Written: June 1970
Distributed: August 1970

LA-4410
UC-32, MATHEMATICS
AND COMPUTERS
TID-4500

LOS ALAMOS SCIENTIFIC LABORATORY
of the
University of California
LOS ALAMOS • NEW MEXICO

ANDY1G2 and ANDY1R2, Monte Carlo Programs for Time-Dependent Monoenergetic Particle Transport in General Geometries and Repeating Arrays

by

D. R. Harris

Contents

Table

Figures

ANDY1G2 AND ANDY1R2, MONTE CARLO PROGRAMS

FOR TIME-DEPENDENT MONOENERGETIC PARTICLE TRANSPORT

IN GENERAL GEOMETRIES AND REPEATING ARRAYS

by

D. R. Harris

ABSTRACT

The ANDY1G2 and ANDY1R2 programs in FORTRAN IV for the CDC 6600 compute time-dependent, monoenergetic particle transport in, respectively, general geometries without repeating arrays, and general geometries with repeating arrays. In the latter case, moments of spatial distributions of collisions are computed at option. Particle emission from collisions is isotropic in the laboratory coordinate system and can be delayed.

I. INTRODUCTION

The ANDY series of Monte Carlo programs are developed to compute particle transport in situations where rapid computation and small program size are desired over an acceptable loss in detail of physical description. The physical approximations common to these programs are multigroup treatment of particle energy together with simplified patterns of particle emission after collision. These approximations permit use of well-understood multigroup cross section sets, and they provide extension of familiar multigroup S_N and diffusion theory methods to complex geometries. Geometrical and real-time properties of particle flight are not approximated.

The applications of the ANDY programs are sufficiently varied, requiring different routines for various particle sources, geometry options, tallies, etc., that a series of ANDY programs has been created rather than a single program with many options. In consequence inputs are simplified, and program modifications are facilitated. The ANDY programs are in FORTRAN IV for the CDC 6600, and those described in this report require less than 60_8K core storage.

In this report are described ANDY1G2 and ANDY1R2, where the first suffix, 1, represents the number of energy groups, the second suffix, G(eneral) or R(epeating), indicates the geometry treatment, and the third suffix, 2, is the version number. ANDY1G2 treats any general geometry which can be defined by combinations of possibly intersecting planes, spheres, ellipsoids, cones, and circular cylinders with any axial orientation. ANDY1R2 includes this general geometry description together with two treatments of repeating cells in an array. The two repeating cell treatments correspond to reflective and across-the-cell displacement boundary conditions. In the latter case, low-order moments of spatial distributions of collisions are computed. Particle emission from collisions is isotropic in the laboratory coordinate system* and can be delayed, as, e.g., following decay of a radioactive fission-product nucleus. Particle splitting occurs if the

*This restriction, which corresponds to the transport approximation used in many S_N calculations, is relaxed in other ANDY programs.

particle weight exceeds an input level, and particles are removed in a relatively unbiased manner if the particle weight drops below another input level.

Tallies provided in both programs include the weight of particles colliding in each spatial region and in each time box, the weight of particles crossing each surface segment in each direction in each time box, and the weight of particles leaking from the system in each time box. In addition an array is dimensioned, initialized, printed, and punched for tallies introduced at the discretion of the user. No provision is made for eigenvalue calculation such as that involved in calculation of reactor multiplication factor or decay constant.[1]

Program output is printed and/or punched. The punched card output from a run or from different runs of the same physical problem is processed and plotted by a FORTRAN IV program, ANDYAV (not described here). If punched cards from three or more runs of the same physical problem are supplied to ANDYAV, the latter program computes mean values and standard errors of means. Such standard errors provide realistic estimates of uncertainties in the Monte Carlo calculations.

Program notation and time box and surface descriptions are detailed in Appendix A. In Section II are described certain calculational techniques including the novel geometry routine (listed in Appendix B). Monte Carlo geometry considerations also are examined in Appendix C. Input is described in Section III and exemplified in Appendices D and E. Output is described in Section III and exemplified in Appendices F and G.

