(30,000)	The master storage array containing cross sections, geometry, input and flux data. A complete description appears in Appendix D. The array is equivalenced with array MASTER. The name of the array in BAND and BEDIT is XSECT.

Common REGPAR

ISC	The location, in MASTER, of the flux scores for scanning region ISC.
NREG	The composition number for region IR.
IRW	The location, in MASTER, of the energy weight for region IR.
IEW	The location, in MASTER, of the energy weight table for region IR.
IAM	The location, in MASTER, of the aiming angle for region IR.
IANG	The location in MASTER of the angular weight table for region IR.

Note that the above six items are all keyed to a region number, "IR". The six items are retrieved from the MASTER array by a call to UNPR.

Common INPUT

NSTART	The number of real time seconds of running time before terminating and editing.
NSTOP	Number of the last history to be treated.
NSTAT	Number of histories per statistical group.
NRMAX	Number of regions in the geometry.
NG	Either 0 for a neutron problem or 1 for a gamma problem.

NT	Number of output time bins.
NOUT	Number of output energy bins.
NUMSC	Number of flux scoring regions.
NRWL	Number of distinct region weights.
IREX	The escape region number.
NEWL	Number of energy bins for energy weighting.
NEW	Number of distinct energy weight sets. If NEWL and NEW = 0, the problem contains no energy weighting.
NAIML	Number of distinct aiming angles.
NUMANL	Number of angular bins for angular weighting.
NUMANG	Number of distinct angular weight sets.
JRT	Not used.
ECUT	Low energy cutoff (ev). Tracking of a particle is terminated if its energy degrades below ECUT.
ETHERM	Thermal energy if a thermal group is required. ETHERM must be within the energy limits of the problem.
TCUT	Leave blank.
FZ	See discussion in Section 4.1.
EHIGH	High energy cutoff (ev). This should be less than or equal to the highest energy for which cross sections are available.
EBL	Lower bound of the current supergroup.
EBH	Upper bound of the current supergroup.

Common PAREM

XB(3)	The X, Y, Z coordinates of the current particle's position.
WB(3)	The direction cosines of the current particle.
E	The current energy of the particle.
IR	The region number of the particle.
T	The current time of flight of the particle.
IDET	A detector number for flux-at-a-point.
F	An importance sampling parameter.
NHIST	The current history number.
WC	A weight parameter.
SP	The distance, measured from XB, to the next collision.
WP(3)	The direction cosines of the particle after scattering.

Common COMPUT

NUMNOU	The product of NUMSC (the number of scoring regions) and NOUT (the number of output energy bins).
JONUM	An index used in flux scoring.
LNCNOL	The location in MASTER of the collision by region table.
LBIRTH	The location in MASTER of the birth by region table.
LREGT	The location in MASTER of the region data table.
LFKILL	The location in MASTER of the kills by region table.

LESCAP	The location in MASTER of the escapes by region table.
LLAST	The location of last word in MASTER used by the program.
NDQ	The size of the MASTER array (30,000).
LNDEG	The location in MASTER of the degrade by region table.
LNABS	The location in MASTER of the absorption by region table.
LSCORE	The location in MASTER of the flux scoring array.
LPACK	Not used.
NTOT	NUMSC times (the number of energy bins in the largest supergroup).
LGEOM	The location in MASTER of geometry data.
LEGEOM	The last location in MASTER of geometry data.
KSOUR	The location in MASTER of the source data.

Common CROSS

EPRIM	The energy of the particle after scattering.
ATWT	The atomic weight of the scatterer.
NCDB	An integer variable denoting the types of collision (see table below).

NCDB	Type of Collision
1	Hydrogen scattering
2	Isotropic elastic scattering
3	Anisotropic elastic scattering

NCDB	Type of Collision
4	Scattering in the LAB system no degradation.
5	Compton scattering for gammas
6	Absorption
7	Inelastic continuum scattering
8	Inelastic discrete scattering
CSTHT	The cosine of the angle of scatter.
U	The macroscopic total cross section.
JEN1	The location in MASTER of the first energy of the mesh.
JEN2	The location in MASTER of the last energy of the mesh.
KPHYS	The number of physical compositions.
NENERG	The number of energies in the current supergroup.
NCOUNT	The number of data words in the current supergroup.
NBAND	The number of cross-section supergroups.
EBLX	The lower energy of the cross-section supergroup.

Common METRY

LBASE	The location in MASTER of the geometry data (same as LGEOM).
RLN	The distance from a point $\underline{XB}$, on the ray $\underline{XB} + \underline{WB} \cdot S$, to the first contact with a body.
ROUT	The distance on the ray, from XB, to the last contact with the body.
LRI	The entering surface number.

LRO	The leaving surface number.
PINF	The machine infinity (10^{+50}).
IERR	An error counter for input errors or "G1" errors.
DIST	The distance to the current body measured from XB.
NRPP	The number of rectangular parallelepipeds (input).
NTRIP	The number of triplets (input).
NSCAL	The number of scalars (input).
NBODY	The number of bodies (input).
NNNNN	Same as NRMAX in Common INPUT.
LTRIP	The location in MASTER of the triplet data.
LSCAL	The location in MASTER of the scalar data.
LREGD	The location in MASTER of the region description data.
LDATA	The location in MASTER of the floating point data.
LRIN	The location in MASTER of the RIN for each body.
LROT	The location in MASTER of the ROUT for each body.
LIO	The location in MASTER of the surface numbers for each body.
LOCDA	The location in MASTER of the data for a particular body.
I15	$= 2^{15}$. Used in unpacking data from the MASTER array.

