

AD-A051 057

ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND ABERD--ETC F/G 6/21  
RAYMAN: A FORTRAN COMPUTER CODE FOR TRACING RAYS THROUGH A DETA--ETC(U)  
NOV 77 W B BEVERLY  
ARBR-TR-2030

UNCLASSIFIED

NL

| OFR |  
AD  
A051 057



AD A 051057

ARBRL-TR-2030

# BRL

12

Q

AD

TECHNICAL REPORT ARBRL-TR-2030

RAYMAN: A FORTRAN COMPUTER CODE FOR TRACING RAYS THROUGH A DETAILED HUMAN PHANTOM

William B. Beverly

November 1977

AD No. \_\_\_\_\_  
DDC FILE COPY

DDC  
RECEIVED  
MAR 10 1978  
B

Approved for public release; distribution unlimited.

USA ARMAMENT RESEARCH AND DEVELOPMENT COMMAND  
USA BALLISTIC RESEARCH LABORATORY  
ABERDEEN PROVING GROUND, MARYLAND

Destroy this report when it is no longer needed.  
Do not return it to the originator.

Secondary distribution of this report by originating  
or sponsoring activity is prohibited.

Additional copies of this report may be obtained  
from the National Technical Information Service,  
U.S. Department of Commerce, Springfield, Virginia  
22161.

The findings in this report are not to be construed as  
an official Department of the Army position, unless  
so designated by other authorized documents.

*The use of trade names or manufacturers' names in this report  
does not constitute indorsement of any commercial product.*

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 74 TECHNICAL REPORT ARBRL-TR-2030	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) RAYMAN: A FORTRAN COMPUTER CODE FOR TRACING RAYS THROUGH A DETAILED HUMAN PHANTOM		5. TYPE OF REPORT & PERIOD COVERED Final rept.
7. AUTHOR(s) William B. Beverly		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Ballistic Research Laboratory Aberdeen Proving Ground, Maryland 21005		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Materiel Development & Readiness Command 5001 Eisenhower Avenue Alexandria, VA 22333		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 16 1L662618AH80
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE NOVEMBER 1977
		13. NUMBER OF PAGES 65 1263 p.
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A
16. DISTRIBUTION STATEMENT (of this Report)  Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)  Computer Man                      Shotline Ballistic Vulnerability          Projectile Ballistic Threat		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  A FORTRAN computer program RAYMAN has been written that simulates an average human adult male in a hostile ballistic environment. Projectiles are picked from a user-chosen ballistic threat and those striking the phantom human are tracked to their exit from his body. Ballistic vulnerability factors, associated with the physiological detail of the phantom, are retrieved at regular intervals along the shotlines of the impacting projectiles. The retrieved vulnerability factors are averaged over many shotlines to yield a histogrammic		

D D C  
RECEIVED  
MAR 10 1978  
RECEIVED  
B

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

393 471 |

*Illu*

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

representation of the average vulnerability factor versus tissue penetration.  
Results are printed in tabular form along with identifying captions. ←

2

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

TABLE OF CONTENTS

	Page
I. INTRODUCTION . . . . .	5
II. CODE CAPABILITY . . . . .	7
III. SUBROUTINES AND VARIABLES . . . . .	19
IV. RAYMAN INPUT . . . . .	28
V. TEST PROBLEM . . . . .	31
VI. CONCLUSIONS . . . . .	31
ACKNOWLEDGEMENT . . . . .	34
APPENDIX . . . . .	35
DISTRIBUTION LIST . . . . .	63

ACCESSION for		
NTIS	White Section	<input checked="" type="checkbox"/>
DDC	Buff Section	<input type="checkbox"/>
UNANNOUNCED		<input type="checkbox"/>
JUSTIFICATION _____		
BY		
DISTRIBUTION/AVAILABILITY CODES		
Dist.	AVAIL and/or	SPECIAL
A		

## I. INTRODUCTION

A description of a standard man (the Computer Man)<sup>1</sup> has been built by the Target Assessment Branch (TAB), Vulnerability/Lethality Division (VLD), Ballistic Research Laboratory, using the properly-chosen, horizontal, anatomical cross sections (slices) collected by Eycleshymer and Schoemaker<sup>2</sup> (see Figure 1). These slices have been uniformly gridded so that the physical detail of the Computer Man can be referenced by a Cartesian lattice. Current descriptions assume that all tissue in a cell is the principal tissue in that cell. The two currently-used lattices have a 0.5 cm and 1.0 cm horizontal grid respectively and both descriptions use 2.6 cm thick lower-body slices and 1.2 cm thick upper-body slices.

A phantom of the man, when used for incapacitation studies, is assumed to be standing in a box that barely encloses him. The box is placed in the first octant of a Cartesian coordinate system so that the sides of the box are parallel to the coordinate-axis planes. A bottom corner of the box is located at the coordinate-system origin so that the phantom faces the positive-y direction.

A three-dimensional matrix is constructed whose elements reference the Computer Man cells. The matrix is currently filled with the mortality rankings gathered by Cooper<sup>3</sup> but it can be used to store other, user-chosen quantitative descriptions. The indices of the cell enclosing a point in the phantom man may be calculated and used to retrieve the corresponding matrix element.

The FORTRAN Computer Code RAYMAN can generate rays whose origins and directions are picked from user-selected distribution functions and project them into a phantom of the Computer Man. Promenades are conducted along the rays as they penetrate tissue and a point is picked uniformly along each step. The mortality ranking,  $M$ , corresponding to the cell enclosing the point, is retrieved and added to those retrieved for the same tissue penetration,  $R$ , along earlier rays. A histogrammic representation of the mean mortality ranking,  $\bar{M}$ , versus  $R$  is calculated at the end of a statistically meaningful sampling of rays and the results are printed in tabular form. This output may be used to assess personnel vulnerability to the ballistic threat as discussed by Kokinakis and Bruchey.<sup>4</sup>

- <sup>1</sup>. Charles A. Stanley, Michael Brown, "The Computer Man," to be published as a BRL Report.
- <sup>2</sup>. Albert C. Eycleshymer, Daniel M. Schoemaker, "A Cross-Section Anatomy," D. Appleton-Century Company, Inc., New York, London 1938.
- <sup>3</sup>. Walter R. Cooper, William Kokinakis, "Vulnerability Rankings of Human Tissue," to be published as a BRL Report.
- <sup>4</sup>. William Kokinakis, William J. Bruchey, "An Engineering Approach to the Assessment of Personnel Vulnerability," American Defense Preparedness Association, October 1975.

<u>SLICE NO.</u>	<u>ANATOMICAL SECTION</u>	<u>DISC RECORD NO.</u>
( 1 - 6 )	FEET	1
( 7 - 33 )	LEGS	2,-6
(34-39)	ARMS, LEGS	6,-7
(40-59)	ARMS, TORSO	7,-10
(60-64)	TORSO	10,-11
(65-82)	HEAD	11,-14
(83-84)	DUMMY SLICES	14

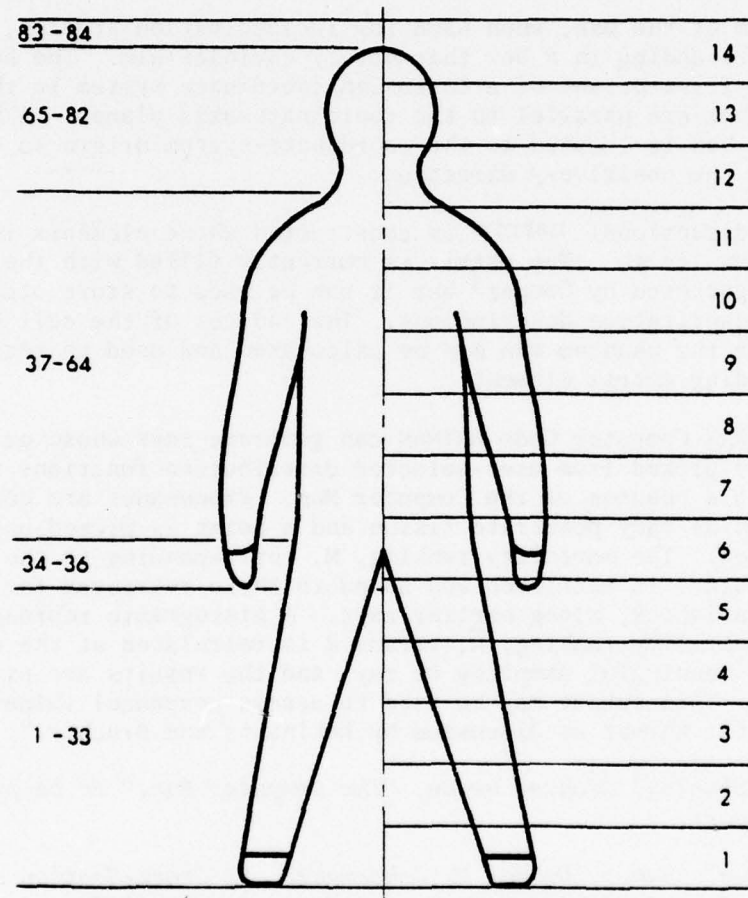


Figure 1. The Computer Man Layout

## II. CODE CAPABILITY

Simulated projectiles (rays) can be started from origins whose locations are picked from a user-selected choice of nine different distributions. The distributions are:

1. User-supplied coordinates. The present dimensions of the storage arrays permit the user to submit a maximum of 100 points.
2. Points generated uniformly on the overhead hemisphere. The user will choose the radius for all hemispherical origins.
3. Points generated uniformly on the underfoot hemisphere.
4. Points generated uniformly on the  $X > 0$  hemisphere.
5. Points generated uniformly on the  $X < 0$  hemisphere.
6. Points generated uniformly on the  $Y > 0$  hemisphere.
7. Points generated uniformly on the  $Y < 0$  hemisphere.
8. Points generated uniformly on a horizontal circle parallel to the plane on which the man is standing. The user will provide the height and radius of the circle.
9. Points generated on an overhead, quarter circle arc. The plane of the circle will include the Z-axis and make a user-supplied aspect angle with the XZ plane. Points will be picked from a user-chosen distribution. The two possible distribution functions are, (a) uniform, and (b) the sine of the polar angle of the radius vector of the point (see Figure 2).

The directions taken by the rays are picked from a user-selected choice of three different angular distributions. These distributions are:

1. Isotropic about the point of origin. In practice, the direction of the ray is usually chosen so that it falls within the solid angle subtended by the sphere enclosing the man box. A normalization factor, that is the quotient of the solid angle subtended by the sphere divided by  $4\pi$ , is calculated and printed in the output.
2. Bivariate normal about the line from the ray origin to a user-supplied point within the man. This capability was provided to the code to simulate the marksmanship of an individual aiming a weapon toward a point on the man. The horizontal and vertical standard deviations of the marksman's efforts in degrees are provided by the user.

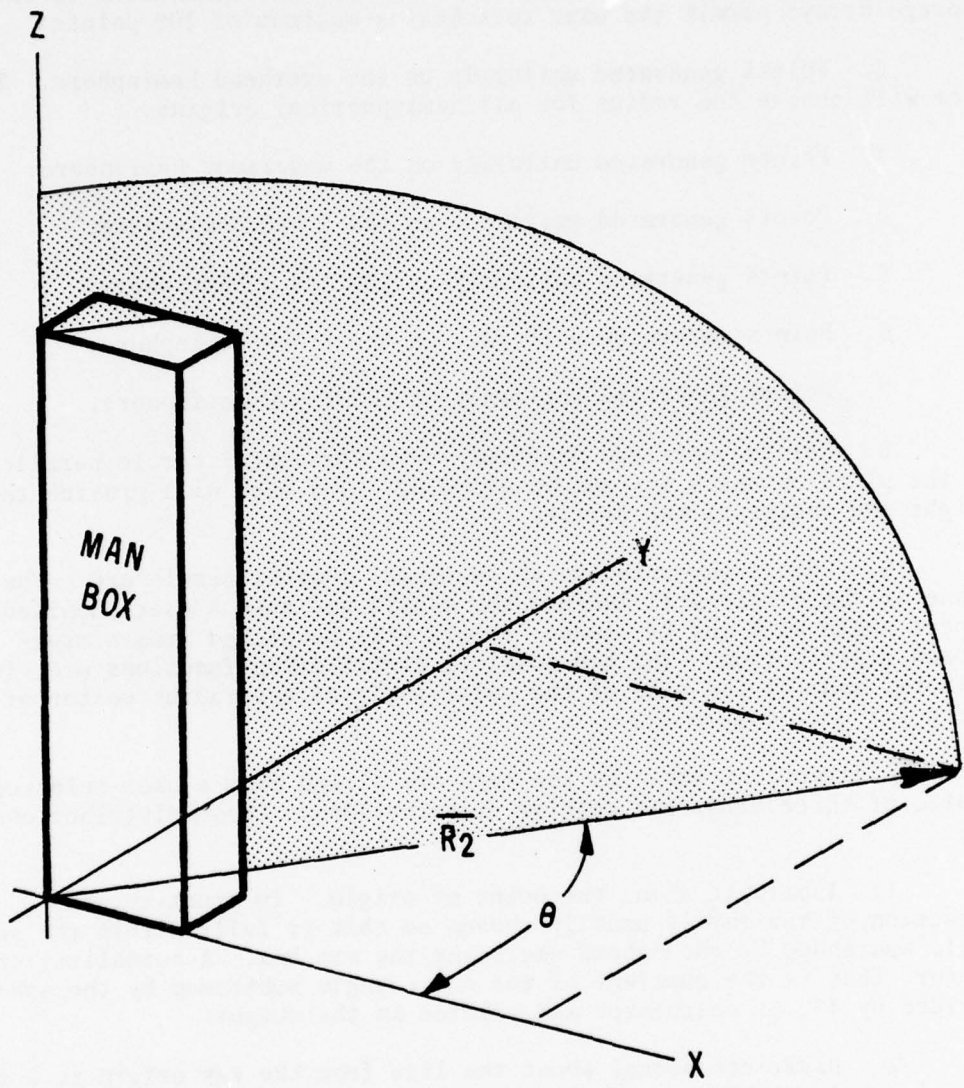


Figure 2. Typical Quarter-Circle Arc

3. User-supplied direction cosines. The code will automatically restrict the problem to one history per point when this option is exercised.

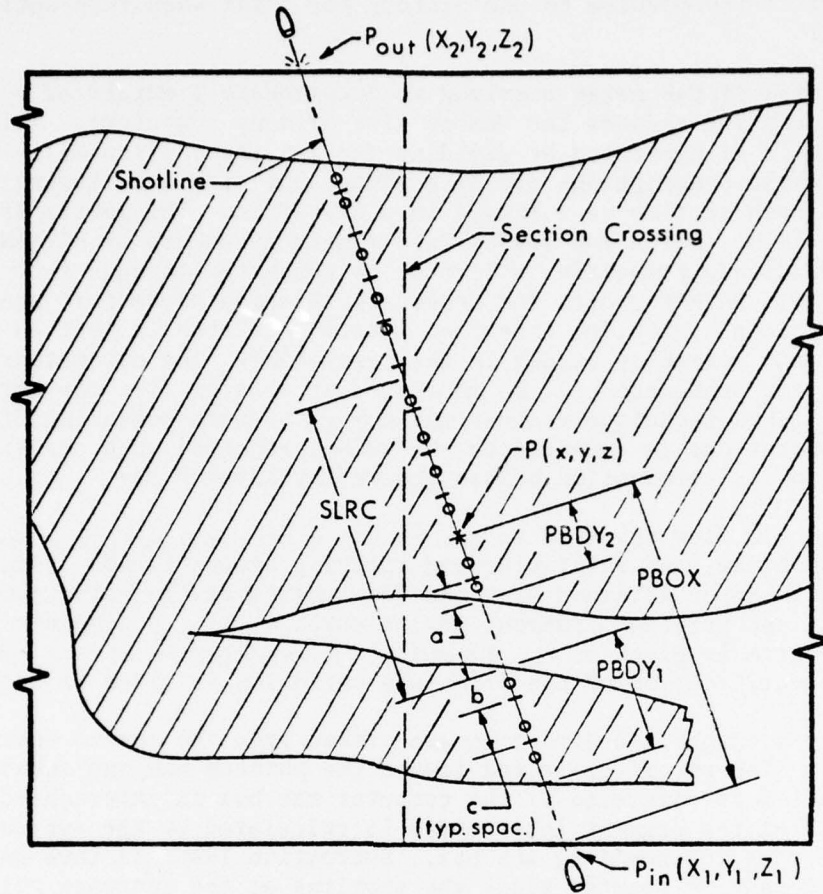
The size of the array required to accommodate a matrix of a complete description exceeds the memory size of many computers. This limitation is circumvented by dividing the man into horizontally-bounded sections whose descriptions are of a manageable size and storing the matrix of each section as a record in a disc file. The matrix of a section will be chosen and loaded into memory as needed by RAYMAN generated commands and promenades will then be conducted through that part of the phantom. RAYMAN stores the necessary tracking parameters when a promenade along a shotline enters an inactive section (a section whose matrix is not currently loaded in the array MAN). The parameters of such a latent promenade will be retrieved at a later time for further tracking. This technique reduces the number of time-consuming disk loading operations as compared to the number required when tracking each shotline to its termination before generating another ray.