II. CALCULATIONS

The structure of the particle transport parts of the ANDY programs includes: (1) Choose source particle location, direction, and other required parameters; (2) Find the closest point where the particle trajectory passes into another region; (3) Determine if particle passes into the new region [then go back to (2)] or determine its point of collision in the present region; (4) If a collision occurs, multiply the particle weight by the secondaries per collision ratio and test for splitting or removal of particle; (5) Choose direction of emission and other parameters of a particle continued after collision. We discuss items (2), (3) and (5) in reverse order and finally comment on programming and testing.

A. Direction Finder

The routine for choice of direction in isotropic particle emission is conventional[2] and involves a division and a square root. Timing tests show that about one-quarter of problem execution time is spent in this, the most time-consuming part of problems with simple geometry.

B. Enter New Region or Collide

Suppose D represents the distance along a particle trajectory to the point of passage into a new region. Let Σ represent the macroscopic cross section in the present region. Choice of a single random number Q uniformly distributed on (0.,1.) determines if the particle indeed passes into the new region, $Q \varepsilon (0, e^{-\Sigma D})$, or collides at a distance x along the trajectory. We require

$$\int_0^x dx' \frac{\Sigma}{1 - e^{-\Sigma D}} e^{-\Sigma x'} = \int_{e^{-\Sigma D}}^{Q} dQ' \frac{1}{1 - e^{-\Sigma D}} , \quad (1)$$

so that

$$x = -(\Sigma^{-1}) \ln (1 + e^{-\Sigma D} - Q) . \quad (2)$$

C. ANDY General Geometry Routine

For most geometries the computation of the distance D is the most complicated and time-consuming section of ANDY1G2 and ANDY1R2. The general geometry routine which calculates D in ANDY appears to have complexity, speed and generality lying between the GEM treatment of shell geometries[3,4] and the very general geometry treatments of ØSR.[5] Here we introduce a topological entity, the surface segment, in a hierarchy between the usual entities, region and surface. A region from an operational point of view is a spatial domain in which cross sections are uniform or are specified by a single table or function. A surface, from an operational point of view, is a set of points in Euclidean space which satisfy an algebraic equation, usually linear or quadratic. A region is bounded in ANDY, not by surfaces, but by surface segments. A surface segment is defined as a set of points all of which lie in a specified containing surface and which may be all or only a segment of the surface. In the latter case, the segment

is defined by the intersections of other, bounding surfaces with the containing surface. In the ANDY programs a point in a surface segment which is only part of the containing surface is identified as having the proper parity or sense with respect to each of the bounding surfaces. Sense is defined as +1 outside a sphere or cylinder or ellipsoid and on the side of a plane toward which its normal points, and as -1 otherwise. We make this procedure more clear and illustrate its advantages by examples of program input (Section III). The geometrical relations among regions, surfaces, surface segments, and straight-line trajectories are examined in Appendix C.

D. Repeating Array Options in ANDYLR2

When a particle crosses the ISSth surface segment [identification number IAS(IR,ISS)] adjacent to region IR, it enters a region with identification number IAR(IR,ISS). If the latter is negative the particle is reflected (NCØN=0) back into region IR from the ISSth surface segment, or it is displaced (NCØN=1) across the cell into region |IAR(IR,ISS)|. Surface segments bounding cells must be planes normal to the x, y or z axes, so that the cells can be slabs, rectangular cylinders, or rectangular parallelopipeds. For hexagonal and rectangular arrays it appears that the cell displacement option is more general in that there are some periodic arrays which have no reflective cells. On the other hand, when a reflective cell exists it frequently is simpler than the equivalent displacement cell. Such a case is illustrated for a hexagonal lattice in Figure 1.

In the displacement option (NCØN=1), the particle coordinates x, y, and z are displaced to the opposite side of the cell; e.g., X, Y, and Z are transformed to X ± XCELL, Y, Z upon crossing a cell wall (7 and 9 in Figure 1) perpendicular to the x axis. Particle coordinates XCUM, YCUM, and ZCUM are not so displaced and thus represent the real particle location as it moves through the repeating array. Zeroth, first, and second-order spatial moments of colliding particle weights in various regions are computed, printed, and punched. Zeroth, first, and second-order moments of the spatial distributions of source particles also are determined.

E. Programming

A number of programming alternatives arise within the planned limitation to FORTRAN IV. Aside from removal of unnecessary divisions and multiplications (worth about 10% in execution time), the principal reduction in execution time (about 14%) arose from placing the random number generator in line, i.e., as FORTRAN statements where required in the main program, rather than as a FORTRAN or machine language* subroutine.