I30	= 2^{30}. Used in unpacking data from the MASTER array.
LBODY	The location in MASTER of the body data.
NASC	The body number of the body being examined by G1.
KLOOP	A counter used by G1.

Common FAP

XAD(25)	The array of X-coordinates for the detectors.
YAD(25)	The array of Y-coordinates for the detectors.
ZAD(25)	The array of Z-coordinates for the detectors.
ID	The number of the detector being processed.
NDFAP	The number of detectors for flux-at-a-point.
NDET	The total number of detectors.
LSCFAP	The location in MASTER of the scoring array for flux-at-a-point.
LPAFAP	Not used.
J12345	A particle type flag (see the discussion of PICK).

Common PUTAD

IRT(10)	Transmission region numbers. Only the first three are used.
IWA	Absorption recording flag. Nonzero for recording.
IWI	Inelastic recording flag. Nonzero for recording.
IWE	Elastic recording flag. Nonzero for recording.

IWD	Degradation recording flag. Nonzero for recording.
IWO	Multiplying factor for transmission recording.
OTHER(5)	Not used.

4.4 PROGRAM IMPLEMENTATION

4.4.1 Operation Instructions

The following discussion describes the operating procedure for the CDC-6600 using the Chippewa* operating system.

The deck configuration consists of a control card deck followed by a data deck. The control card deck is shown below. Note that the card data is to be punched starting in card column 1. A complete discussion of control cards appears in the Chippewa systems manual.

* Control Data 6000 Series Computer Systems, Chippewa Operating System Reference Manual, Publication Number 60134400, 8100 34th Avenue, South Minneapolis, Minnesota 55440

Card	Comments
ID card	See the Chippewa systems manual for a description of this card.
REQUEST TXXX.	XXX is a tape number supplied by the computer center. This tape contains the SAM-C program in binary format.
REWIND (TXXX)	Rewind the program tape.
COPYBF(TXXX,TUNC) COPYBF(TXXX, MONTE)	Read the two sections (TUNC, MONTE) that comprise the SAM-C program into memory.
UNLOAD(TXX)	Return the program tape, it is no longer needed.
REQUEST TYYY, TAPE 11. REWIND(TAPE 11)	YYY is the tape number of the element data tape to be used.
REQUEST TZZZ, TAPE 15. REWIND (TAPE 15)	ZZZ is the tape number of the source tape. Omit these cards if no source tape is used.
REQUEST TVVV, TAPE 14. REWIND (TAPE 14)	VVV is the interaction/transmission tape if required.
TUNC.	Execute the TUNC program which reads and processes all input.
UNLOAD (TAPE 11)	The element data tape is no longer needed.
MONTE.	Execute the Monte-Carlo and edit portions of SAM-C.
UNLOAD (TAPE 14) UNLOAD (TAPE 15)	Release the remaining tapes.
EXIT.	Terminate the problem.

Note that the edit tape (TAPE 16), the supergroup storage tapes (TAPE 9, TAPE 17) and the organized data tape (TAPE 10) were not mentioned explicitly in the above control

cards. These tapes are usually not required for subsequent problems and are assigned to disk storage. The operating system will automatically assign a disk storage area to any tape not mentioned in the control cards.

The following figure (Fig. 4.1) shows the complete deck configuration for a computer run. Note that the "EOR" card is a system end of record card and that "EOF" is a system end of file card.