The logic flow involved in generating a ray, conducting a promenade along a shotline, and retrieving and tallying scores is now discussed. More detail can be obtained from the flow charts of the main routine (Figure 4) and principle subroutines (Figures 5 -9). A complete listing of the program is given in the Appendix. A description of the subroutines and the identification of the principle variables is given in Section III.

A ray's origin and direction are picked from the chosen density functions. The ray is projected toward the phantom man and subroutine BOX is called to determine if the computer man box is intercepted. The length SLR of the segment intercepted is calculated by the subroutine for those rays intercepting the box. Subroutine TRACK is then entered and a promenade is started along the shotline at the entrance point of the ray if the section of the man being entered is the section whose matrix is currently stored in the array MAN. The promenade step width is set to the histogram bin width when a promenade is started. The parameters of the promenade are stored in latent storage if the matrix stored in MAN does not belong to the section being entered.

The promenade is continued inside the box until tissue is encountered. The distance traversed through the box  $PB\emptyset X$  is incremented after each step and compared to SLR to determine if another full-width step in the box is possible. The distance to a section crossing SLRC is calculated by subroutine CR $\emptyset$ SS to determine if another full width step in the active section is possible (see Figure 3).


The first score is made when the promenade enters tissue. The score is estimated stochastically when the complete length of that step is in tissue. This is accomplished by picking a point with equal probability at any site along the step and retrieving the M-value from the array MAN



$$SLR = [(x_2 - x_1)^2 + (\overline{XL_2} - 0)^2 + (z_2 - z_1)^2]^{\frac{1}{2}}$$

$P(x, y, z)$  = Current Position Along Promenade.

$$PBDY = \overline{PBDY_1} + \overline{PBDY_2}$$

 = Tissue.

o = Scoring Point.

↔ = Sampling Step Width.

Figure 3. A Cross Section of the Computer Man that Contains a Shotline

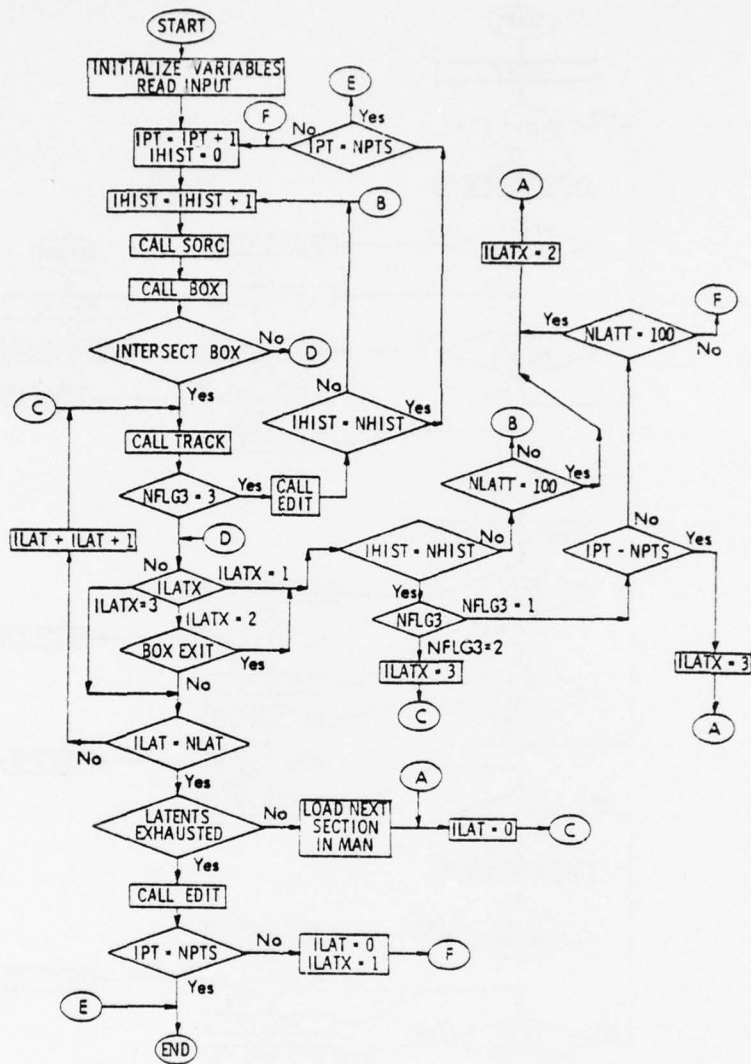


Figure 4. Main Routine

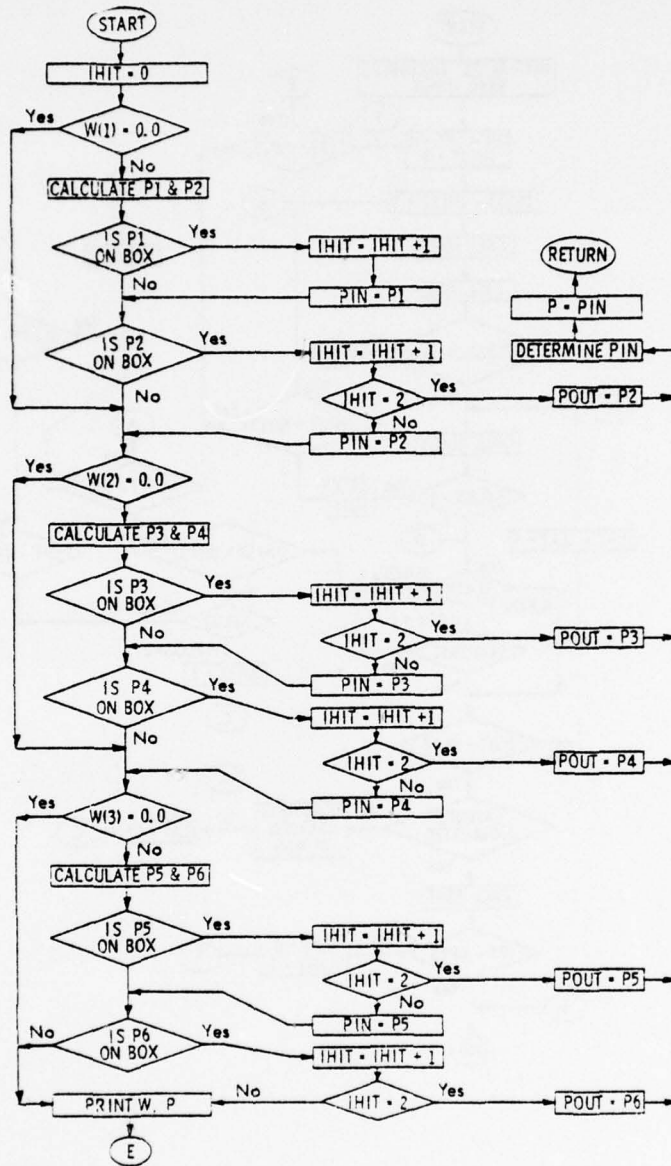


Figure 5. Box Subroutine

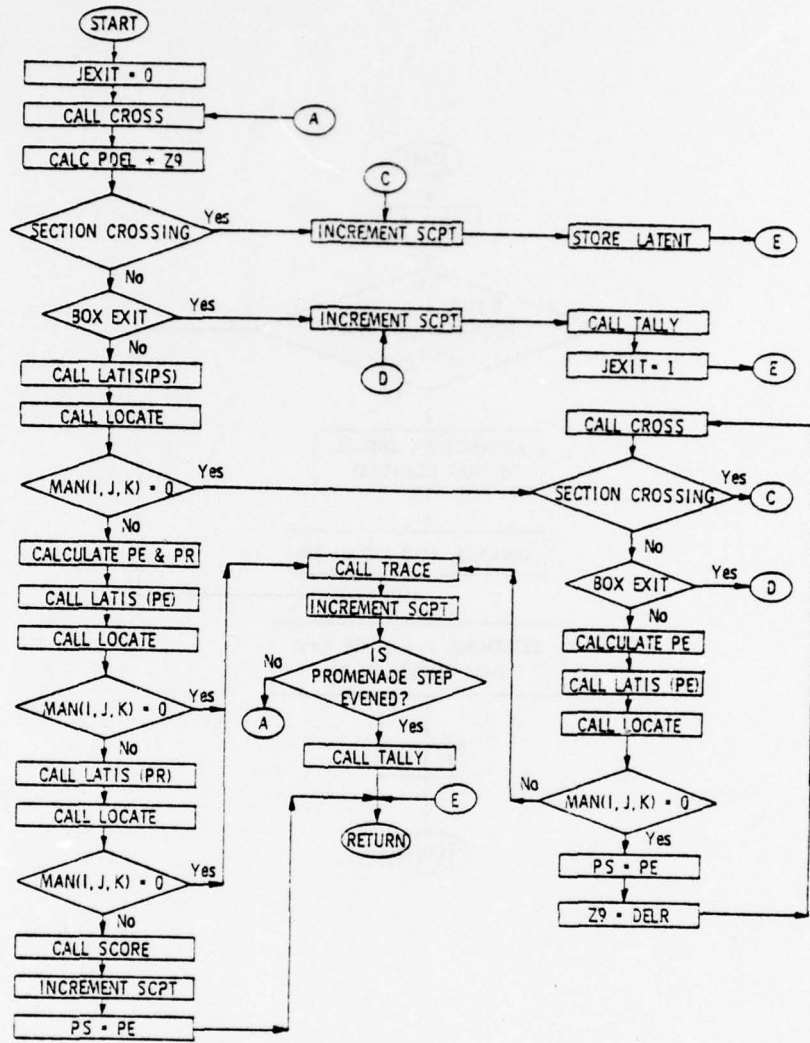


Figure 6. Even Subroutine

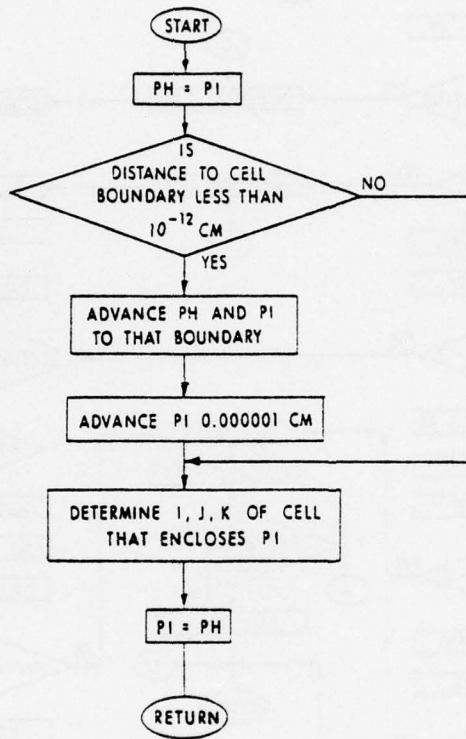


Figure 7. Latis Subroutine

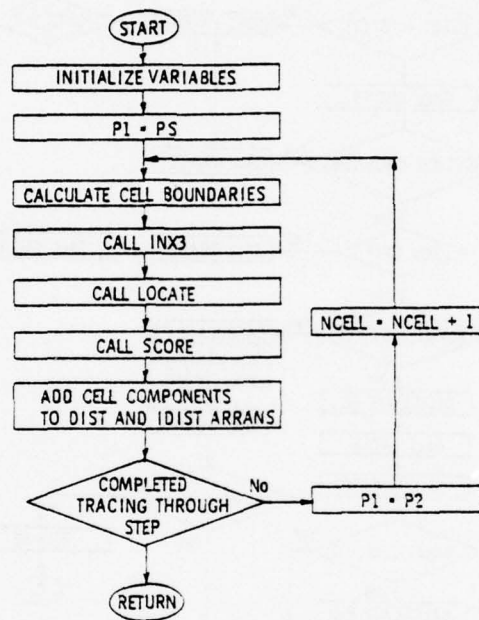


Figure 8. Trace Subroutine

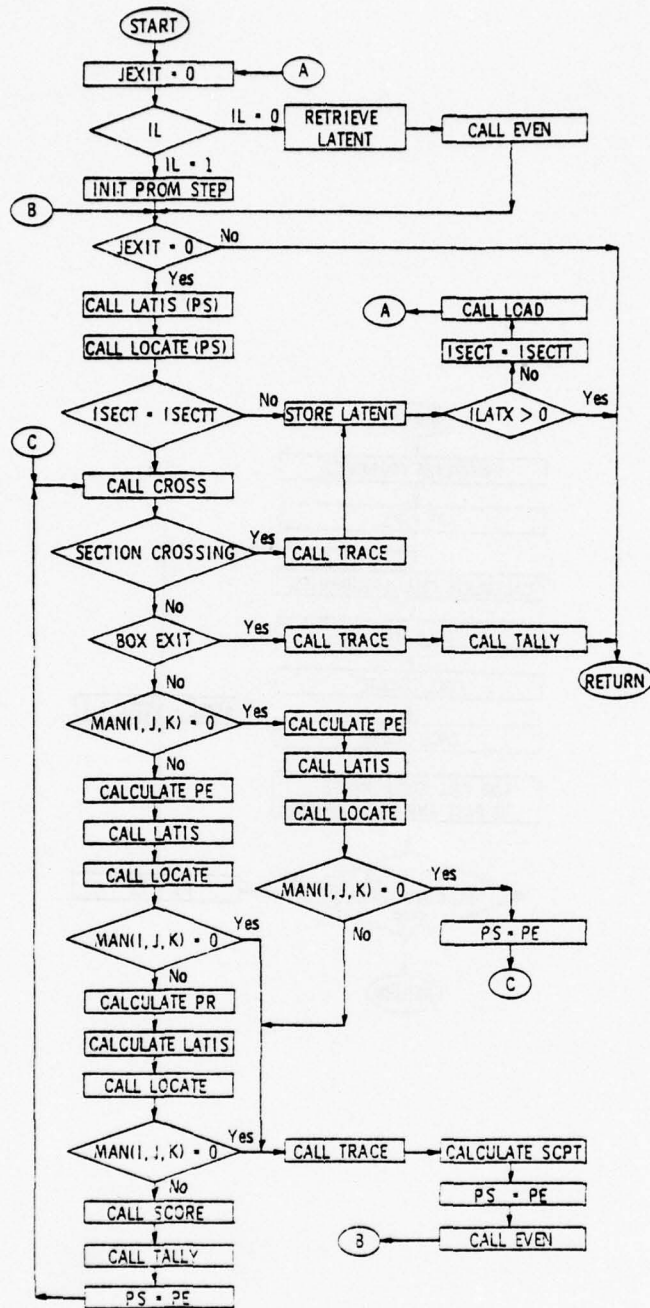


Figure 9. Track Subroutine

for the cell enclosing that point. This score is then added to the first bin of the array VI. The square of the score is added to the first bin of the V12 array. The array NVI, used to tally the number of promenades penetrating that amount of tissue, has its first bin incremented by one. The amount of tissue penetrated during the current promenade PBDY is set to the histogrammic bin width DELR.