For the convenience of the user in adding special features to the programs, all variables are defined on comment cards and the geometry sections and time tally sections are described by comment cards. These are listed in Appendix A.

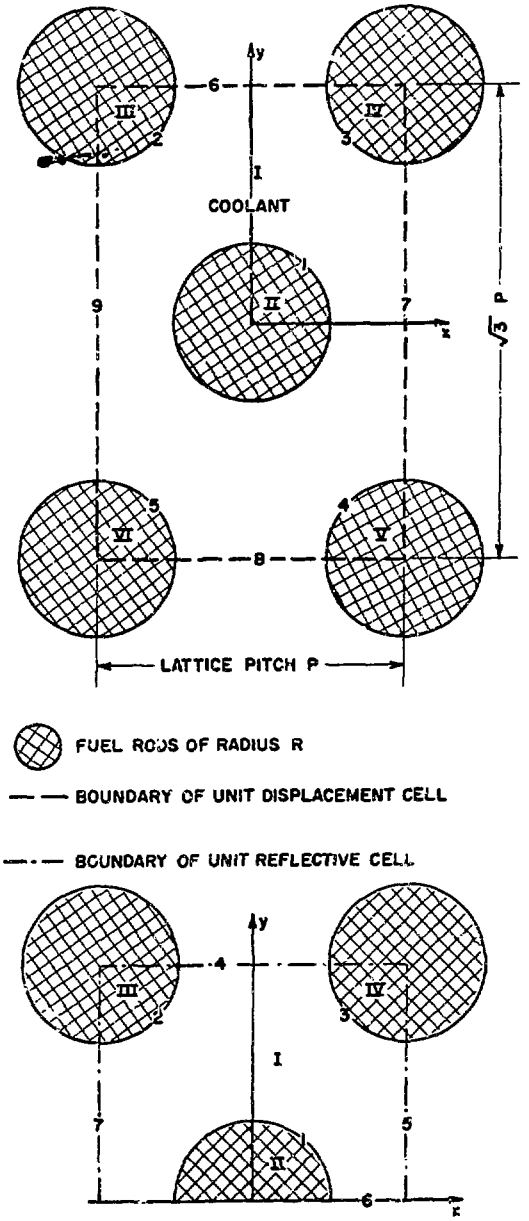

Fig. 1. Comparison of Displacement (top) and Reflective (bottom) Cells for a Hexagonal Lattice of Cylindrical Rods. Regions are Designated by Roman Numerals, and Surface Segments by Arabic Numerals.

*For this test, R. M. Frank created an optimized machine language version of the random number generating subroutine.

F. *Program Tests*

Various program tests have been carried out. Of these, the most convincing are direct comparison with S_N calculations and with known analytic solutions. In Figure 2 is shown a comparison of a neutron wave front development calculated by ANDY1G2 and by RZTRAN, a time-dependent S_N program developed by F. B. Lathrop. The Monte Carlo results are plotted with a bar for each of three independent runs (100000 starters and 140000 collisions per run), this permits a visual comparison between the Monte Carlo statistical uncertainties and the S_N mesh-spacing uncertainties.

A useful and apparently novel test is permitted for ANDY1R2 by: a. Construct a complicated but repeating lattice in the program; b. Set transport and absorption cross sections to be constant throughout the lattice; c. Compute the mean square distance to collision, $\langle r^2 \rangle$, and compare with the known analytic result, $2/\Sigma_{tr}\Sigma_a$. For example, for a hexagonal lattice like that shown in Figure 1 with fuel rod diameter, lattice pitch, transport cross section, and absorption cross section equal to 1.151 cm, 1.806 cm, 0.44 cm^{-1}, and 0.22 cm^{-1}, respectively, the Monte Carlo result for $\langle r^2 \rangle$ was 20.68 ± 0.14 cm^2 while the analytic result is 20.66 cm^2. Here the particles frequently cross region boundaries, a circumstance which tests the geometry portion of the program and also is responsible for the relatively long running time, 16 minutes in the central processor, for 40000 starters and 440000 collisions.