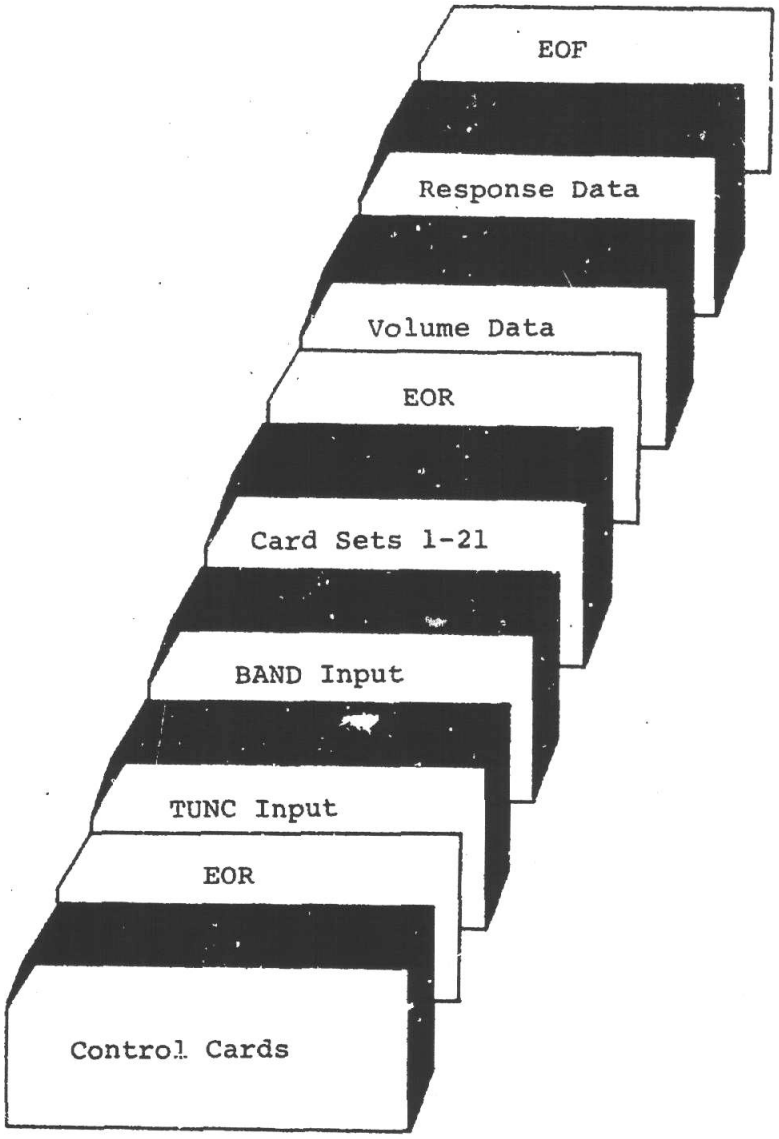


Fig. 4.1 - Deck Configuration for SAM-C

4.4.2 Tape Utilization

The following describes the function of each tape used in the program. Tape numbers refer to FORTRAN logical numbers. All tapes are used in the binary mode except for tape 11, which is BCD mode and contains 80 column card images.

Tape 9

A temporary storage tape used by the BAND program. The tape is also used for temporary storage of particle latents in the PICK routine.

Tape 10

The organized data tape (ODT). The tape contains cross-section data for a given problem. The tape may also hold geometry data if requested by an input option.

Tape 11

The element data tape (EDT). The tape contains a library of available elements. Subroutine BAND uses this tape to get the data for a given problem. BAND is the only routine using this tape. The organization of the EDT is given in Appendix B.

Tape 14

The interaction/transmission tape. All interactions and transmissions are written on this tape for use in subsequent problems. A discussion of format is included in the section on Subroutine WRT14.

Tape 15

An external source tape. The tape is created by the GASP program using tape 14. The tape may also be the transmission part of tape 14.

Tape 16

The statistical aggregate tape. The MONTE routine uses this tape to record each completed aggregate. The edit routines then process the tape to obtain the final flux, dose results.

Tape 17

A temporary storage tape for latents. The tape is used by the PICK routine.

4.4.3 Error Messages

Three types of error indications are given by the SAM-C program. Type 1 errors give an error message and cause the program to terminate. Type 2 errors give an error message, but do not terminate the calculation. Type 3 errors terminate the calculation, but give no error message. The possible error stops and messages are discussed below.

Type 1 Errors

1. SEEK ERROR

   The error occurs in SEEK and is caused by an argument out of range of the vector being searched. The most probable causes are an

input error in the output energy bins or the energy or angular weight bins.

2. OUT OF RANGE ON EBAND

   The error occurs in BAND and is caused by the input cross-section energy band mesh being outside the range of the energies on the EDT. The most probable causes are an input error or using the wrong EDT.

3. NO MORE ELEMENTS ERROR IN BAND

   The error occurs in BAND and is given when an isotope identifier in the BAND input cannot be found on the EDT. The most probable causes are an input error or using the wrong EDT.

4. *** ERROR - BOTH FIRST AND LAST BIN BOUNDARIES MUST BE FLAGGED WITH NEGATIVE SIGNS

5. *** ERROR - EHIGH MUST BE WITHIN ENERGY BINS

6. **** ERROR - ECUT MUST BE WITHIN ENERGY BINS

7. **** ERROR - ETHRM MUST BE WITHIN ENERGY BINS

8. THE NUMBER OF ENERGY BINS IS TOO BIG THE MAX IS

   The error indicates too many energy output bins to fit in the machine. The input must be modified. A suggestion is to allow another output supergroup.

9. ERROR IN NUMSC

   The error occurs when a scoring region, as supplied by the region parameter input, is bigger

than the input value of NUMSC. The input should be checked.

10. NO ROOM FOR DATA

The error indicates that the total room occupied by the input is greater than the allowable maximum. The input must be modified.

Note that errors (4) through (10) all occur in INPUTD. These errors will allow the remainder of the input to be processed but the program will not perform the Monte Carlo calculation.

11. **** ERROR - SPECTRUM NOT DEFINED BETWEEN EHIGH AND ECUT

The error occurs in SOUCAL and indicates that the energy spectrum violated one of the following constraints:

$$ECUT \leq ET(IFLAG)$$

$$EHIGH \geq ET(1)$$

where ET is the input energy spectrum, and IFLAG is the number of entries in the spectrum table.

12. CANNOT HAVE ANG. IMP. FOR ANISTROPIC SOURCE

The error occurs in SOUCAL and indicates that the special source direction option was used in a region containing angular importance. This condition is not allowed.

13. ERROR IN NOUT

The error indicates that a source energy was

calculated by SOUPIC and is outside the range of the output energy mesh. Check the input.

14. BAD IRPRIM IN CARLO

An IRPRIM of zero was calculated in the G1 routine. The geometry input and the source position data should be checked.

15. S1 OUTSIDE BOUNDS

The error indicates that S1, the distance to the next boundary, as calculated by G1, is either zero or greater than 10^{11}. The geometry input should be checked. A more serious cause is an error in compilation.

16. ERROR IN SP IN CARLO

"SO" the distance to the next collision is greater than S1, the total distance through the region. The FORTRAN statements in CARLO should be checked for a compile error.

17. ERROR IN NREG

A composition number of zero or greater than the KPHYS, the number of compositions, was encountered. Check the region input.

18. ERROR IN IRPRIM

A region number, IRPRIM, was greater than the number of regions. Check the region input.

19. ERROR IN ISC

A scoring region number, ISC, was greater than the number of scoring regions. Check the region input.

20. ERROR IN NCDB IN CARLO

An illegal interaction digit, NCDB, has been generated by DR, the interaction routine. The cross-section data as edited by BEDIT should be checked.