The score is calculated exactly when the promenade initially enters tissue and the entering step is only partially in tissue. The final score deposited in the VI-array will also be made for a tissue penetration depth of DELR but more than one promenade step will be needed to attain that penetration. A partial score SCPT is obtained for the tissue-entering step by calculating the product of the fraction of a bin width intercepted by a cell and the M-value of that cell and summing those products over all cells encountered along the step.

This is accomplished in the code by calling subroutine TRACE to calculate the necessary cell intercepts and to retrieve the necessary M-values. The intercept lengths are stored in DIST(1,m) and the M-values in DIST(2,m). A partial score SCPT is then calculated that is given by

$$SCPT = \frac{1}{DEL R} \sum_{m=1}^{NCELL} DIST(1,m) \cdot DIST(2,m) \quad (1)$$

where

NCELL = the number of cells intercepted during the step

m = the running index over cells intercepted.

The variable PBDY will be incremented by

$$PBDY = \sum_{m=1}^{NCELL} DIST(1,m) \cdot H[DIST(2,m) - \epsilon] \quad (2)$$

where

$H[DIST(2,m) - \epsilon]$  = the Heaviside step function,

$\epsilon$  = a positive infinitesimal.

The variable PBOX will, as usual, be incremented by the full step width.

Subroutine EVEN is now called to calculate the partial score(s) of the additional step(s) needed to obtain a full bin width of tissue penetration. These latter score(s) may be estimated stochastically or calculated analytically as determined by the tissue or non-tissue status of

successive cells. The total score for a full bin-width of tissue penetration is added to the VI array and the other variables and arrays are properly incremented. The step width is then reset to DELR and a normal-gaited promenade through tissue is ready to be started.

That pace, conducted in subroutine TRACK, is maintained until the promenade exits the box or current section, or leaves tissue. A point is picked uniformly along each step and the M-value associated with the cell enclosing that point is retrieved from the array MAN and stored in the proper bin of the VI-array.

Subroutine TRACE is called again when checks predict that the next step will exit tissue. The variable values retrieved or calculated by TRACE are used to calculate a partial score SCPT, as described in Eq. 2 for the amount of tissue penetrated during the step. Subroutine EVEN is then entered and the promenade is continued until tissue is re-entered (or the box is exited or a new section is entered). The partial score for a segment of a length sufficient to complete the bin is calculated when tissue is reencountered and is added to the previously calculated partial score. The total score is added to the proper VI bin, all variables are updated, and control of the promenade is returned to subroutine TRACK where a normal gait is resumed.

The partial score will always be calculated for the last step of a promenade in the box (regardless of whether that step was encountered in subroutine TRACK or EVEN). The partial score is calculated in the customary manner, normalized to a full bin width, in a statistically valid way, and added to the proper VI bin.

The partial score will always be calculated for the last step before a section crossing. SCPT is calculated in the usual manner and placed in latent storage along with the other necessary parameter values. Subroutine LATENT performs this storage as well as the later retrieval.

The parameter values for 100 promenades can be stored as latents. The flag ILATX is used to direct the storage and retrieval of these parameters. ILATX is set to 1 at the beginning of each editing unit of histories. It remains at this value until latent storage is saturated or all promenades required by the current editing group have been started. This value of 1 directs the code to store latents when necessary but to not yet retrieve latents.

ILATX is set to 2 when the parameters for the 100 latent promenades have been stored and will remain at that value until all histories requested by the current editing group have been started. This mode orders the retrieval of the next latent promenade that is entering the currently loaded section of the man when the tracking of the previous promenade (either latent or virgin) ended in a section crossing, with a subsequent

latent storage. A box exit of the previous promenade will, on the other hand, return a space for the potential storage of a new latent that is derived from a presently unstarted promenade. New rays may therefore be generated and their promenades conducted until one of them has to be stored as a latent.

The sweep of the latent storage arrays for retrieving latent promenades entering the currently loaded section of the man is started when ILATX is set to 2. The VI-matrix of the section having the most latent promenades making an entrance, is loaded into MAN at the completion of a sweep and a new sweep will be initiated.

ILATX is set to 3 when all histories in an editing group have been started. New rays will not be generated and latent promenades will be retrieved and tracked until latent storage is exhausted. Restorage of a latent promenade after its retrieval and tracking is possible in this as well as the preceding mode of ILATX.

The editing of results may be implemented in the ends of each history, at the end of each group of histories emanating from a point, or at the end of a run. The mean of the score and its standard deviation are calculated for each bin. These values, as well as the number of promenades penetrating that amount of tissue, are printed in tabular form along with identifying captions.

### III. SUBROUTINES AND VARIABLES

#### A. Subroutine Descriptions

1. Driver Routine. This routine reads in the user-supplied data and instructions. Once calculated values of variables are derived, before entering the ray-generating cycle, the ILATX-routing to different modes of operation and choosing of VI-matrices for loading into the array MAN are also conducted in this routine.

2. Subroutine AD. This routine directs the logic flow to the user-chosen ray angular distribution.

3. Subroutine ARC. This subroutine picks an origin for the ray on an overhead, quarter-circle arc. These points may be picked from two distributions. The distributions are: (a) uniform, and (b) sine of the polar angle of the point.

4. Subroutine BIVAR. This subroutine picks the direction of a ray from a bivariate normal distribution about the line from the ray origin to a user-supplied point usually (but not necessarily) located in the man. The vertical and horizontal standard deviations of the marksman are supplied by the user.

5. Subroutine BOX. The man is assumed to be barely enclosed by a rectangular box. This subroutine will project the current ray toward the box and calculate the length SLR of the segment intercepted. The coordinates of the entering point of the ray are also returned to be used as the start of the promenade.

6. Subroutine CHECK. The Computer Man description is double-valued on the cell boundaries. Subroutine CHECK inspects rays to determine if they will travel on these boundaries. A ray that would have such a path will have the critical coordinate of its origin randomly translated by a small amount so that its path will not lie on a cell boundary.

7. Subroutine CRØSS. This subroutine is used to calculate the distance from the beginning of the current promenade step to the nearest z boundary. This distance SLRC is returned to the main program.

8. Subroutine EDIT. This subroutine directs the calculation of the mean and standard deviations of accumulated scores. It will also print the results in tabular form along with identifying captions.

9. Subroutine EVEN. This subroutine is called by subroutine TRACK after a promenade has entered (or reentered) tissue or has been revived after storage as a latent. The promenade gait will be brought into phase with the histogrammic bin structure of tissue penetration. Subroutines SCORE and TALLY are called in this subroutine.

10. Subroutine HEMI. This subroutine picks a ray-origin point uniformly on any of the user-chosen hemispheres.

11. Subroutine INX3. This subroutine calculates the distance from a point on a shotline in a cell to the exit of the shotline from the cell. This distance, as well as the exit point, are returned to subroutine TRACE for use in calculating partial scores.

12. Subroutine ISOT. This subroutine is used to pick the direction of a ray uniformly about its origin. In practice, the wasteful practice of picking directions for rays that would not intercept the man is improved by picking directions that will project a ray within the solid angle subtended by a sphere barely enclosing the man box. A normalization factor that is the ratio of this solid angle to  $4\pi$  is calculated so that the user may calculate absolute probabilities. The direction parameters are converted to direction cosines and the latter quantities are returned to the main program.

13. Subroutine LATENT. This subroutine is called to store or retrieve latent promenades. The 0 or 1 value of the first variable IL in the argument directs toward a retrieval or a storage respectively.

14. Subroutine LATIS. This subroutine locates the cell that encloses the point PI. The x-axis index (I1), y-axis index (J1), and z-axis index (K1) are returned. The I, J, and K indices are derived from I1, J1, and K1 and are used in their place for retrieving scores so that program abortion caused by truncation approximation will be avoided at the box boundaries. A point on a shotline that lies on a cell boundary will be assumed to lie within the cell being entered.

15. Subroutine LOAD. The principal function of this subroutine is to load the VI-matrix of a chosen section into the array MAN from disc. A secondary role is to call subroutine TEST to generate a test MAN array when requested by the user.

16. Subroutine LOCATE. The role of this subroutine is to determine the section of the Computer Man that includes the z-axis index KP (absolute). This index is then converted to its value relative to the bottom of that section and returned as KP (relative).

17. Subroutine SCORE. This subroutine retrieves the score from MAN(I,J,K). The score is returned in both fixed point and floating point mode. It is anticipated that this subroutine may be rewritten as required by other forthcoming descriptions of the Computer Man.

18. Subroutine SORC. This subroutine provides the routing to the chosen ray-generating routines. It will also calculate the distance from the ray origin to the center of the box or to the aiming point in the man.

19. Subroutine TALLY. This subroutine will add the score that was estimated or calculated to that already accumulated in the proper bin in the VI array. The bin of the array that tallies the number of shotlines reaching that penetration in tissue NVI is also incremented by 1. The square of the score is added to the VI2 array to be used at the completion of the editing group to calculate standard deviations.

20. Subroutine TEST. This subroutine is used to build a test pattern of scores in the MAN array. It can be easily rewritten to satisfy the needs of the user.

21. Subroutine TRACE. This subroutine is used to advance a distance R along a shotline while calculating the distances penetrated in intercepted cells and retrieving the scores in these cells. The number of cells in which the segment R lies is also calculated and returned as NCELL.

22. Subroutine TRACK. This subroutine initiates the promenade along a newly-generated shotline or resumes the promenade along a retrieved latent shotline. Subroutine EVEN is called when needed to bring the promenade in phase with the histogrammic bin structure. TRACK will conduct an even-gaited pace at other times (except for box-, section-,

or tissue-exits) with stochastic estimation of scores. TRACK will also calculate partial scores for the noted exceptions to the even gait when the preceding step was in TRACK.

23. Subroutine TRANS. The Computer Man box is assumed to be placed entirely in the first Cartesian octant with a corner located at the axis-origin. The RAYMAN subroutines that pick ray origins will use quarter-circles or spheres that are centered at the origin. This subroutine will translate the ray-origins, after being picked, so that they will lie on circles of spheres centered at the center of the man box.

24. Subroutine VAR. This subroutine calculates the standard deviation of each scoring bin.

#### B. Variable Description

1. BULSI(3). The coordinates of the aiming point in the Computer Man. It is used only when the bivariate normal angular distribution is used.

2. BKPT. The height at which the thickness of the slices changes.

3. DELR. This is the width of the histogrammic bins in which scores are accumulated. DELR is also the preferred step width of the promenade through tissue.

4A. DIST(2,IZ). The score retrieved in the IZth cell tracked by subroutine TRACE.

4B. DIST(1,IZ). The length of the segment intercepted by the IZth cell in the tracking by subroutine TRACE.

5A. FACT(I) I=1,10. An adjustment factor for quantizing a ranking in the VI-matrix. All elements in FACT are currently set to unity.

5B. FACT(11). This element of FACT is set to 1.0 and is used in calculating partial scores.

6. GCH. The height of the great circle above the plane = 0.

7. GCR. The radius of the great circle about the vertical axis of the Computer Man.

8A. GRID(I) I=1,2. The x- and y-cell width of the Computer Man.

8B. GRID(3). The z-cell width of the lower part of the Computer Man.

8C. GRID(4). The z-cell width of the upper part of the Computer Man.

9. I. The x-axis cell index of the Computer Man. This quantity is adjusted at the box boundaries to prevent computer truncation approximations from causing the logic errors that would lead to incorrect answers or run abortion.

10. I1. The same as I for internal cells but it is not adjusted at the box boundaries.

11. IBKPT. The z-axis of the top slice of cells possessing a thickness of GRID(3).

12. IDIST(IZ). The fixed point score in the IZth cell tracked in subroutine TRACE.

13. IHIST. The running index for the rays generated from a point.

14. IHIT. The number of interceptions made in the walls of the Computer Man box by the ray.

15. ILAT. The running index for the retrieval of latent promenades.

16. ILATX. The flag for directing latent promenade storage and retrieval. The value ILATX = 1 directs toward storage only. The value ILATX = 2 permits either storage or retrieval. The value ILATX = 3 permits retrieval only.

17. IPT. The running index for the ray origin point.

18. ISECT. The current section of the Computer Man stored in the array MAN.

19. ISECTT. The section of the Computer Man in which a designated point is located.

20. ISEED. A fixed point number used as an argument by the (0,1) random number generator.

21. ISL. The maximum number of horizontal slices of the Computer Man stored in a section. This number is the same for all sections except possibly the topmost section.

22. ISC. The fixed point value of the score in a cell.

23. J. The y-axis cell index of the Computer Man. This quantity is adjusted at the box boundaries to allow for computer truncation approximations.

24. J1. The same as J for internal cells but it is not adjusted at the box boundaries.

25A. JLAT(1,IP). The section number into which the IPth latent promenade is entering.

25B. JLAT(2,IP). The number of histogramic bins into which scores have been tallied for the IPth latent promenade.

26. JSEED. The fixed point argument for generating a pair of random numbers picked from a normal distribution.

27. JEXIT. A flag showing that a cell of subroutine EVEN has led to a box exit by the current promenade.

28. K. The z-axis cell index of the Computer Man. This quantity is adjusted at the box boundaries to allow for computer truncation approximations during calculations along the promenade.

29. KI. The same as K for internal cells but it is not adjusted at the box boundaries.

30. MAN(I,J,K). The array used to store the incapacitation input data for the current section of the Computer Man.

31. NARC. A flag for choosing the distribution from which an origin point of a ray on the quarter-circle is picked. The value NARC = 0 leads to picking by a distribution given by the sine of the polar angle while the value NARC = 1 leads to picking a point uniformly on the arc.

32. NCELL. The number of cells penetrated by the shotline segment when subroutine TRACE is used.

33. NDEL. The number of histogramic bins into which scores have been added for the current promenade.

34. NFACT. A flag for determining if FACT values are to be read in as part of the input or if the values are to be defaulted to 1.0.

35. NFLG1. A flag for determining the distribution from which ray origins will be picked. The options are discussed in the user-supplied input section of this report.

36. NFLG2. A flag for determining the distribution from which ray directions are chosen. The options are discussed in the user-supplied input section of this report.

37. NFLG3. The flag for determining the editing group to be used. The options are discussed in the user-supplied input section of this report.

38. NHIST. The number of rays to be started at a point.

39. NHITS. The number of rays intercepting the Computer Man for an editing group.
40. NLAT(ISECT). The number of latent promenades that are entering section ISECT.
41. NLATT. The number of latent promenades stored.
42. NLOAD. The number of times that a different VI-matrix was loaded into MAN.
43. NPTS. The number of points from which rays are to be started.
44. NSECT. The number of sections into which the Computer Man is divided.
45. NTEST. A flag to determine if test values are to be generated for the array MAN.
46. NVI(I) I=1,200. The counter for determining the number of shotlines that have penetrated a distance in tissue greater than (I-1) · DELK.
47. NX. The number of Computer Man cells in the x-direction.
48. NY. The number of Computer Man cells in the y-direction.
49. NZ. The number of Computer Man cells in the z-direction.
50. P(I) I=1,3. The coordinates of the ray origin.
51. PBDY. The distance in tissue penetrated during a promenade.
52. PBØX. The distance in the man box penetrated during a promenade.
53. PC(I) I=1,3. The coordinates of the point at which a promenade will cross the next section boundary.
- 54A. PLAT(1,IP). The initial value of PBDY for the IPth latent.
- 54B. PLAT(2,IP). The initial value of PBØX for the IPth latent.
- 54C. PLAT(3,IP). The value of SLR for the IPth latent.
55. PS(I) I=1,3. The coordinates of the beginning of the step in a promenade.
56. PR(I) I=1,3. The coordinates of the uniformly-picked point on a promenade step.

57. PE(I) I=1,3. The coordinates of the end of the step in a promenade.
58. PTAR(I,IP) I=1,3. The coordinates of the beginning of the IPth latent promenade or the coordinates of the origin of a user-provided ray.
59. PTAR(I,IP) I=4,6. The direction cosines of the shotline or ray described in item 58.
60. P1(I), P2(I), I=1,3. The coordinates of the intersection of the ray with the X=0 and X=XL3 plane respectively. This usage occurs in subroutine BOX.
61. P3(I), P4(I), I=1,3. The coordinates of the intersection of the ray with Y=0 and Y=XL2 plane respectively. This usage occurs in subroutine BOX.
62. P5(I), P6(I) I=1,3. The coordinates of the intersection of the ray with the Z=0 and Z=XL3 plane respectively. This usage occurs in subroutine BOX.
63. P1(I), P2(I). These arrays are also used in subroutines TRACE and INX3 to store the coordinates of the start or entrance of a segment into a cell and its termination or exit from the cell.
64. PDEL. The amount of tissue penetrated in a new bin. This variable is used in subroutine TRACK.
65. PL1. This variable is used in subroutine TRACE and INX3. The plane X=PL1 includes one side of the cell which could be the exit boundary of the shotline from the cell.
66. PL2. This variable is used in subroutine TRACK and TRACE. The plane Y=PL2 includes another side of the cell which could be the exit boundary of the shotline from the cell.
67. PL3. This variable is used in subroutine TRACK and TRACE. The plane Z=PL3 includes a third side of the cell which could be the exit boundary of the shotline from the cell.
68. R1. The radius of the sphere that barely encloses the Computer Man box.
69. R2. The distance from the ray origin to the center of the man box. It is also used as the distance from the ray origin to the point BULSI.
70. SCPT. A partial score that is calculated exactly when a promenade is being brought into phase with the histogrammic bin structure.

71. SD1. The standard deviation (degrees) in the vertical direction of the shots triggered by a marksman.
72. SD2. The standard deviation (degrees) in the horizontal direction of the shots triggered by a marksman.
73. SLR. The length of the segment of the ray intercepted by the Computer Man box.
74. SPCT. The ratio of the solid angle subtended by the Computer Man box sphere to  $4\pi$ .
75. SLRC. The distance from the beginning of a step to a section crossing. This variable is defaulted to 2,000,000 when  $W(3) = 0.0$ .
76. SPLT. An array for storing the heights of the section boundaries. The values are stored in ascending order.
77. SC. The floating-point score.
78. VI(IP). The bin for accumulating VI scores whose penetration in tissue is between  $(IP-1)* DELR$  and  $IP* DELR$ .
79. VI2(IP). The array for storing the square of the scores deposited in the VI array.
80. W(IZ), IZ=1,3. An array for storing the direction cosines of the current ray.
81. WP(IZ), IZ=1,3. An array for storing the direction cosines of the line from the ray origin to either the aiming point (BULSI) or the center of the Computer Man box.
82. XL1. The plane  $X=XL1$  is an outside boundary of the Computer Man box.
83. XL2. The plane  $Y=XL2$  is an outside boundary of the Computer Man box.
84. XL3. The plane  $Z=XL3$  is an outside boundary of the Computer Man box.
85. Z9. The length of the part of a segment of the shotline that is in the current cell. This variable name is used in subroutine TRACE and INX3.

#### IV. RAYMAN INPUT

##### 1. TITLE

A maximum of 80 alphanumeric identifying characters.

(8A10).

##### 2. ISEED , JSEED

ISEED = (0.1) random number routing argument.

JSEED = normal random number routine argument (2I10).

##### 3. NTEST, NFACT, NARC

NTEST = Flag for generating test numbers for the MAN array.

NTEST = 1, generates numbers for MAN.

NTEST = 0, loads data from disc to MAN.

NFACT = Flag for filling the FACT array.

NFACT = 0, reads in user-chosen numbers.

NFACT = 1, all elements in FACT are defaulted to unity.

NARC = Flag to determine distribution of ray origins on the quarter circle.

NARC = 0, points are picked from the sine of the polar angle distribution.

NARC = 1, points are picked uniformly along the arc.

(1015)

##### 4. NFLG1, NFLG2, NFLG3, NPTS, NHIST

NFLG1 = Ray origin flag.

NFLG1 = 1, read in points.

= 2, uniformly distributed on the upper hemisphere.

= 3, uniformly distributed on the lower hemisphere.

= 4, uniformly distributed on the front hemisphere.

= 5, uniformly distributed on the back hemisphere.

= 6, uniformly distributed on the right hemisphere.

= 7, uniformly distributed on the left hemisphere.

= 8, uniformly distributed on the great circle.

= 9, distributed on quarter-circle arc.

NFLG2 = Ray direction flag.

NFLG2 = 1, isotropic angular distribution.

= 2, bivariate normal angular distribution.

= 3, read in directions.

NFLG3 = Editing group flag.

NFLG3 = 1, composite editing group.  
= 2, each point editing group.  
= 3, each history editing group.

NPTS = Number of ray origin points.

NHIST = Number of rays per point.

(10I5)

5. NX, NY, NZ, IBKPT, ISECT, NSECT

NX = Number of x-axis cell boundaries in box.

NY = Number of y-axis cell boundaries in box.

NZ = Number of z-axis cell boundaries in box.

IBKPT = Number of z-axis cells up to slice thickness change.

ISECT = Initial section of the man loaded into MAN.

NSECT = Number of sections.

(10I5)

6. ISL

ISL = Number of slices per section (except for possibly the topmost section).

(10I5)

7. GRID(I), I=1,4 DELR

GRID(1) = x-axis cell width (cm).

GRID(2) = y-axis cell width (cm).

GRID(3) = z-axis lower cell width (cm).

GRID(4) = z-axis upper cell width (cm).

DELR = promenade step width (cm).

(6E10.3)

8A. IF NFLG1 = 1, provide

PO(I), I=1,3; WC(I), I=1,3 IF NPTS=1

PTAR(I,IPT) I=1,6 IF NPTS>1

PO(1) or PTAR(1,IPT) = x-coordinate of ray origin

PO(2) or PTAR(2,IPT) = y-coordinate of ray origin

PO(3) or PTAR(3,IPT) = z-coordinate of ray origin

W(1) or PTAR(4,IPT) = x-axis direction cosine of ray

W(2) or PTAR(5,IPT) = y-axis direction cosine of ray

W(3) or PTAR(6,IPT) = z-axis direction cosine of ray  
(6E10.3)

8B. IF NFLG1 = 2-7, provide

R2  
R2 = radius of hemisphere  
(6E10.3)

8C. IF NFLG1 = 8, provide

GRC, GCH  
GRC = Radius of circle (cm).  
GCH = Height of circle above z=0  
(6E10.3)

8D. IF NFLG1 = 9, provide

R2, THETA  
R2 = Radius of arc (cm)  
THETA = azimuth angle (degrees) with x-axis  
(6E10.3)

9. IF NFLG2 = 2, provide

SD1, SD2  
SD1 = azimuthal standard deviation (degrees) of the marksman.  
SD2 = elevation standard deviation (degrees) of the marksman.  
(6E10.3)

10. IF NFLG2 = 2, provide

BULSI(I) I=1,3  
BULSI(1) = x-coordinate of aiming point  
BULSI(2) = y-coordinate of aiming point  
BULSI(3) = z-coordinate of aiming point  
(6E10.3)

11. IF NFACT = 0, provide

FACT(I) I=1,10  
FACT(1) = adjustment for VI = 1  
.  
.  
.  
FACT(10) = adjustment factor for VI = 10  
FACT(11) = 1.0 always  
(6E10.3)

## V. TEST PROBLEM

The input needed for a test problem is given in Table I. This data is presented in the format in which it is printed in the output by the computer. The output average VI-Scores of this test problem is presented in Table II.

## VI. CONCLUSIONS

The RAYMAN Computer Code has been run successfully on the BRLESC computer. Readers desiring a copy of the code may send a blank magnetic tape to the Target Assessment Branch, Vulnerability/Lethality Division, ATTN: DRDAR-BVL, Ballistic Research Laboratory, Aberdeen Proving Ground, Maryland 21005. Users are cautioned that the BRLESC names of certain library routines may not agree with those used by other computer facilities and will need to be changed.

BEST AVAILABLE COPY

Table I. Sample Input

```

UPPER HEMISPHERE TEST
ISRD = 11111111 JSEED = 11113111
NTEST = 1 NFACT = 1
NARC = 1
NO PTS = 100 NO HIST = 1
ILATX = 1
NFLG1 = 2 NFLG2 = 1 NFLG3 = 1
NO XYS = 110 NO YIS = 95 NO ZIS = 84
IBKPT = 64 SECT NO = 2 NO SECT = 14 NO SLICE = 6
X-CELL = 0.5000E 00 Y-CELL = 0.5000E 00 Z1-CELL = 0.2600E 01 Z2-CELL = 0.1200E 01
R-INCRM = 0.1000E 01
SOURCE-X = 0.0000E 00 SOURCE-Y = 0.0000E 00 SOURCE-Z = 0.0000E 00
X-DIR CS = 0.0000E 00 Y-DIR CS = 0.0000E 00 Z-DIR CS = 0.0000E 00
BOX-RAD = 0.1000E 03 PT-RAD = 0.1000E 04
X-SIDE = 0.5500E 02 Y-SIDE = 0.2750E 02 Z-SIDE = 0.1704E 03
FACT MAT = 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
BELT RAD = 0.0000E 00 BELT HGT = 0.0000E 00
BULSI(1) = 0.0000E 00 BULSI(2) = 0.0000E 00 BULSI(3) = 0.0000E 00
GREAT CIRCLE SH-ANGLE = .000000E 00
GAPAT CIRCLE CS-ANGLE = .000000E 00
HDR SD = 0.0000E 00 VERT SD = 0.0000E 00
H-GW FCT = 0.0000E 00 V-GW FCT = 0.0000E 00

```

# BEST AVAILABLE COPY

Table II. Sample Output

INCREMENT	VUL INDEX	STAND DEV	NO RAYS
1* * *	0.7000E 01+-	0.3980E 01* * *	20
2* * *	0.7150E 01+-	0.3392E 01* * *	20
3* * *	0.7150E 01+-	0.2758E 01* * *	20
4* * *	0.7300E 01+-	0.2510E 01* * *	20
5* * *	0.7201E 01+-	0.2550E 01* * *	20
6* * *	0.6765E 01+-	0.2930E 01* * *	17
7* * *	0.6588E 01+-	0.3483E 01* * *	17
8* * *	0.6647E 01+-	0.3605E 01* * *	17
9* * *	0.7200E 01+-	0.2931E 01* * *	15
10* * *	0.6857E 01+-	0.3207E 01* * *	14
11* * *	0.8077E 01+-	0.2290E 01* * *	13
12* * *	0.7231E 01+-	0.2242E 01* * *	13
13* * *	0.8154E 01+-	0.2444E 01* * *	13
14* * *	0.8417E 01+-	0.2392E 01* * *	12
15* * *	0.7818E 01+-	0.2892E 01* * *	11
16* * *	0.7571E 01+-	0.3452E 01* * *	11
17* * *	0.7823E 01+-	0.3629E 01* * *	10
18* * *	0.8111E 01+-	0.3257E 01* * *	9
19* * *	0.7703E 01+-	0.3230E 01* * *	9
20* * *	0.9000E 01+-	0.1927E 01* * *	8
21* * *	0.9250E 01+-	0.1753E 01* * *	8
22* * *	0.8750E 01+-	0.2310E 01* * *	8
23* * *	0.8522E 01+-	0.3149E 01* * *	8
24* * *	0.9571E 01+-	0.1134E 01* * *	7
25* * *	0.9286E 01+-	0.1970E 01* * *	7
26* * *	0.7714E 01+-	0.4071E 01* * *	7
27* * *	0.7857E 01+-	0.3338E 01* * *	7
28* * *	0.7714E 01+-	0.3592E 01* * *	7
29* * *	0.7571E 01+-	0.3359E 01* * *	7
30* * *	0.7800E 01+-	0.3538E 01* * *	6
31* * *	0.8000E 01+-	0.2739E 01* * *	5
32* * *	0.8000E 01+-	0.3082E 01* * *	5
33* * *	0.7000E 01+-	0.3606E 01* * *	3
34* * *	0.6709E 01+-	0.4851E 01* * *	2
35* * *	0.1000E 02+-	0.0000E 00* * *	2
36* * *	0.1000E 02+-	0.1738E 06* * *	2
37* * *	0.1000E 02+-	0.9990E 01* * *	1
38* * *	0.1000E 02+-	0.9990E 01* * *	1
39* * *	0.1000E 01+-	0.9990E 01* * *	1
40* * *	0.2000E 01+-	0.9990E 01* * *	1
41* * *	0.2000E 01+-	0.9990E 01* * *	1
42* * *	0.3000E 01+-	0.9990E 01* * *	1
43* * *	0.4000E 01+-	0.9990E 01* * *	1
44* * *	0.5000E 01+-	0.9990E 01* * *	1
45* * *	0.6000E 01+-	0.9990E 01* * *	1

ACKNOWLEDGEMENT

The author would like to thank Mr. Charles Stanley for his aid in running and debugging the program. He also appreciates the graphic work provided by Mr. Tom Long.