4.4.4 Sense Switch Options

The following describes the Sense Switch utilization in the SAM-C program.

Sense Switch 2

Terminate the problem and edit the completed aggregates.

Sense Switch 1

Debug printout for the CARLO, DR, FAP routines.

Sense Switch 4

Debug printout for the geometry dependent routines.

Note that physical sense switches are not available on the CDC-6600. The sense switches are designated by means of control cards and cannot be changed during execution.

APPENDIX A

DISCUSSION OF IMPORTANCE SAMPLING

It is fairly well known[1] by now that Monte Carlo calculations can be carried out so that a single linear functional (e.g., dose, Fe activation) of the radiation field may be calculated with zero variance. To do this requires pre-knowledge of an "importance function" which gives for any particle its expected contribution to that single functional. This importance function is used to bias histories of particles so as to direct them toward those positions, energies, and directions at which the chance of contributing to an answer is large. In addition, to get strictly zero variance it is necessary to compute an answer (to "score") only when an importance modified history is terminated.

To be more explicit, let $\underset{\sim}{x}$, E, $\underset{\sim}{\Omega}$ denote the position, energy, and direction of a particle and let $J(\underset{\sim}{x}, E, \underset{\sim}{\Omega})$ be the expected total answer eventually given by that particle. Then if $K(\underset{\sim}{x}', E', \underset{\sim}{\Omega}' \rightarrow \underset{\sim}{x}, E, \underset{\sim}{\Omega})$ is the density of

particles coming out of collisions at $\underset{\sim}{x}$, E, $\underset{\sim}{\Omega}$ given that one came out of collision at $\underset{\sim}{x}'$, E', $\underset{\sim}{\Omega}'$ define the altered kernel $K(\underset{\sim}{x}', E', \underset{\sim}{\Omega}' \rightarrow \underset{\sim}{x}, E, \underset{\sim}{\Omega}) = \frac{J(\underset{\sim}{x}, E, \underset{\sim}{\Omega})}{J(\underset{\sim}{x}', E', \underset{\sim}{\Omega}')} x$

$\tilde{K}(\underset{\sim}{x}', E', \underset{\sim}{\Omega}' \rightarrow \underset{\sim}{x}, E, \underset{\sim}{\Omega})$. This kernel is to be used in generating histories if zero variance is to be obtained. Another requirement is that the source density $S(\underset{\sim}{x}, E, \underset{\sim}{\Omega})$ be replaced by

$$\tilde{S}(\underset{\sim}{x},E,\underset{\sim}{\Omega}) = J(\underset{\sim}{x},E,\underset{\sim}{\Omega})\ S(\underset{\sim}{x},E,\underset{\sim}{\Omega})/J(\underset{\sim}{x}',E',\underset{\sim}{\Omega}')\ S(\underset{\sim}{x}',E',\underset{\sim}{\Omega})d\underset{\sim}{x}'dE'D\underset{\sim}{\Omega}'$$

in selecting starting coordinates for histories.

In practice, the conditions for obtaining zero variance are never achieved for any practical problem. It is useful to note the reasons.

1. The exact importance function is never known.
2. Scoring is not carried out in the manner required for zero variance.
3. Approximate importance functions cannot be specified in enough detail.
4. General purpose codes do not carry out the history sampling using the altered kernel $\tilde{K}$, but some alternative (unbiased) form such as splitting at boundaries. This always increases the variance.

The first point is a difficulty in principle. In practical problems, it has been found that approximate forms for J can reduce the variance in difficult problems

to the point where modern computing machines can achieve satisfactory results. Several methods have been used to establish the function J. Kalos[2] used a multicollision approach (essentially consideration of the Neumann series of the adjoint problem) and was able to calculate very deep penetration. At the other extreme, Cain[3] has recently used S_n codes to find numerical values of J. Unfortunately for two-or three-dimensional problems this requires computations as extensive, or more so than the Monte Carlo calculations themselves. Finally, it is possible to use existing results for similar problems (e.g., moments method results for gamma rays) as one does in obtaining rough engineering approximations to shielding results themselves. This appears to be the most practical and will be illustrated later.

It has long been suspected[2] that the essential part of the variance reduction is the use of a properly biased altered kernel and that the particular scoring scheme is less important. This has, in effect, been shown recently.[4]

In many codes the importance sampling depends only upon space or occasionally on energy. This, coupled with the fact that extra variance arises in sampling the altered kernel, may well have the effect that the optimum parameters cannot be easily deduced from even a reasonable guess for J. In particular, the failure to alter the source distribution consistent with the history biasing requires special

treatment.

The properties of SAM-C are such that the optimum use of the code should result by using a set of weights inversely proportional to $J(\underline{x}, E, \Omega)$. These weights can depend upon position, energy, and direction. The source spectrum is altered automatically according to the weights. Finally, the biased histories, particularly the flight of a particle, are carried out so as to provide little additional variance.

Thus establishing efficient sampling reduces to calculating a reasonable approximation to J. Before giving an example, certain general remarks can be made.

First, because the importance function is specified through weights that are constant in spatial regions, the requirements of biasing reflect upon the geometry used to describe the problem. Thus an effort should be made to provide enough regions so that in significant regions of space, and for the part of the spectrum responsible for penetration to the detector, the variation in average importance from one region to the next is no more than (very roughly) a factor of four. This requirement can be relaxed for regions far from source and detector or for radiation too weakly penetrating or too weak to enter into the further penetration.

The use of appropriate symmetries simplifies the importance sampling. Although, in principle, a result can be obtained as easily in a small region as in a shell in a

spherically symmetric problem, in practice extra care must be given to focus the radiation appropriately toward a small detector. Also the importance function will contain geometric (e.g., r^{-2}) factors. For a shell surrounding a spherical source, only the exponential attenuation with position is needed.

When a reasonable set of weights has been given for a set of detectors in a given geometry, it can be used for a variety of sources. Thus a good set of importance weights for gamma rays in a given situation can be used for monoenergetic gammas, for a prompt fission gamma source, or for spatially and energy-distributed gammas arising from neutron interactions. Naturally, if space or energy regions previously thought unimportant are no longer so, extensions must be made.

In some problems it is necessary (or thought necessary) to obtain several different answers in a single computer run. If these answers depend essentially upon the same or very similar histories, (as for example, the biological dose at two detectors separated by less than a relaxation length for the dominant radiation), then a biasing scheme for one (or for a detector at a mean position) does reasonably well for the other. When rather different histories are required (as for widely separated detectors or for neutron dose contrasted with sources of secondary gamma radiation) then it may well be true that separate runs with separate importance

weights will be computationally most efficient. In intermediate cases where it is suspected that the computational gain from running a single problem is important, the following prescriptions are useful.

1. Take weights proportional to flux over the important range of position and energy. This has the effect of giving generally good statistics for flux over that range (though likely at the expense of computing time). The numerical values of flux may be taken from related problems or from a preliminary run.
2. Take the spatial dependence of weight proportional to the expected dose over regions of adjacent detectors. Outside regions of detectors compute relative importance for nearest detector.
3. Take a linear combination of the importance function for single detectors, the coefficient being inversely proportional to the estimated result at that detector. The weight is, of course, inversely proportional to the compositive importance.

Note that in SAM-C the magnitude of the weight is irrelevant.

Finally we give the following warning. It is often tempting to try a rather sharp biasing, particularly in direction. It should always be kept in mind that the im-

portance function is the expected answer after all future collisions, not just a single one. Thus a particle pointed within some neighborhood of a detector (e.g., within a cone that passes as close as roughly a mean free path for scattered radiation) is likely to be nearly as important as a ray that passes directly through. Neglect of this leads to biasing in which the most probable result, except in very long sampling, omits this scattered radiation. In somewhat the same way, radiation directed away from a detector may, in consequence of multiple scattering, be nearly as important as radiation directed toward a detector. When diffusion dominates, there is little direction effect.

The last section will describe the procedure used to set up weights for gamma-ray transport in the atmosphere above a ground interface.

It was required to calculate free air gamma-ray doses in two sets of detectors. The first set consisted of detectors centered at 250, 500, 750 meters and the second at 1000, 1250, 1500 meters. Each set was considered a separate problem. Each 'detector' was in fact a ring of air, 7 meters high, and whose inner and outer radii differed by 20 meters. The rings abutted the ground.

The air was subdivided into large concentric rings, 600 meters high, centered on the source. The increment in radius was taken as 125 meters; this is adequately small compared with the average relaxation length of about 600

meters. Each detector ring was contained in one of the larger rings.

The energy grouping was taken as follows. Energies above 4 Mev were lumped together. Other divisions were taken at 2, 1, 0.5, and 0.2 Mev. Except below the last, the variation in cross-section in a group is less than about 30% so that the importance is reasonably taken to be constant in each group. The properties of gammas were evaluated at the lower end except for the lowest, for which 0.1 Mev was used.

The procedure for establishing weights was the following. Assume we are dealing with the second group of detectors. The effectiveness of a gamma was taken (for calculating importance only) to be proportional to energy. At 1000 meters therefore, the weight was set equal to the reciprocal of the group energy in Mev. From 1000 to 1500 meters the weight was assumed to decrease exponentially (i.e., the importance increased exponentially) with a relaxation length of 550 meters, taken from an earlier calculation as characteristic of dose. The same ratios in weights were used in each energy range. The importance at 500 and at 1000 were evaluated as $(E\ B(E)r)e^{-\mu(E)r})^{-1}$ using r = 500 and 1000 meters, respectively. These are, of course, the distances to the first detector. Buildup factors B(E) were taken for point isotropic sources from Goldstein's book[5] except at the lowest energies where very rough extrapolations were made. At points 500 meters and 1000 meters beyond the last

detector, the weights were made to rise in the same ratios. These points were plotted on semilog paper and curves drawn joining them. This is shown in Fig. A-1. Rough average weights are then taken directly from the graph.

It is easy to see from this prescription that, for detectors in the range 250 to 750 meters, one may use the same curve by shifting the entire curve left by 750 meters. This is rather generally applicable.

No angular weights were used, primarily because the regions are so large that the assumption of constant preferred direction is not applicable. For this reason (among others) it would have been preferable to subdivide the vertical section by several parallel planes. Since satisfactory answers were obtained in short computing times we did not go on in this direction for gammas, although significant improvements were found in using more vertical divisions in neutron problems.

The weights in ground were always taken to be four times the corresponding weights in the air above. Gammas histories were terminated above 600 meters or beyond 1900 meters from the source.

REFERENCES FOR APPENDIX A

1. Kalos, M. H. and Goertzel, G.: "Monte Carlo Methods in Transport Problems," Progress in Nuclear Energy, p. 315, Pergamon Press, New York, 1958.
2. Kalos, M. H.: "Nuclear Science and Engineering," Vol. 16, p. 227 (1963).
3. Cain, V. R.: "Transactions of American Nuclear Society," San Diego Meeting, June 1957.
4. Coveyou, R. R., Cain, V. R., and Yost, K. J.: "Nuclear Science and Engineering", Vol. 27, p. 219 (1967).
5. Goldstein, H.: Fundamental Aspects of Reactor Shielding , Addison-Wesley Publishing Co., 1959.