APPENDIX  
RAYMAN LISTING

# BEST AVAILABLE COPY

```
C PROGRAM RAYMAN
  DIMENSION TITLE(8)

  COMMON/MANN/MAN(110,55,6),NTEST
  COMMON/LATEN/JLAT(2,100),NLAT(20),ILAT,NLATT,ISECT,ILATX,ISECTT,
  * PLAT(4,100),PTAR(6,100),PHDY,PROX,SCPT,NDEL,IMIST,IPT,NARC,NPTS
  COMMON/LATIX/GRID(4),BKPT,IBKPT,SPLT(10)
  COMMON/POINT/PS(3),PR(3),PE(3),PC(3),DELR,NFLG3
  COMMON/MISC/R1,R2,GCR,GCH,NFLG1,NFLG2,SD1,SD2,GW1,GW2
  COMMON/TRAC/SLR,XL1,XL2,XL3,NHITS,DIST(2,10),IDIST(10),NCELL
  COMMON/EVAL/VI(200),NVI(200),VI2(200),FACT(11)
  COMMON/SORS/WP(3),W(3),P(3),PO(3),BULSI(3),SPCT,ISEED,JSEED,
  * SNT,CST
  COMMON/LAT/NX,NY,NZ,NSECT,ISL
  COMMON/LJDE/NLOAD
500 FORMAT(F10.1)
501 FORMAT(BA10)
502 FORMAT(10I5)
503 FORMAT(6E10.3)
504 FORMAT(1M )
505 FORMAT(1M1)
506 FORMAT(2I10)
507 FORMAT(6(A10,I5,5X))
508 FORMAT(4(A10,E11,4,5X))
509 FORMAT(2A10,A3,E11,7)
510 FORMAT(A10,A4,I5)
511 FORMAT(A10,I1(F9.4,1X))
512 FORMAT(2(A10,I10,2X))
513 FORMAT(19F10.5)
514 FORMAT(19X,I2,4X,F10.5)
515 FORMAT(6(E20.14),/,I20)
516 FORMAT(3(E20.14))
  READ(5,501) TITLE
  READ(5,506) ISEED,JSEED
  READ(5,502) NTEST,NFACT,NARC
  READ(5,502) NFLG1,NFLG2,NFLG3,NPTS,NHIST
  IF(NFLG2.EQ.3) NHIST = 1
  READ(5,502) NX,NY,NZ,IBKPT,ISECT,NSECT
  READ(5,502) ISL
  READ(5,503) GRID,DELR
  IF(NFLG1.EQ.1) GO TO 6
  IF(NPTS.EQ.1) GO TO 4
  READ(5,503) PO,W
  GOTO 9
6 IF(NFLG1.LT.9) GO TO 5
  READ(5,503) R2,THETA
  THETA = 3.14159*THETA/180.0
  CST = COSF(THETA)
  SNT = SINF(THETA)
  GO TO 9
```

# BEST AVAILABLE COPY

```
4 READ(5,503) ((PTAR(I,J),I=1,6),J=1,NPTS)
   GOTO 9
5 IF(NFL31.GT.7) GO TO 8
   READ(5,503) R2
   GO TO 9
8 READ(5,503) GCR,QCH
   WRITE(6,504)
9 IF(NFL32.NE.2) GO TO 10
   READ(5,503) SD1,SD2,GM1,GM2
   READ(5,503) RULSI
10 IF(NFACT.EQ.0) GOTO 998
   DO 997 I=1,10
     FACT(I) = 1.0
997 CONTINUE $ GOTO 996
998 CONTINUE
   READ(5,503) FACT
996 CONTINUE
   FACT(11) = 1.0
   BKPT = GRID(3)*FLOATF(1BKPT)
   XL1=GRID(1)*FLOATF(NX)
   XL2=GRID(2)*FLOATF(NY)
   XL3=BKPT+GRID(4)*FLOATF(NZ-1BKPT)
   R1 = (0.5*XL1)**2 + (0.5*XL2)**2 + (0.5*XL3)**2
   R1 = SQRTF(R1)
   J=0
   DO 777 I=1,NSECT
     J=J+13L
     IF(J.GT.1BKPT) GO TO 776
     XJ=FLOATF(J)
     SPLT(I)=XJ*GRID(3)
     GO TO 777
776 XJ=FLOATF(J-1BKPT)
     SPLT(I)=9KPT+(XJ)*GRID(4)
777 CONTINUE
   ILATX = 1
   WRITE(6,505)
   DO 778 I=1,NSECT
     WRITE(6,514)(I,SPLT(I))
778 CONTINUE
   WRITE(6,501) TITLE
   WRITE(6,504)
   CAP1 = 10HISEED = SCAP2 = 10HJSEED =
   WRITE(6,512) CAP1,ISEEU,CAP2,JSEED
   WRITE(6,504)
   CAP1 = 10HNTST = SCAP2 = 10HNFACT =
   WRITE(6,507)CAP1,NTEST,CAP2,NFACT
   WRITE(6,504)
   CAP1 = 10HVARC=
   WRITE(6,507)CAP1,NARC
   WRITE(6,504)
```

BEST AVAILABLE COPY

```

CAP1 = 10H NO PTS = SCAP2 = 10HNO HIST =
WRITE(6,507)CAP1,NPTS,CAP2,NHIST
WRITE(6,504)
CAP1 = 10HILATX =
WRITE(6,507)CAP1,ILATX
WRITE(6,504)
CAP1 = 10HNYFLG1 = SCAP2 = 10HNFLG2 = SCAP3 = 10HNFLG3 =
WRITE(6,507)CAP1,NFLG1,CAP2,NFLG2,CAP3,NFLG3
WRITE(6,504)
CAP1 = 10HNO X'S = SCAP2 = 10HNO Y'S = SCAP3 = 10HNO Z'S =
WRITE(6,507)CAP1,NX,CAP2,NY,CAP3,NZ
WRITE(6,504)
CAP1 = 10HIBKPT = SCAP2 = 10HSECT NO = SCAP3 = 10HNO SECT =
CAP4 = 10HNO SLICE =
WRITE(6,507)CAP1,IBKPT,CAP2,ISECT,CAP3,NSECT,CAP4,ISL
WRITE(6,504)
CAP1 = 10HX-CELL = SCAP2 = 10HY-CELL = SCAP3 = 10HZ1-CELL =
CAP4 = 10HZ2-CELL =
WRITE(6,508)CAP1,GRID(1),CAP2,GRID(2),CAP3,GRID(3),CAP4,GRID(4)
WRITE(6,504)
CAP1 = 10HR-INCREM =
WRITE(6,508)CAP1,DELR
WRITE(6,504)
CAP1 = 10HSOURCE-X = SCAP2 = 10MSOURCE-Y = SCAP3 = 10MSOURCE-Z =
WRITE(6,508)CAP1,PO(1),CAP2,PO(2),CAP3,PO(3)
WRITE(6,504)
CAP1 = 10HX-DIR CS = SCAP2 = 10HY-DIR CS = SCAP3 = 10HZ-UIR CS =
WRITE(6,508)CAP1,W(1),CAP2,W(2),CAP3,W(3)
WRITE(6,504)
CAP1 = 10HBOX-RAD = SCAP2 = 10HPT-RAD =
WRITE(6,508)CAP1,R1,CAP2,R2
WRITE(6,504)
CAP1 = 10HX-SIDE = SCAP2 = 10HY-SIDE = SCAP3 = 10HZ-SIDE =
WRITE(6,508)CAP1,XL1,CAP2,XL2,CAP3,XL3
WRITE(6,504)
WRITE(6,504)
CAP1 = 10HFACT MAT =
WRITE(6,511)CAP1,FACT
WRITE(6,504)
CAP1 = 10HBELT RAD = SCAP2 = 10HBELT HGT =
WRITE(6,508)CAP1,GCR,CAP2,GCH
WRITE(6,504)
CAP1 = 10HBULSI(1) = SCAP2 = 10HBULSI(2) = SCAP3 = 10HBULSI(3) =
WRITE(6,508)CAP1,BULSI(1),CAP2,BULSI(2),CAP3,BULSI(3)
WRITE(6,504)
CAP1=10HGREAT CIRC$ CAP2=10HLE SN-ANGL$ CAP3=3HE$
$ CAP4=10HLE CS-ANGL
WRITE(6,509)CAP1,CAP2,CAP3,SNT
WRITE(6,504)
WRITE(6,509) CAP1,CAP4,CAP3,CST
```

# BEST AVAILABLE COPY

```
WRITE(6,504)
CAP1 = 10HHR SD = SCAP2 = 10HVRT SD =
WRITE(6,508)CAP1,SD1,CAP2,SD2
WRITE(6,504)
CAP1 = 10HH-GW FCT =SCAP2 = 10HV-GW FCT =
WRITE(6,508) CAP1,GW1,CAP2,GW2
WRITE(6,504)
WRITE(6,504)
WRITE(6,504)
WRITE(6,504)
WRITE(6,504)
WRITE(6,504)
7 CALL LOAD(ISECT) $ NLOAD=1
1 NLATT = 0
11 IPT = 0
12 IHIST = 0
13 IPT = IPT + 1
14 IHIST = IHIST + 1
3 CALL SORC
20 CALL BOX(IHIT)
IF(IHIT.LT.2) GO TO 995
CALL CHECK
IF(NFLG3.LT.3) GO TO 21
CALL LAVIS(P,I,J,K)
CALL LOCATE(K,ISECT)
CALL LATENT(1,IPT,P)
NLATT=NLATT+1
NLAT(ISECT)=NLAT(ISECT)+1
994 ISECT = JLAT(1,IPT)
IF(ISECT.EQ.0) GO TO 995
CALL LOAD(ISECT) $ NLOAD = NLOAD + 1
CALL TRACK(0)
GO TO 994
21 CALL TRACK(1)
995 IF(NFLG3.GT.1) GO TO 103
IF(IHIST.LT.NHIST) GO TO 108
IF(IPT.EQ.NPTS) GO TO 102
IHIST = 0 $ IPT = IPT + 1
GO TO 108
102 ILATX = 3 $ GO TO 113
103 IF(NFLG3.GT.2) GO TO 116
IF(IHIST.LT.NHIST) GO TO 108
GO TO 102
108 IF(NLATT.LT.100) GO TO 14
IF(ILATX.EQ.2) GO TO 109
ILATX = 2
113 ILAT = 0
114 ILAT = ILAT + 1
IF(JLAT(1,ILAT).NE.ISECT) GO TO 120
CALL TRACK(0)
```

# BEST AVAILABLE COPY

```

IF(ILATX.EQ.3) GO TO 120
109 IF(JLAT(1,ILAT).EQ.0) GO TO 14
120 IF(ILAT.LT.NLATT) GO TO 114
115 NLAT(ISECT) = 0
I=15 ISECT=1
2 I=I+1
IF(NLAT(I).GT.NLAT(ISECT)) ISECT=I
IF(I.LT.NSECT) GOTO 2
IF(NLAT(ISECT).EQ.0) GO TO 116
CALL LOAD(ISECT) $ NLOAD = NLOAD + 1
GO TO 113
116 CAP1 = 10HNO MAN HITS CAP2 = 4HS =
WRITE(6,510)CAP1,CAP2,NVI(1)
WRITE(6,504)
CAP1 = 10HNDRMALIZATS CAP2 = 10HION FACTORS CAP3 = 3H =
WRITE(6,509)CAP1,CAP2,CAP3,SPCT
WRITE(6,504)
CAP1 = 10HISEED = $CAP2 = 10MJSEED =
WRITE(6,512) CAP1,ISEED,CAP2,JSEED
WRITE(6,504)
CAP1=10HNO DISC LOS CAP2=4HAD =
WRITE(6,510)CAP1,CAP2,NLOAD
CALL EDIT
ILATX = 1
NLATT = 0 $ NHITS = 0
220 IF(NFLG3.EQ.1) GO TO 117
IF(NFLG3.GT.2) GO TO 999
IF(IPT.EQ.4PTS) GO TO 117
GO TO 12
999 IF(IHIST.LT.NHIST) GO TO 14
IF(IPT.LT.4PTS) GO TO 12
117 STOP
98765 CONTINUE
98766 CONTINUE
WRITE(6,517) SCH1,SCATT,NLOAD,NTEST,ILAT,NLATT,ISECT,ILATX,ISECTT
517 FORMAT (2(E12.4,2X),7(14,2X))
WRITE(6,518) PHDY,PHOX,SCPT,NDEL,IHIST,IPT,NARC,NPTS
518 FORMAT(3(E12.4,2X),5(15,2X))
WRITE(6,519) GRID,BKPT,IHKPT,SPLT
519 FORMAT(5(E10.2,2X),14,(E20.12,2X),/,6(E20.12,1X),/,/,
*6(E20.12,1X))
WRITE(6,520) PS,PR,PE,PG,DELR,NFLG3
520 FORMAT(9(E12.4,1X),/,4(E12.4,1X),14)
WRITE(6,521) R1,R2,GCR,GCH,NFLG1,NFLG2,SD1,SD2,GW1,GW2
521 FORMAT(4(E12.4,1X),2(14,2X),4(E12.4,1X))
WRITE(6,522) SLR,XL1,XL2,XL3,NHITS,DIST,IDIST,NCELL
522 FORMAT (4(E12.4,1X),15,/,10(E12.4,1X),/,10(E12.4,1X),/,11(15,1X))
WRITE(6,523) FACT
523 FORMAT(11(E12.4))
WRITE(6,524) WP,W,P,PO,BULSI,SPCT,ISEED,JSEED

```

# BEST AVAILABLE COPY

```
524 FORMAT(6(E12.4,1X),/,6(E12.4,1X),/,4(E12.4,1X),2(I9,1X))
      WRITE(6,504)
      WRITE(6,525) NLAT
525 FORMAT(10(2X,15))
      WRITE(6,504)
      WRITE(6,525) JLAT
      WRITE(6,504)
      STOP
      END

C
C
      SUBROUTINE AD
      COMMON/MISC/R1,R2,GCR,GCH,NFLG1,NFLG2,SD1,SD2,GW1,GW2
      IF(NFLG2.GT.1) GOTO 2
      CALL ISOT
      GO TO 3
2 CONTINUE
      IF(NFLG2.GT.2) GOTO 3
      CALL BIVAR
3 RETURN
      END

C
C
      SUBROUTINE ARC(NARC)
      NARC = 1 IS UNIFORM, NARC = 0 IS ISOTROPIC IE COSINE WEIGHTED
      COMMON/SORS/WP(3),W(3),P(3),PO(3),BULSI(3),SPCT,ISEED,JSEED,
      * SNT,CST
      COMMON/MISC/R1,R2,GCR,GCH,NFLG1,NFLG2,SD1,SD2,GW1,GW2
      IF(NARC.EQ.0) GO TO 2
1 X1=URAN31(ISEED)
      X2=URAN31(ISEED)
      X1S = X1**2  X2S = X2**2
      T=X1S+X2S
      IF(T.GT.1.0) GOTO 1
      SNPHI=2.0*X1*X2/T
      CSPHI=ABSF((X1S-X2S)/T)
      GOTO 3
2 CONTINUE
      CSPHI = URAN31(ISEED)
      SNPHI = SQRTF(1.0 - CSPHI**2)
3 CONTINUE
      P(3)=R2*CSPHI
      P(1)=R2*SNPHI*CST
      P(2)=R2*SNPHI*SNT
      RETURN
      END

C
C
      SUBROUTINE BIVAR
      COMMON/SORS/WP(3),W(3),P(3),PO(3),BULSI(3),SPCT,ISEED,JSEED,
```

```

* SNT,CST
COMMON/MISC/R1,R2,GCR,GCH,NFLG1,NFLG2,SD1,SD2,OW1,OW2
IF(WP(3).EQ.1.0) GO TO 5
IF(WP(3).EQ.1.0) GOTO 0
IF(IPRIM.EQ.1) GOTO 1
PSI=WP(3)
PSI=ARCCOS(PSI)
THET=WP(2)/SQRTF(1.0-WP(3)**2)
THET=ARCCOS(THET)
IPRIM=1
SPCT=1.0
SD1 = SD1*3.14159/180.0
SD2 = SD2*3.14159/180.0
1 CALL NRVN31(RNN1,RNN2,JSEED)
DEL=SD2*RNN2
RNN3=URAN31(ISEED)
IF(RNN3.GT.0.5)DEL=-DEL
PSIH=PSI+DEL
W(3)=COSF(PSIH)
DEL=SD1*RNN1
RNN3=URAN31(ISEED)
IF(RNN3.GT.0.5)DEL=-DEL
THETH=THET+DEL
SINT=SQRTF(1.0-WP(3)**2)
W(2)=SINT*COSF(THETH)
W(1)=SINT*SINF(THETH)
WRITE(6,10) WP,W,P,RNN1,RNN2,THETH
10 FORMAT(6(E10.3,2X))
5 RETURN
END

```

C  
C

```

SUBROUTINE BOX(IHIT)
DIMENSION PIN(3),POUT(3),NPL(2)
DIMENSION P1(3),P2(3),P3(3),P4(3),P5(3),P6(3)
COMMON/SQRS/WP(3),W(3),P(3),PO(3),BULSI(3),SPCT,ISEED,JSEED,
* SNT,CST
COMMON/TRAC/SLR,XL1,XL2,XL3,NHITS,DIST(2,10),IDIST(10),NCELL
IHIT=0
COM P1 AND P2 ARE X=0 AND X=XL1 INTERCEPTS
P1(1)=0.0 $ P2(1)=XL1
IF(W(1).EQ.0.0) GOTO 1
Z9=-P(1)/W(1)
P1(2)=W(2)*Z9+P(2)
P1(3)=W(3)*Z9+P(3)
Z9=Z9+XL1/W(1)
P2(2)=W(2)*Z9+P(2)
P2(3)=W(3)*Z9+P(3)
COM X=0 AND X=XL1 PLANES
COM CHECK FOR BOX INTERCEPT