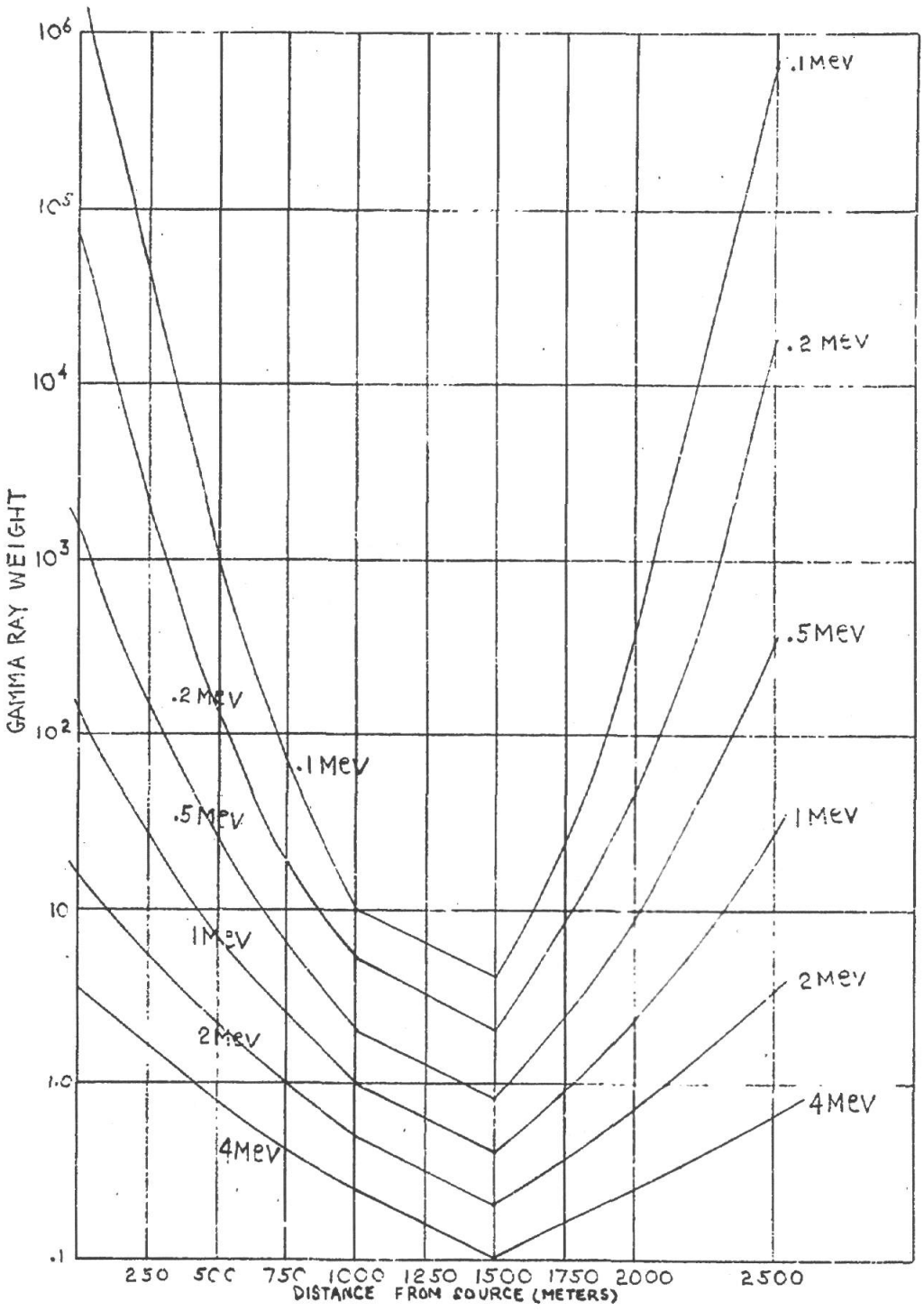


Fig. A-1 - Gamma Ray Weights vs Distance from Source for Detectors in 1000 to 1500 Meter Range

APPENDIX B

FORMAT OF THE ELEMENT DATA TAPE (EDT)

A description of the format of the EDT follows. Since the EDT is made up of card images, the description is in terms of card images.

Item No.	No. of Entries	Card Description	Format
1	1	NENERG NENERG is the number of energies in the energy table.	I10
2	NENERG	ETABLE(1), ETABLE(2), ETABLE(3), ETABLE(4), ETABLE(5), ... ETABLE(NENERG). ETABLE is the energy mesh for all elements. ETABLE(1) is the lowest energy and ETABLE(NENERG) is the highest energy.	5E14.5
3	3	AWT, IAWT, J AWT is the floating point atomic weight. IAWT is the fixed point identifier. J is an end of data flag. J ≠ 0 means this card is the last card of the EDT.	E16.8, 2I6
4	5	OF, NOW, NFI, ISI, IPIN OF ≠ 0 means this is last card of element.	E16.8, 4I6

Item No.	No. of Entries	Card Description	Format
4 con'd.	5	NOW is the number of words in the next group of data. NFI is a file number used for identification only. ISl ≠ 0 means the next group of data is the scatter index. IPIN ≠ 0 means this element has no inelastic scattering.	
5	NOW	SIGMA(1), SIGMA(2), SIGMA(3), SIGMA(4), etc. SIGMA is the microscopic total cross section. Note that NOW must equal NENERG and that OF, ISI, IPIN must all be zero.	5E14.5
6	5	Repeat item 4.	E16.8, 4I6
7	NOW	PSCAT(1), PSCAT(2), PSCAT(3), PSCAT(4), PSCAT(5), ..., PSCAT(NENERG). PSCAT is the probability of elastic scattering. PSCAT(1) corresponds to ETABLE(1) and PSCAT(NENERG) corresponds to ETABLE(NENERG).	5E14.5
8	5	Repeat item 4.	E16.8, 4I6
9	NOW	PABS(1), PABS(2), PABS(3), PABS(4), PABS(5), ..., PABS(NENERG).	5E14.5
10	5	OF, NOW, NFI, ISI IPIN ISI must equal 1 because the next data group is the scatter index. If this element has no inelastic scattering, then IPIN must equal 1.	E16.8, 4I6
11	NENERG	IP, ID, IT, IA There are NENERG cards in this data group. Each card has the quantities IP, ID, IT, IA defined as follows:	4I10

Item No.	No. of Entries	Card Description	Format
11 con'd.	NENERG	IP = 1 for inelastic discrete scattering = 2 for inelastic continuum scattering ID = the location of an ENN or PLEV table relative to the first word of the SIGMA table IT = 1 for isotropic scattering in center of mass = 2 for scattering in hydrogen = 3 for anisotropic scattering in center of mass = 4 isotropic scattering in the lab system IA = location of CHI Table if IT = 3.	4I10
12	5	Repeat item 4 with NOW = 11.	E16.8, 4I6
13	11	CHI(1), CHI(2), CHI(3), CHI(4), CHI(5), etc. CHI is a table of 11 entries. CHI(1) = 1.0, CHI(11) = 0. Items 12, 13 are repeated for each energy at which "IT" of item 11 is 3.	
14	5	Repeat item 4 with NOW = 11.	E16.8, 4I6
15	11	ENN(1), ENN(2), ENN(3), ENN(4), ENN(5), etc. The ENN table is used to determine the energy after scattering for an inelastic continuum interaction. Items 14 and 15 are repeated for each energy at which "IP" of item 11 is 2. If the element has no continuum scattering items, 14 and 15 are omitted.	5E14.5
16	5	Repeat item 4 with NOW = number of excitation levels for discrete scattering.	E16.8, 4I6

Item No.	No. of Entries	Card Description	Format
17	NOW	ELEV(1), ELEV(2), ELEV(3) ELEV(4), ELEV(5), ..., ELEV(NOW). The ELEV table is a list of possible excitation levels; each entry corresponds to a probability in a PLEV table.	5E14.5
18	5	Repeat item 16.	E16.8, 4I6
19	NOW	PLEV(1), PLEV(2), PLEV(3), PLEV(4), PLEV(5), ..., PLEV(NOW). The PLEV tables are tables of probabilities of scattering from the current energy to the excitation levels. Items 18 and 19 are repeated for each energy at which inelastic discrete scattering occurs (IP = 1).	5E14.5
20	5	OF, NOW, ISI, IPIN OF = 1.0. NOW, ISI, IPIN = 0. Item 20 is the last card of an element.	E16.8, 4I6
21	3	AWT, IATWT, J If there are no more elements AWT, IATWT = 0. J = 1. If there are more elements repeat from item 3.	

APPENDIX C

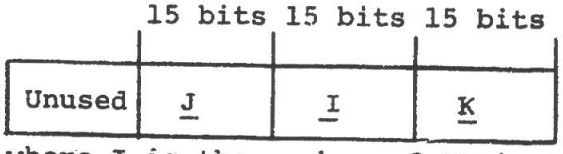
ORGANIZATION OF THE GEOMETRIC PORTIONS OF MASTER ARRAY FOR MAGIC

The MASTER storage array has a complex organization of integer and floating point data. The integer portions are, in general, packed three items per word and fifteen bits per item.

The following explains the construction of the table.

1. RPP DATA DESCRIPTIONS

   Stored six words per RPP beginning at LBASE (for MAGIC, LBASE = 1 and for SAM-C, LBASE = LGEOM). Each word has three numbers.



   Fixed point where I is the number of RPP's that abut this surface, J is the location within the MASTER array of the list of these RPP's, K is the

Address In MASTER (AIM) where the floating point number corresponding to this surface is stored.

2. RPP DATA

X	floating point

Full word

These are the numbers read in RPP Data cards and referred to as K in Item 1.

3. BODY DATA DESCRIPTORS

Each body data descriptor consists of two computer words. Each of the two computer words contains three integers as shown below.

	15 bits	15 bits	15 bits
Not used	I	J	K
Not used	L	M	N

Fixed point

where:

I is the number of possible regions to be entered when entering this body.

J is the number of possible regions to be entered when leaving this body.

K is the type of body (0-8 for RPP-ARB)

L AIM of a list of I

M AIM of a list of J

N AIM of the body data pointer for this body (see Item 4 below)

Note that the variable LBODY (in Common) is the AIM of the first word of the first body data descriptor.

4. BODY DATA POINTERS

This information is referenced by N in Item 3 and is of variable size depending on the type of body. Each body has between two and four parameters and they are stored as follows.

	15 bits	15 bits	15 bits	
Not used	I	J	K	4 parameters
Not used	Not used		L	

Not used	I	J	K	3 parameters

Not used	I	J	2 parameters

In all cases the parameters are fixed point and are AIM's to the floating point data, which were originally read in for the bodies.

5. REGION DESCRIPTORS

There are NRMAX region descriptors of one computer word each, as shown below.

	15 bits	15 bits	
Not used	I	J	Fixed point

where:

I is the number of bodies in the region description.