```

```

IF(P1(2).GE.XL2.OR.P1(2).LE.0.0)GOTO 5
IF(P1(3).GE.XL3.OR.P1(3).LE.0.0)GOTO 5
IHIT=1 $ NPL(1)=1
DO 100 I=1,3
PIN(I)=P1(I)
100 CONTINUE
5 IF(P2(2).GE.XL2.OR.P2(2).LE.0.0)GOTO 201
IF(P2(3).GE.XL3.OR.P2(3).LE.0.0)GOTO 201
IHIT=IHIT+1
DO 7 I=1,3
IF(IHIT.EQ.2)GOTO 6
PIN(I)=P2(I) $ NPL(1)=2 $ GOTO 7
6 POUT(I)=P2(I) $ NPL(2)=2
7 CONTINUE
201 IF(IHIT.EQ.2)GOTO 20
COM P3 AND P4 ARE Y=0 AND Y= XL2 INTERCEPTS
1 P3(2)=0.0 $ P4(2)=XL2
IF(W(2).EQ.0.0) GOTO 3
Z9=-P(2)/W(2)
P3(1)=W(1)*Z9+P(1)
P3(3)=W(3)*Z9+P(3)
Z9=Z9+XL2/W(2)
P4(1)=W(1)*Z9+P(1)
P4(3)=W(3)*Z9+P(3)
COM Y=0 AND Y=XL2 PLANES
IF(P3(1).GE.XL1.OR.P3(1).LE.0.0) GOTO 10
IF(P3(3).GE.XL3.OR.P3(3).LE.0.0) GOTO 10
IHIT=IHIT+1
DO 101 I=1,3
IF(IHIT.EQ.2)GOTO 9
PIN(I)=P3(I) $ NPL(1)=3 $ GOTO 101
9 POUT(I)=P3(I) $ NPL(2)=3
101 CONTINUE
10 IF(P4(1).GE.XL1.OR.P4(1).LE.0.0) GOTO 202
IF(P4(3).GE.XL3.OR.P4(3).LE.0.0) GOTO 202
IHIT=IHIT+1
DO 12 I=1,3
IF(IHIT.EQ.2)GOTO 11
PIN(I)=P4(I) $ NPL(1)=4 $ GOTO 12
11 POUT(I)=P4(I) $ NPL(2)=4
12 CONTINUE
202 IF(IHIT.EQ.2)GOTO 20
COM P5 AND P6 ARE Z=0 AND Z=XL3 INTERCEPTS
3 P5(3)=0.0 $ P6(3)=XL3
IF(W(3).EQ.0.0) GO TO 203
Z9=-P(3)/W(3)
P5(1)=W(1)*Z9+P(1)
P5(2)=W(2)*Z9+P(2)
Z9=Z9+XL3/W(3)
P6(1)=W(1)*Z9+P(1)

```

```

P6(2)=W(2)+Z9+P(2)
COM Z=0 ANU Z=XL3 PLANES
IF(P5(1).GE.XL1.OR.P5(1).LE.0.0) GOTO 16
IF(P5(2).GE.XL2.OR.P5(2).LE.0.0) GOTO 16
IHIT=IHIT+1
DO 15 I=1,3
IF(IHIT.EQ.2) GOTO 14
PIN(I)=P5(I) $ NPL(1)=5 $ GOTO 15
14 POUT(I)=P5(I) $ NPL(2)=5
15 CONTINUE
IF(IHIT.EQ.2) GOTO 20
16 IF(P6(1).GE.XL1.OR.P6(1).LE.0.0) GOTO 203
IF(P6(2).GE.XL2.OR.P6(2).LE.0.0) GOTO 203
IHIT=IHIT+1
DO 18 I=1,3
POUT(I)=P6(I) $ NPL(2)=6
18 CONTINUE
203 IF(IHIT.EQ.2) GO TO 20
IF(IHIT.EQ.0) GO TO 52
COM QUESTIONABLE RAY
19 PRINT 50,W,P
50 FORMAT(6E14.7)
GO TO 52
20 Z8=0.0 $ Z9=0.0
DO 21 I=1,3
Z8=Z8+(PIN(I)-P(I))**2
Z9=Z9+(POUT(I)-P(I))**2
21 CONTINUE
IF(Z8.LE.Z9) GOTO 23
DO 22 I=1,3
Z7=PIN(I)
PIN(I)=POUT(I)
POUT(I)=Z7
22 CONTINUE
Z7=NPL(1) $ Z6=Z8
NPL(1)=NPL(2) $ Z8=Z9
NPL(2)=Z7 $ Z9=Z6
23 SLR = SQRTF(Z9) - SQRTF(Z8)
25 DO 51 I=1,3
P(I) = PIN(I)
51 CONTINUE
52 RETURN
END

C
C
SUBROUTINE CHECK
COMMON/SORS/MP(3),W(3),P(3),PO(3),BULSI(3),SPCT,ISEED,JSEED,
* SNT,CST
COMMON/LATIX/GRID(4),BKPT,IBKPT,SPLT(19)
DO 1 I = 1,2

```

# BEST AVAILABLE COPY

```
IF(W(1).NE.0.0) GO TO 1
X = P(1)/GRID(1)
IX = XFIXF(X)
IF(X.NE.FLOATF(IX)) GO TO 1
X = URAN31(ISEED)
P(1) = P(1) + 0.00001
IF(X.GT.0.5) P(1) = P(1) - 0.00002
1 CONTINUE
IF(W(3).NE.0.0) GO TO 2
IF(P(3).GT.BKPT) GO TO 3
X = P(3)/GRID(3)
IX = XFIXF(X)
IF(X.NE.FLOATF(IX)) GO TO 2
GO TO 4
3 X = (P(3) - BKPT)/GRID(4)
IX = XFIXF(X)
IF(X.NE.FLOATF(IX)) GO TO 2
4 X = URAN31(ISEED)
P(3) = P(3) + 0.00001
IF(X.GT.0.5) P(3) = P(3) - 0.00002
2 CONTINUE
RETURN
END
```

C  
C

```
SUBROUTINE CROSS(SLRC)
COMMON/LATEN/JLAT(2,100),NLAT(20),ILAT,NLATT,ISECT,ILATX,ISECTT,
* PLAT(4,100),PTAR(6,100),PBDY,PBOX,SCPT,NDFL,IMIST,IPT,NARC,NPTS
COMMON/POINT/PS(3),PR(3),PE(3),PC(3),DELR,NFLG3
COMMON/SORS/WP(3),W(3),P(3),PO(3),BULSI(3),SPCT,ISEED,JSEED,
* SNT,CST
COMMON/LATIX/GRID(4),BKPT,IBKPT,SPLT(19)
COMMON/LAT/NX,NY,NZ,NSECT,ISL
CJM DETERMINES IF SECTION CROSSING
ISECTT=ISECT+1
BLRC = 2000000.0
IF(W(3).GE.0.0.AND.ISECT.EQ.NSECT)GOTO 3
IF(W(3).LE.0.0.AND.ISECT.EQ.1) GO TO 3
IF(W(3).EQ.0.0) GO TO 3
IF(W(3).LT.0.0) GO TO 1
PC(3) = SPLT(ISECT)
GO TO 2
1 PC(3) = SPLT(ISECT - 1)
ISECTT = ISECT - 1
2 CONTINUE
SLRC = (PC(3) - PS(3))/W(3)
PC(2) = PS(2) + W(2)*SLRC
PC(1) = PS(1) + W(1)*SLRC
3 RETURN
END
```

BEST AVAILABLE COPY

C  
C

```
SUBROUTINE EDIT
COMMON/MANN/MAN(110,55,6),NTEST
COMMON/EVAL/VI(200),NVI(200),VI2(200),FACT(11)
COMMON/TRAC/SLR,XL1,XL2,XL3,NHITS,DIST(2,10),IDIST(10),NCELL
COMMON/SORS/WP(3),W(3),P(3),PO(3),BULSI(3),SPCT,ISEED,JSEED,
  * SNT,CST
100 FORMAT(1H )
101 FORMAT(A10,5X,A10,A1,2X,A10,A1,5X,A7)
102 FORMAT(110,A5,E11.4,A2,E11.4,A5,17)
103 FORMAT(5A10)
110 FORMAT(1H1)
CAP1 = 10H INCREMENTS CAP2 = 5H. * *5 CAP3 = 2H*
CAP4 = 7HND RAYS      CAP5 = 10H STAND DEVS CAP6 = 1H
CAP7 = 10H VUL INDES CAP8 = 1HXSCAP9=10H- - -
CALL VAR
IZ=0
10 WRITE(6,110)
WRITE(6,101)CAP1,CAP7,CAP8,CAP5,CAP6,CAP4
WRITE(6,103) CAP9,CAP9,CAP9,CAP9,CAP9
11 IZ = IZ + 1
112 WRITE(6,102) IZ,CAP2,VI(IZ),CAP3,VI2(IZ),CAP2,NVI(IZ)
IF(NVI(IZ+1).EQ.0) GO TO 20
IF(MOD(IZ,50).EQ.0) GO TO 10
GO TO 11
20 RETURN
END
```

C  
C

```
SUBROUTINE EVEN(JEXIT)
COM BRINGS PBDY TO EVEN NUMBER OF DELR OR EXITS BOX OR CROSSES SECTION
COMMON/MANN/MAN(110,55,6),NTEST
COMMON/LATEN/JLAT(2,100),NLAT(20),ILAT,NLATT,ISECT,ILATX,ISECTT,
  * PLAT(4,100),PTAR(6,100),PBDY,PBOX,SCPT,NDEL,IMIST,IPT,NARC,NPTS
COMMON/EVAL/VI(200),NVI(200),VI2(200),FACT(11)
COMMON/SORS/WP(3),W(3),P(3),PO(3),RULSI(3),SPCT,ISEED,JSEED,
  * SNT,CST
COMMON/TRAC/SLR,XL1,XL2,XL3,NHITS,DIST(2,10),IDIST(10),NCELL
COMMON/POINT/PS(3),PR(3),PE(3),PC(3),DELR,NFL03
JEXIT = 0
7 CALL CROSS(SLRC)
PDEL = PBDY - DELR*FLOATF(NDEL)
Z9 = DELR * PDEL
IF(SLRC.LE,Z9) GO TO 20
IF(SLR = PBOX.LT,Z9 + 1.0E-10) GO TO 28
CALL LATS(PS,I,J,K)
CALL LOCATE(K,ISECTT)
IF(MAN(I,J,K).EQ.0) GO TO 31
RN1 = URAN31(ISEED)
```

# BEST AVAILABLE COPY

```
DO 2 IZ = 1,3
PE(IZ) = PS(IZ) + Z9*W(IZ)
PR(IZ) = PS(IZ) + Z9*W(IZ)*RNI
2 CONTINUE
CALL LAVIS(PE,I,J,K)
CALL LOCATE(K,ISECTY)
IF(MAN(I,J,K).EQ.0) GO TO 3
CALL LAVIS(PH,I,J,K)
CALL LOCATE(K,ISECTY)
IF(MAN(I,J,K).EQ.0) GO TO 3
CALL SCORE(SC,ISC,I,J,K)
SCPT = SCPT + Z9*SC*FACT(ISC)/DELR
INX = 11
WRITE(6,1000) PS,PE,W,INX
1000 FORMAT(9(E12.4,1X),/,15)
DO 130 IZ = 1,3
PS(IZ) = PE(IZ)
130 CONTINUE
PBDY = DELR*FLOATF(NDEL+1)  S PBOX = PBOX + Z9
GOTO 51
3 CALL TRACE(Z9)
IZ = 0
5 IZ = IZ + 1
IF(DIST(2,IZ).EQ.0.0) GO TO 14
PDEL = PDEL + DIST(1,IZ)
ISC = IDIST(IZ)
IF(PDEL.LT.DE LR) GO TO 17
DIST(1,IZ) = DIST(1,IZ) + PDEL + DELR
NCELL = IZ
17 SCPT = SCPT + DIST(1,IZ)*DIST(2,IZ)*FACT(ISC)/DELR
PBDY = PBDY + DIST(1,IZ)
14 PBOX = PBOX + DIST(1,IZ)
DO 6 IX = 1,3
PS(IX) = PS(IX) + DIST(1,IZ)*W(IX)
6 CONTINUE
IF(IZ.LT.NCELL) GO TO 5
INX = 12
WRITE(6,1000) PS,PE,W,INX
IF(PDEL.LT.DE LR) GO TO 7
GOTO 51
8 CALL CROSS(SLRC)
31 IF(SLRC.LE.Z9) GO TO 20
IF(SLRC - PBOX.LT.Z9 + 1.0E-10) GO TO 28
DO 9 IZ = 1,3
PE(IZ) = PS(IZ) + Z9*W(IZ)
9 CONTINUE
CALL LAVIS(PE,I,J,K)
CALL LOCATE(K,ISECTY)
IF(MAN(I,J,K).GT.0) GO TO 3
DO 62 IZ = 1,3
```

# BEST AVAILABLE COPY

```
PS(IZ) = PE(IZ)
62 CONTINUE
PBOX = PBOX + Z9
Z9 = DELR
INX = 13
WRITE(6,1000) PS,PE,W,INX
GOTO 8
20 IF(SLR = PBOX.LT.SLRC + 1.0E-10) GO TO 28
IFLAG = 1 $ Z9 = SLRC
GO TO 29
28 IFLAG = 0 $ Z9 = SLR = PBOX
29 CALL TRACE(Z9)
INX = 14
WRITE(6,1000) PS,PE,W,INX
IZ = 0 $ IFLAG2 = 0
21 IZ = IZ + 1
IF(DIST(2,IZ).EQ.0.0) GO TO 24
ISC = IDIST(IZ)
PDEL = PDEL + DIST(1,IZ)
IF(PDEL.LT.DELR) GO TO 27
DIST(1,IZ) = DIST(1,IZ) = PDEL + DELR
NCELL = IZ $ IFLAG2 = 1
27 SCPT = SCPT + DIST(1,IZ)*DIST(2,IZ)*FACT(ISC)/DELR
PBUY = PBUY + DIST(1,IZ)
24 PBOX = PBOX + DIST(1,IZ)
DO 25 IX = 1,3
PS(IX) = PS(IX) + DIST(1,IZ)*W(IX)
25 CONTINUE
IF(IZ.LT.NCELL) GO TO 21
IF(IFLAG2.EQ.1) GO TO 51
JEXIT = 1
IF(IFLAG.EQ.0) GO TO 52
IF(NFLG3.EQ.3) NLATT = 0
IF(ILATX.GT.1) GO TO 40
NLATT = NLATT + 1 $ IZ = NLATT
GO TO 41
40 IZ = ILAT
41 CONTINUE
NLAT(ISECTT) = NLAT(ISECTT) + 1
CALL LATENT(1,IZ,PC)
IF(NFLG3.LT.3) GO TO 100
CALL LOAD(ISECTT)
ISECT = ISECTT $ ILAT = 1
GO TO 100
52 Z9 = DELR*FLOATF(NDEL)
IF(PBUY.EQ.Z9) GO TO 50
RN1=ORANJ1(ISEED)
IF(RN1.GT,PBUY-Z9)GO TO 50
SC = SCPT*DELR/(PBUY - Z9)
GO TO 51
```

# BEST AVAILABLE COPY

```
50 SC = 0.0 $ GO TO 100
51 NDEL = NDEL + 1
   CALL TALLY(NDEL,SC,11)
100 SCPT = 0.0
   RETURN
   END
```