J is the AIM of a list of the bodies.

Note that LREGD (in Common) is the AIM of the first region descriptor.

6. BODY DATA

These items are one computer word in length and contain the floating point data referred to in Item 4.

7. REGION DATA

	15 bits	15 bits	
Not used	I	J	Fixed point

These are the body numbers referred to in Item 5. I is an operator (+, −, or OR) converted to a numeric code between 1, 8.

J is the actual body number. These data are from the region description input.

8. ENTERING AND LEAVING TABLES

	15 bits	
Not used	I	Fixed point

These are the entering and leaving tables and are referenced by IT and L in Item 4.

9. LRIN is the AIM of a table where the code will store RIN as computed to each body. There are NBODY floating point words.

10. LROT is the AIM of a table to store ROUT as computed for each body.There are NBODY floating point words.

11. LIO is the AIM of a table where the code will store three fixed point items per body, as shown.

I	J	K

Fixed point

where:

I is the entering surface number.

J is the leaving surface number.

K is the index of or ray number.

There are NBODY words.

12. Region ID and space codes. This is a table of NRMAX fixed point words whose AIM is LIRFO = NDQ -10-NRMAX, where NDQ is the size of the MASTER array. At LIRFO + REGION NO -1 is stored the two items in fixed point format

Not used	I	J

Fixed point

where:

I is the region ID.

J is the space code + one.

APPENDIX D

ORGANIZATION OF THE MASTER ARRAY FOR SAM-C

The following section describes the organization and contents of the MASTER storage array. The capitalized name in the upper right-hand corner of each box refers to a "pointer". These "pointers" are used to locate sections of data with a minimum of calculation.

	Contents	Comments
(1)	Cross section data	Calculated by BAND and read in from the ODT by MONTE.
(2)	LGEOM Geometry data	LGEOM. The data are read in and stored by GENI.
(3)	LSCORE Region flux scores for each statistical aggregate	The scores are accumulated and stored by CARLO. The scores are stored as a function of energy and scoring region.
(4)	LREGT Region-dependent parameters	This section of the array contains region-dependent parameters stored two computer words per region. The actual data are discussed in the UNPR section of this report. The data are read in and stored by INPUTD.

	Contents	Comments
(5)	LNCOL Number of collisions per region	
(6)	LBIRTH The number of births per region	
(7)	LDEATH The number of deaths per region	
(8)	LESCAP The number of escapes per region	
(9)	LNDEG The number of degrades per region	
(10)	LNABS The number of absorptions per region	

For items (5) through (10) the starting locations are computed in INPUTD. The counts are accumulated in CARLO and are printed by TALLY.

(11)	LSCFAP Flux-at-a-point for all detectors	The scores are computed by CARLO. This section of the array will exist only if flux-at-a-point detectors are present in the problem.

	Contents	Comments
(12)	LRAW Region weights	The actual region weights to be used for region importance. The weights are read in by INPUTD.
(13)	LREW Region energy weight sets	The energy weight sets for importance sampling. This section exists if energy importance is used in the problem. The data are read in by INPUTD.
(14)	LAIM Aiming angles	The aiming angles for angular importance. Three words per angle denoting direction cosines are stored. The data are read by INPUTD. The array exists only if angular importance is used.
(15)	LAWS Region angular weight sets	The angular weight sets for angular importance. The array exists only if angular importance is used. The data are read by INPUTD.
(16)	KSOUR Source data	The energy, position, and direction data for the source distribution. The data are read in by SOUCAL.
(17)	LLAST Latent storage for supergroups	This section uses up all available room in the MASTER array. Supergroup latents are stored here. Tapes will be used for latent storage if the available room is insufficient.

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*

Mathematical Applications Group, Inc.
180 South Broadway
White Plains, New York

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

A GEOMETRIC DESCRIPTION TECHNIQUE SUITABLE FOR COMPUTER ANALYSIS OF BOTH NUCLEAR AND CONVENTIONAL VULNERABILITY OF ARMORED MILITARY VEHICLES

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Final Report

5. AUTHOR(S) *(First name, middle initial, last name)*

Walter Guber, Roger Nagel, Robert Goldstein, Phillip S. Mittelman and Malvin H. Kalos

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
August 1967	283	5

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
DAAD05-67-C-0041 b. PROJECT NO.	MAGI-6701
c. d.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

Approved for public release; distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	U.S. Army Aberdeen Research and Development Center Aberdeen Proving Ground, Maryland

13. ABSTRACT

A suitable three-dimensional geometrical description technique and two computer programs, MAGIC and SAM-C, were developed and demonstrated for application to the computer analysis of both the nuclear and conventional vulnerability of armored military vehicles. The geometric technique was applied, for demonstration purposes, to the M60A1 tank. A single body of geometric input target data for the M60A1 was used by the MAGIC program to address the tank from any attack angle by conventional armor-defeating projectile systems, and by the SAM-C Monte Carlo program for the determination of internal radiation dose. The radiation dose (in selected target regions) is obtained for primary neutrons, primary gammas, and secondary gammas produced by neutron interactions occurring either outside or within the vehicle. The two programs, MAGIC and SAM-C, are currently operational on both the CDC-6600 and BRL-BRLESC computers. Both programs are written in FORTRAN.

DD FORM 1 NOV 68 1473 REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

Unclassified

Security Classification

Unclassified
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Target Analysis						
MAGIC						
SAM C						
Operations Research						
Systems Analysis						
Nuclear Radiation Transport						
Computer Programs						
Radiation Shielding						
Armored Combat Vehicle Vulnerability						
Target Description						

Unclassified
Security Classification