C  
C

```
   SUBROUTINE HEMI
   COMMON/SORS/WP(3),W(3),P(3),PO(3),BULS1(3),SPCT,ISEED,JSEED,
   * SNT,CST
   COMMON/HISC/R1,R2,GCR,GCH,NFLG1,NFLG2,SD1,SU2,OW1,OW2
   IF(NFLG1.EQ.8)GOTO 6
   CSTHT = URANJ1(ISEED)
   W=R2
   GOTO 1
6 W=SQRTF(GCR**2+GCH**2)
   CSTHT=GCR/R
1 X1 = URANJ1(ISEED)
   X2 = URANJ1(ISEED)
   X1S=X1**2
   X2S=X2**2
   T=X1S+X2S
   IF(T.GE.1.0)GOTO 1
   CSPHI=(X1S-X2S)/T
   SNPHI=2.0*X1*X2/T
   X1 = URANJ1(ISEED)
   IF(X1.GT.0.5) GOTO 2
   SNPHI=-SNPHI
2 SNTHT=SQRTF(1.0-CSTHT**2)
   WP(1)=SNTHT*SNPHI
   WP(2)=SNTHT*CSPHI
   WP(3)=CSTHT
   IF(NFLG1.EQ.2) GOTO 8
   IF(NFLG1.GT.3) GOTO 3
   WP(3)=-WP(3) $ GOTO 8
3 IF(NFLG1.GT.5) GOTO 5
   X1=ABSF(WP(2))
   WP(3)=X1*WP(3)/WP(2)
   IF(NFLG1.GT.4) GOTO 4
   WP(2)=X1 $ GOTO 8
4 WP(2)=-X1 $ GOTO 8
5 IF(NFLG1.GT.7) GOTO 8
   X1=ABSF(WP(1))
   WP(3)=X1*WP(3)/WP(1)
   IF(NFLG1.GT.6) GOTO 7
   WP(1)=X1 $ GOTO 8
7 WP(1)=-X1
8 DO 10 I=1,3
   PO(I)=R*WP(I) $ WP(I)=WP(I)
```

# BEST AVAILABLE COPY

```

      P(1) = P0(1)
10  CONTINUE
      RETURN
      END

C
C
      SUBROUTINE INX3
COM  CALCULATES EXIT POINT FROM CELL
      COMMON/TRAX/P1(3),P2(3),BOUND(6),PL1,PL2,PL3,Z9
      COMMON/SORS/WP(3),W(3),P(3),P0(3),BULSI(3),SPCT,ISEED,JSEED,
      * SNT,CST
      IF(W(1).EQ.0.0) GO TO 2
      P2(1) = PL1
      Z9 = (PL1 - P1(1))/W(1)
      P2(2) = P1(2) + Z9*W(2)
      P2(3) = P1(3) + Z9*W(3)
      IF(P2(2).GT.BOUND(4).OR.P2(2).LT.BOUND(3)) GO TO 2
      IF(P2(3).GT.BOUND(6).OR.P2(3).LT.BOUND(5)) GO TO 2
      GO TO 5
2  IF(W(2).EQ.0.0) GO TO 3
      P2(2) = PL2
      Z9 = (PL2 - P1(2))/W(2)
      P2(1) = P1(1) + Z9*W(1)
      P2(3) = P1(3) + Z9*W(3)
      IF(P2(1).GT.BOUND(2).OR.P2(1).LT.BOUND(1)) GO TO 3
      IF(P2(3).GT.BOUND(6).OR.P2(3).LT.BOUND(5)) GO TO 3
      GO TO 5
3  IF(W(3).EQ.0.0) GO TO 4
      P2(3) = PL3
      Z9 = (PL3 - P1(3))/W(3)
      P2(1) = P1(1) + Z9*W(1)
      P2(2) = P1(2) + Z9*W(2)
      IF(P2(1).GT.BOUND(2).OR.P2(1).LT.BOUND(1)) GO TO 4
      IF(P2(2).GT.BOUND(4).OR.P2(2).LT.BOUND(3)) GO TO 4
      GO TO 5
4  CONTINUE
COM  TRACE ERROR
5  CONTINUE
      RETURN
      END

C
C
      SUBROUTINE ISOT
      COMMON/SORS/WP(3),W(3),P(3),P0(3),BULSI(3),SPCT,ISEED,JSEED,
      * SNT,CST
      COMMON/MISC/R1,R2,GCR,GCH,NFLG1,NFLG2,SD1,SD2,GW1,GW2
      X1 = URANS1(ISEED)
      IF(R2.GE.R1) GOTO 1
      CSTHT=2.0*X1-1.0 $ SPCT=1.0
      GOTO 2

```

# BEST AVAILABLE COPY

```

1 X2 = SQRT(R2**2 - R1**2)/R2
  SPCT = 1.0 - X2
  CSTHT = 1.0 - X1*(1.0 - X2)
2 X1 = URANJ1(ISEED)
  X2 = URANJ1(ISEED)
  X1S=X1**2
  X2S=X2**2
  T=X1S+X2S
  IF(T.GE.1.0) GO TO 2
  CSPHI=(X1S-X2S)/T
  SNPHI=2.0*X1*X2/T
  X1 = URANJ1(ISEED)
  IF(X1.GT.0.5) GOTO 3
  SNPHI=-SNPHI
3 T1=WP(1)**2+WP(2)**2
  T3=1.0-CSTHT**2
  T4=T3/T1
  IF(T1.GT.1.0E-10)GOTO 4
  SNTHT=SQRT(T3)
  W(1)=SNTHT*CSPHI
  W(2)=SNTHT*SNPHI
  W(3)=CSTHT*WP(3)
  GOTO 5
4 T2=SQRT(T4)
  W(1)=T2*(WP(1)*WP(3)*SNPHI+WP(2)*CSPHI)+WP(1)*CSTHT
  W(2)=T2*(WP(2)*WP(3)*SNPHI-WP(1)*CSPHI)+WP(2)*CSTHT
  W(3)=WP(3)*CSTHT-T1*T2*SNPHI
5 RETURN
  END

```

C  
C

```

COM SUBROUTINE LATENT(IL,IP,P1)
  STORES AND RETRIEVES LATENT RAYS
  DIMENSION PI(3)
  COMMON/MANV/MAN(110,55,6),NTEST
  COMMON/LATEN/JLAT(2,100),NLAT(20),ILAT,NLATT,ISECT,ILATX,ISECTT,
  * PLAT(4,100),PTAR(6,100),PBDY,PBOX,SCPT,NDEL,IHIST,IPT,NARC,NPTS
  COMMON/SORS/WP(3),W(3),P(3),PO(3),BULSI(3),SPCT,ISEED,JSEED,
  * SNT,CST
  COMMON/TRAC/SLR,XL1,XL2,XL3,NHITS,DIST(2,10),IDIST(10),NCELL
  IF(IL.GT.0) GO TO 2
  PBDY = PLAT(1,IP)
  PBOX = PLAT(2,IP)
  SLR = PLAT(3,IP)
  SCPT = PLAT(4,IP)
  DO 1 IZ = 1,3
  PI(IZ) = PTAR(IZ,IP)
  W(IZ) = PTAR(IZ+3,IP)
1 CONTINUE
  ISECT = JLAT(1,IP)

```

# BEST AVAILABLE COPY

```

NDEL = JLAT(2,IP)
GO TO 10
2 PLAT(1,IP) = PBDY
  PLAT(2,IP) = PROX
  PLAT(3,IP) = SLR
  PLAT(4,IP) = SCPT
  DO 3 IZ = 1,3
    PTAR(IZ,IP) = PI(IZ)
    PTAR(IZ+3,IP) = N(IZ)
3 CONTINUE
  JLAT(1,IP) = ISECTY
  JLAT(2,IP) = NDEL
10 RETURN
  END

C
C
SUBROUTINE LATIS(PI,I,J,K)
DIMENSION PI(3),PH(3)
COMMON/SORS/WP(3),W(3),P(3),PO(3),BULST(3),SPCT,ISEED,JSEED,
* SNT,CST
COMMON/LATIX/GRID(4),BKPT,IBKPT,SPLT(19)
COMMON/LAT/NX,NY,NZ,NSECT,ISL
COMMON/INDEX/I1,J1,K1
COM LOCATES CELL I,J,K FOR POINT P
DO 13 IZ=1,3
  PH(IZ)=PI(IZ)
13 CONTINUE
19 IZ=0
12 IZ=IZ+1
  IF(IZ.LT.3) GO TO 102
  IF(PI(3).GT.BKPT) GO TO 103
102 X=PI(IZ)/GRID(IZ)
  GO TO 104
103 XBKPT=FLDADF(1BKPT)
  X=XBKPT+(PI(3)-BKPT)/GRID(4)
104 K=XFIX(X)
  IF(X.LT.0.0) K=K-1
  KK=K
  IF(W(IZ).GT.0.0) KK=K+1
  XK=FLUADF(KK)
  XXX=X-XK
  IF(W(IZ).GT.0.0) XXX=-XXX
  IF(XXX.GT.1.0E-12) GO TO 107
  IF(IZ.LT.3) GO TO 105
  IF(KK.LT.1BKPT) GO TO 105
  PI(3)=BKPT+(XK-XBKPT)*GRID(4)
  PH(3)=PI(3) $GO TO 3
105 PI(IZ)=XK*GRID(IZ)
  PH(IZ)=PI(IZ) $ GO TO 3
107 GOTO (1,2,10),IZ

```

BEST AVAILABLE COPY

```
1 I=K $GO TO 12
2 J=K $GO TO 12
3 CONTINUE
  DO 4 IZ = 1,3
  PI(IZ) = PI(IZ) + 0.000001*W(IZ)
4 CONTINUE
  GO TO 19
10 DO 11 IZ = 1,3
  PI(IZ)=PH(IZ)
11 CONTINUE
  I = I + 1 $ J = J + 1 $ K = K + 1
  I1 = I $ J1 = J $ K1 = K
  IF(I.GT.NX) I = I - 1
  IF(I.LT.1) I = I + 1
  IF(J.GT.NY) J = J - 1
  IF(J.LT.1) J = J + 1
  IF(K.GT.NZ) K = K - 1
  IF(K.LT.1) K = K + 1
  RETURN
  END

C
C
C SUBROUTINE LOAD(ISECT)
C LOADS LETHALITY DESCRIPTION INTO CORE
C ISECT EQUALS SECTION NUMBER TO BE LOADED(RECORD) 1-3
C NSECT EQUALS NUMBER OF SECTIONS (3)
C NREM EQUALS NUMBER OF VOID SLICES IN LAST SECTION (?)
C COMMON/MANN/MAN(110,55,6),NTEST
C COMMON/LAT/NX,NY,NZ,NSECT,ISL
C DIMENSION NAME(4)
C DEFINE DISC FILES TO BE UTILIZED
C DATA DISC2/10H55D24 KMAN/
C DATA FILE2/10HBODY PARTS/
C OPEN THE DISC
C CALL DISCOL(3,NAME,DISC2)
C IF(NTEST.EQ.1) GO TO 999
C CALL DISCRU(FILE2,ISECT,1,MAN)
C GO TO 1000
999 CALL TEST(ISECT)
1000 RETURN
  END

C
C
C SUBROUTINE LOCATE(KP,IS)
C COMMON/LATIX/GRID(4),BKPT,IBKPT,SPLT(19)
C COMMON/LAT/NX,NY,NZ,NSECT,ISL
COM CALCULATES MAN K FROM POINT K
  IS=0
21 IS=IS+1
  IF(KP.GT.IS*ISL)GOTO 21
```

BEST AVAILABLE COPY

```
KP = KP - ISL*(IS - 1)
RETURN
END
```

C  
C

```
SUBROUTINE SCORE(SC,ISC,I,J,K)
COMMON/MANV/MAN(110,55,6),NTEST
ISC = MAN(I,J,K)
SC = FLOATF(ISC)
RETURN
END
SUBROUTINE SORC
COMMON/LATEN/JLAT(2,100),NLAT(20),ILAT,NLATT,ISECT,ILATX,ISECTT,
* PLAT(4,100),PTAR(6,100),PHDY,PBOX,SCPT,NDEL,IHIST,IPT,NARC,NPTS
COMMON/SORS/WP(3),W(3),P(3),PO(3),BULSI(3),SPCT,ISEED,JSEED,
* SNT,CST
COMMON/TRAC/SLR,XL1,XL2,XL3,NHITS,DIST(2,10),IDIST(10),NCELL
COMMON/MISC/K1,R2,GCR,GCH,NFLG1,NFLG2,SD1,SD2,GW1,GW2
IF(IHIST.GT.1) GO TO 12
IF(NFLG1.GT.1) GO TO 8
IF(NPTS.GT.1) GO TO 2
DO 1 I = 1,3
P(I) = PD(I)
1 CONTINUE
GO TO 4
2 DO 3 I = 1,3
P(I) = PTAR(I,IPT)
W(I) = PTAR(I+3,IPT)
3 CONTINUE
4 CONTINUE
IF(NFLG2.GT.2) GO TO 13
IF(NFLG2.GT.1) GO TO 5
R2 = (P(1) - 0.5*XL1)**2
R2 = R2 + (P(2) - 0.5*XL2)**2
R2 = R2 + (P(3) - 0.5*XL3)**2
R2 = SQRTF(R2)
GO TO 11
5 CONTINUE
R2 = 0.0
DO 6 I = 1,3
R2 = R2 + (P(I) - BULSI(I))**2
6 CONTINUE
R2 = SQRTF(R2)
DO 7 I = 1,3
WP(I) = (BULSI(I) - P(I))/R2
7 CONTINUE
GO TO 12
8 IF(NFLG1.GT.8) GO TO 9
CALL HEMI
GO TO 10
```

# BEST AVAILABLE COPY

```
9 CALL ARC(NARC)
10 CALL TRANS
11 WP(1) = (0.5*XL1 - P(1))/R2
   WP(2) = (0.5*XL2 - P(2))/R2
   WP(3) = (0.5*XL3 - P(3))/R2
12 CONTINUE
   CALL AD
13 RETURN
   END
```

C  
C

```
   SUBROUTINE TALLY(IP,SC,ISC)
   COMMON/MANN/MAN(110,55,6),NTEST
   COMMON/EVAL/VI(200),NVI(200),VI2(200),FACT(11)
   SC = SC*FACT(ISC)
   VI(IP) = VI(IP) + SC
   VI2(IP) = VI2(IP) + SC*SC
3   NVI(IP) = NVI(IP) + 1
   RETURN
   END
```

C  
C

```
   SUBROUTINE TEST(ISECT)
   COMMON/MANN/MAN(110,55,6),NTEST
   DO 12 I = 1,110
   DO 12 J = 1,55
   DO 12 K = 1,6
   MAN(I,J,K) = 0
12 CONTINUE
   DO 1 I = 50,60
   DO 1 J = 1,55
   DO 1 K = 1,6
   MAN(I,J,K) = I - 50
1 CONTINUE
   DO 2 J = 24,32
   DO 2 I = 1,110
   DO 2 K = 1,6
   MAN(I,J,K) = J - 23
2 CONTINUE
   IF(ISECT.GT.8) GO TO 5
   IF(ISECT.LT.7) GO TO 5
   DO 3 I = 1,110
   DO 3 J = 1,55
   DO 3 K = 1,6
   MAN(I,J,K) = 10
3 CONTINUE
5 RETURN
   END
```

C  
C

# BEST AVAILABLE COPY

```

SUBROUTINE TRACE(R)
COM  TRACKS THROUGH CELLS AND STORES SCORES AND DISTANCES
COMMON/SORS/WP(3),W(3),P(3),PO(3),BULST(3),SPCT,ISEED,JSEED,
* SNT,CST
COMMON/TRAX/P1(3),P2(3),BOUND(6),PL1,PL2,PL3,Z9
COMMON/TRAC/SLR,XL1,XL2,XL3,NHITS,DIST(2,10),IDIST(10),NCELL
COMMON/LATIX/GRID(4),BKPT,IRKPT,SPLT(19)
COMMON/POINT/PS(3),PR(3),PE(3),PC(3),DELR,NFLG3
COMMON/INDEX/I1,J1,K1
COM  INITIALIZE ARRAYS
DO 11 J = 1,5
  IDIST(J) = 0
DO 11 I = 1,2
  DIST(I,J) = 0.0
11 CONTINUE
DO 12 J = 1,3
  P1(J) = PS(J)
12 CONTINUE
NCELL = 0 $ PCELL = 0.0
10 NCELL = NCELL + 1
  CALL LATTIS(P1,I,J,K)
  PL1 = GRID(1)*FLOATF(I1)
  PL2 = GRID(2)*FLOATF(J1)
  IF(K.GT.IRKPT) GO TO 1
  PL3 = GRID(3)*FLOATF(K1)
  ZN1 = GRID(3) $ GO TO 2
1  PL3 = BKPT + GRID(4)*FLOATF(K1 - IRKPT)
  ZN2 = GRID(4)
2  CONTINUE
  BOUND(2) = PL1
  BOUND(4) = PL2
  BOUND(6) = PL3
  BOUND(1) = PL1 - GRID(1)
  BOUND(3) = PL2 - GRID(2)
  BOUND(5) = PL3 - ZN2
  IF(W(1).LT.0.0) PL1 = PL1 - GRID(1)
  IF(W(2).LT.0.0) PL2 = PL2 - GRID(2)
  IF(W(3).LT.0.0) PL3 = PL3 - ZN2
  CALL INXS
  CALL LOCATE(K,ISECTT)
  CALL SCORE(SC,ISC,I,J,K)
  DIST(2,NCELL) = SC
  IDIST(NCELL) = ISC
  IF(PCELL + Z9.GE.K) GO TO 14
  DIST(1,NCELL) = Z9
  PCELL = PCELL + Z9
  DO 13 IZ = 1,3
    P1(IZ) = P2(IZ)
13 CONTINUE
  IF(R = PCELL.LT.10E-12) GO TO 7

```

# BEST AVAILABLE COPY

```

      GO TO 10
14  DIST(1,NCELL) = R - PCELL
      7 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE TRACK(IL)
      COMMON/HANN/MAN(110,55,6),NTEST
      COMMON/LATEN/JLAT(2,100),NLAT(20),ILAT,NLATT,ISECT,ILATX,ISECTT,
      * PLAT(4,100),PTAR(6,100),PBDY,PBOX,SCPT,NDEL,IHIST,IPT,NARC,NPTS
      COMMON/EVAL/V1(200),NV1(200),V12(200),FACT(11)
      COMMON/SORS/WP(3),W(3),P(3),PO(3),BULSI(3),SPCT,ISEED,JSEED,
      * SNT,CST
      COMMON/TRAC/SLR,XL1,XL2,XL3,NHITS,DIST(2,10),IDIST(10),NCELL
      COMMON/POINT/PS(3),PR(3),PE(3),PC(3),DELR,NFLG3
      COMMON/LATIX/GHID(4),BKPT,IBKPT,SPLT(19)
      COMMON/INDEX/I1,J1,K1
      COMMON/LODE/NLOAD
COM  TRACKS INITIAL RAYS,LATENTS,OR RAYS RETURNED FROM EVEN
      80 JEXIT = 0
      IF(IL.EQ.1) GO TO 120
      CALL LATENT(0,ILAT,PS)
      WRITE(6,525) ILAT,NLATT,ISECT,ISECTT,JEXIT,NDEL
525  FORMAT(10(2X,I5))
      JLAT(1,ILAT) = 0
      NLAT(ISECT) = NLAT(ISECT) + 1
      IF(PBDY.EQ.0.0) GO TO 2
      CALL EVEN(JEXIT)          $ GO TO 4
120  DO 1 IZ = 1,3
      PS(IZ) = P(IZ)
      1 CONTINUE
      INX = 5
      WRITE(6,1000) PS,PE,W,INX
1000 FORMAT(9(E12.4,1X),/,I5)
      PBDY = 0.0 $ PBOX = 0.0 $ NDEL = 0
      2 CONTINUE
      4 IF(JEXIT.EQ.1) GO TO 70
132  CALL LATTIS(PS,1,J,K)
      CALL LOCATE(K,ISECTT)
      IF(ISECT.EQ.ISECTT) GO TO 43
COM  STORE LATENT
      DO 25 IZ = 1,3
      PC(IZ) = PS(IZ)
      25 CONTINUE
133  IF(NFLG3.EQ.3) NLATT = 0
      IF(ILATX.GT.1) GO TO 40
      NLATT = NLATT + 1 $ IZ = NLATT
      GO TO 41
      40 IZ = ILAT

```

# BEST AVAILABLE COPY

```
41 CONTINUE
NLAT(ISECTT) = NLAT(ISECTT) + 1
CALL LATENT(1, IZ, PC)
IF(NFL33, LT, 3) GO TO 70
CALL LOAU(ISECTT)
ISECT = ISECTT $ ILAT = 1
GO TO 80
COM END OF LATENT STORAGE
43 CONTINUE
CALL CROSS(SLRC)
IF(SLRC.LE.DELR) GO TO 20
IF(SLK = PBOX.LT.DELR + 1.0E-10) GO TO 30
IF(MAN(I, J, K).EQ.0) GO TO 7
RN1 = URAN31(ISEED)
DO 82 IZ = 1, 3
PE(IZ) = PS(IZ) + DELR*w(IZ)
PR(IZ) = PS(IZ) + DELR*w(IZ)*RN1
82 CONTINUE
CALL LATIS(PE, I, J, K)
CALL LOCATE(K, ISECTT)
IF(MAN(I, J, K).EQ.0) GO TO 3
CALL LATIS(PR, I, J, K)
CALL LOCATE(K, ISECTT)
IF(MAN(I, J, K).EQ.0) GO TO 3
CALL SCORE(SC, ISC, I, J, K)
NDEL = NDEL + 1
PBDY = DELR*FLOATF(NDEL)
PBOX = PBOX + DELR
CALL TALLY(NDEL, SC, ISC)
INX = 1
WRITE(6, 1000) PS, PE, W, INX
DO 130 IZ = 1, 3
PS(IZ) = PE(IZ)
130 CONTINUE
GO TO 43
3 CALL TRACE(DELR)
DO 5 IZ = 1, NCELL
IF(DIST(2, IZ).EQ.0.0) GO TO 14
ISC = IDIST(IZ)
SCPT = SCPT + DIST(1, IZ)*DIST(2, IZ)*FACT(ISC)/DELR
PBDY = PBDY + DIST(1, IZ)
14 PBOX = PBOX + DIST(1, IZ)
5 CONTINUE
INX = 2
WRITE(6, 1000) PS, PE, W, INX
DO 6 IZ = 1, 3
PS(IZ) = PE(IZ)
6 CONTINUE
CALL EVEN(JEXIT)
GO TO 4
```

# BEST AVAILABLE COPY

```
8 CALL CROSS(SLRC)
7 IF(SLRC.LE.DELR) GO TO 20
IF(SLRC = PBOX.LT.DELR + 1.0E-10) GO TO 30
DO 9 IZ = 1,3
PE(IZ) = PS(IZ) + DELR*W(IZ)
9 CONTINUE
CALL LAVIS(PE,I,J,K)
CALL LOCATE(K,ISECTY)
IF(MAN(I,J,K).GT.0) GO TO 3
61 DO 62 IZ = 1,3
PS(IZ) = PE(IZ)
62 CONTINUE
PBOX = PBOX + DELR
GO TO 8
20 IF(SLRC = PBOX.LT.SLRC + 1.0E-10) GO TO 30
CALL TRACE(SLRC)
DO 21 IZ = 1,NCELL
IF(DIST(2,IZ).EQ.0.0) GO TO 24
ISC = IDIST(IZ)
SCPT = SCPT + DIST(1,IZ)*DIST(2,IZ)*FACT(ISC)/DELR
PBDY = PBDY + DIST(1,IZ)
24 PBOX = PBOX + DIST(1,IZ)
INX = 3
WRITE(6,1000) PS,PE,W,INX
21 CONTINUE
GO TO 133
30 CALL TRACE(SLR = PBOX)
DO 31 IZ = 1,NCELL
IF(DIST(2,IZ).EQ.0.0) GO TO 34
ISC = IDIST(IZ)
SCPT = SCPT + DIST(1,IZ)*DIST(2,IZ)*FACT(ISC)/DELR
PBDY = PBDY + DIST(1,IZ)
34 PBOX = PBOX + DIST(1,IZ)
31 CONTINUE
INX = 4
WRITE(6,1000) PS,PE,W,INX
Z8 = DELR*FLOAT(NDEL)
IF(PBDY.EQ.Z8) GO TO 70
RN1=URAN31(ISEED)
IF(RN1.GT.PBDY-Z8)GO TO 70
SC = SCPT+DELR/(PBDY - Z8)
51 NDEL = NDEL + 1
CALL TALLY(NDEL,SC,11)
70 SCPT = 0.0
100 CONTINUE
RETURN
END
```

C  
C

SUBROUTINE TRANS

# BEST AVAILABLE COPY

```

COM  TRANSLATES POINT
      COMMON/TRAC/SLR,XL1,XL2,XL3,NHITS,DIST(2,10),IDIST(10),NCELL
      COMMON/SORS/WP(3),W(3),P(3),PO(3),BULSI(3),SPCT,ISEED,JSEED,
      * SNT,CST
      COMMON/MISC/R1,R2,GCR,GCH,NFLG1,NFLG2,SD1,SD2,OW1,OW2
      P(1)=P(1)+0.5*XL1
      P(2)=P(2)+0.5*XL2
      IF(NFLG1,EQ.8) GO TO 100
      P(3)=P(3)+0.5*XL3
100  RETURN
      END

```

```

C
C
      SUBROUTINE VAR
      COMMON/MANV/MAN(110,55,6),NTEST
      COMMON/EVAL/VI(200),NVI(200),VI2(200),FACT(11)
      IZ=0
11  IZ=IZ+1
      IF(NVI(IZ).EQ.0) GOTO 20
      IF(NVI(IZ).GT.1) GOTO 12
      VI2(IZ)=9.99 $ GOTO 11
12  I=NVI(IZ)
      Z8=FLOATP(I)
      Z9=VI(IZ)/Z8
      VI(IZ)=Z9
      Z9=Z9*Z9*Z8
      Z9=VI2(IZ)-Z9
      IF(Z9.LT.0.0.AND.Z9.GT.-1.0E-06)Z9=0.0
      IF(Z9)21,15,16
21  WRITE(6,100)Z9
100  FORMAT(30X,'THE VALUE OF Z9 IS',2X,E12.5)
      GO TO 11
16  Z9=Z9/(Z8-1.0)
      Z9=SQRTF(Z9)
      VI2(IZ)=Z9
      GO TO 11
20  RETURN
      END

```

```

C
C
*   COMPILE DISC,NRAN31,ALL
*   COMPILE DISC,URAN31,ALL
*   LIST
*   DATA
      BULSI CHECK BIVAR DISTRIB
11111111 11113111
  1    1    1
  1    2    1    1  100
110   55   64   64    2  14
  6

```

# BEST AVAILABLE COPY

.5 E 00 .5 E 00 2.6 E 00 1.2 E 00 1.0 E 00  
.275E 02 .100E 04 .952E 02  
1.65 5.7  
.275E 02 .137E 02 .952E 02  
GO TO 11

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
12	Commander Defense Documentation Center ATTN: DDC-TCA Cameron Station Alexandria, VA 22314	1	Commander US Army Missile Research and Development Command ATTN: DRDMI-R Redstone Arsenal, AL 35809
1	Director of Defense Research and Engineering (OSD) Washington, DC 20305	1	Commander US Army Tank Automotive Research & Development Cmd ATTN: DRDTA-RWL Warren, MI 48090
1	Director Weapons Systems Evaluation Gp Washington, DC 20305	1	Commander US Army Mobility Equipment Research & Development Cmd ATTN: Tech Docu Cen, Bldg. 315 DRSME-RZT Fort Belvoir, VA 22060
1	Chairman Defense Science Board Washington, DC 20301	1	Commander US Army Armament Materiel Readiness Command Rock Island, IL 61201
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDMA-ST 5001 Eisenhower Avenue Alexandria, VA 22333	5	Commander US Army Armament Research and Development Command ATTN: DRDAR-LCS, Mr. G. Gaydos DRDAR-SCS, Mr. Mancini DRDAR-SCN, Mr. Malinoski DRDAR-SCH, Mr. H. Kahn DRDAR-SCA, Mr. Bornheim Dover, NJ 07801
1	Commander US Army Aviation Research and Development Command ATTN: DRSAV-E 12th and Spruce Streets St. Louis, MO 63166	1	Commander US Army Harry Diamond Labs ATTN: DRXDO-TI 2800 Powder Mill Road Adelphi, MD 20783
1	Commander US Army Air Mobility Research and Development Laboratory Ames Research Center Moffett Field, CA 94035		
1	Commander US Army Electronics Command ATTN: DRSEL-RD Fort Monmouth, NJ 07703		

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
2	Commander US Army Materials & Mechanics Research Center ATTN: Dr. R. Frost Mr. E. DeLuca Watertown, MA 02172	1	Commander US Army Aviation School ATTN: Aviation Agency Fort Rucker, AL 36362
2	Commander US Army Natick Research and Development Center ATTN: Mr. T. Keville Mr. M. Donomee Natick, MA 01760	1	Infantry Research & Dev Ln Ofc ATTN: COL John O. Ensor Fort Benning, GA 21905
1	Commander White Sands Missile Range ATTN: STEW-TE-PS, Mr. Findley White Sands, NM 88002	1	The Surgeon General Department of the Army Washington, DC 20310
1	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-SL (Tech Lib) White Sands Missile Range NM 88002	1	HQDA (DAMO-CSS, MAJ Rogers) Washington, DC 20310
1	Commander US Army Training and Doctrine Command ATTN: MAJ R. Young Fort Monroe, VA 23651	1	Chief of Naval Operations ATTN: OP-03EG Department of the Navy Washington, DC 20350
1	Commander US Army Infantry School ATTN: MAJ B. Davis Fort Benning, GA 31905	1	Commander US Naval Air Systems Command ATTN: AIR-604 Washington, DC 20360
1	Commander US Army Armor School ATTN: Armor Agency Fort Knox, KY 40121	2	Commander US Naval Weapons Center ATTN: Code 753 Code 12 China Lake, CA 93555
1	Commander US Army Field Artillery School ATTN: Field Artillery Agency Fort Sill, OK 73405	1	Commander US Naval Medical Field Research Laboratories Camp LeJeune, NC 28542
		1	Commander US Naval Research Laboratories ATTN: Code 8434, Mr. Ferguson Washington, DC 20375
		1	Commandant US Marine Corps Washington, DC 20380

DISTRIBUTUON LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>Aberdeen Proving Ground</u>
1	Director MCDEC ATTN: C, Gnd Cbt Div Quantico, VA 22154	Cdr, USAEA, ATTN: DRDAR-CLB-B, Mr. L. Sturdivan Dir, USAMSAA ATTN: Dr. Sperrazza DRXSY-GI, Mr. E. Heiss DRXSY-M, Mr. R. Pollard
1	Director MCDEC ATTN: MAJ D. Crowe Quantico, VA 22134	Dir, USAHEL ATTN: Mr. B. Corona Marine Corps Ln Ofc
1	AFATL (DLRW) Eglin AFB, FL 32542	
1	US Department of Treasury Training Center ATTN: Mr. W. McClarin Powdermill Road Beltsville, MD 20705	
1	Federal Bureau of Investigation ATTN: Mr. C. Cunningham, Rm 7410 Washington, DC 30405	
1	International Association of Chiefs of Police Gaithersburg, MD 20760	
1	Rockwell International Los Angeles Aircraft Division ATTN: Mr. W. Dotseth International Airport Los Angeles, CA 90009	