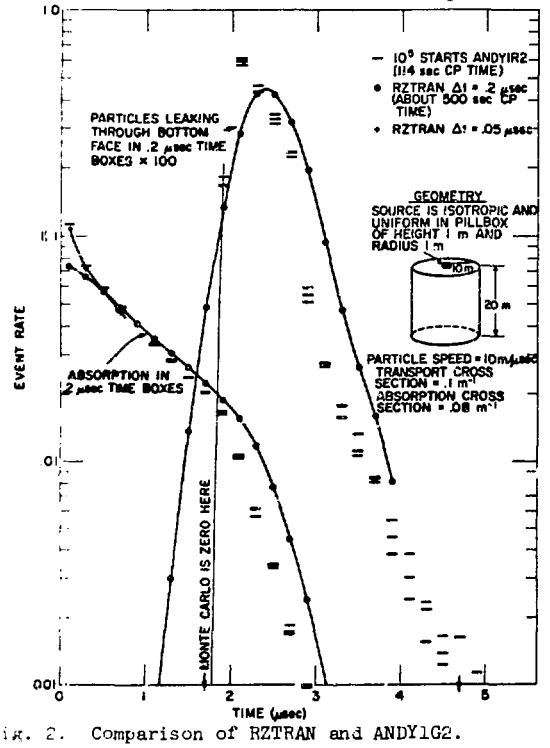

Fig. 2. Comparison of RZTRAN and ANDY1G2.

III. INPUT

Required input is listed in Table I. A sample input for ANDY1G2, listed in Appendix D, describes the time-dependent transport of neutrons from a point isotropic neutron burst near a pair of intersecting spheres similar to those illustrated in

Table I

INPUT FOR ANDY1G2 AND ANDY1R2*

Card	Format	Contents
1	Alphanumeric	Title
2	12I6	NS,NKRN,NIT,MØRE, NPUNCH
3	12I6	NT,ITB1
4	6E12.6	DELT1,DELT2, OFFSET,TSPLIT,TD
5	6E12.6	WSPLT,WCO,WCOCF, V,WDELF
ANDY1R2 Only	6E12.6	XCELL,YCELL,ZCELL
6	12I6	NREG,NSUR,NSEG, NMIX,NCON
7,···,	6E12.6	AS(IS),BS(IS),CS(IS), DS(IS),ES(IS),FS(IS)
(2*NSUR cards)	6E12.6	GS(IS),DS(IS)
A card set	12I6	IDS(ISS),NSEN(ISS)
for each	12I6	IDEN(ISS,IS1),IS1=1,
		no. sense relations
surface segment	12I6	ISEN(ISS,IS1),IS1=1,
		no. sense relations
A card set	12I6	NAS(IR)IMIX(IR)
for each region	12I6	IAS(IR,ISS),ISS=1,
		no. bounding surface segments
	12I6	IAR(IR,ISS),ISS=1,
		no. bounding surface segments
A card		
for each mix	6E12.6	SIG(IM),C(IM),
		BETA(IM)

*Variables are defined in Appendix A.

Figure C-1. This example is chosen to illustrate the point that the region of uniform cross sections enclosed by the intersecting spheres can be treated in ANDY as a single region bounded by two surface segments. In some geometry treatments, two or three regions might be required here. In Appendix E is listed a sample input for ANDY1R2 describing a moment calculation for a hexagonal lattice like that illustrated in Figure 1.

IV. OUTPUT

Input and output quantities and tallies are printed and punched. Printed quantities are labelled by their program variable names as defined in Section III, in Appendix A, and on comment cards at the beginning of the programs.

Tallies provided in the programs include: WIR(IT,IR), the weight of particles colliding in each region IR in each time box IT; WISP(IT,ISS), the weight of particles crossing each surface segment with identification number ISS from region IR, into region $IR_2 \geq IR_1$ in each time box IR; WISN(IT, ISS), the weight of particles crossing each surface segment ISS from region IR_2 into region $IR_1 < IR_2$ in each time box; and WLC(IT), the weight of particles leaking from the system in time box IT. In addition, an array, WIT(IT,N), is dimensioned, initialized, printed, and punched for arbitrary tallies at the discretion of the user. Finally, the total weight (WCUM) of colliding particles is printed and punched, as are the number of starter particles (NS), the number of secondaries (ISPLC), the number of times the bank dimension is exceeded (ISPLO), the total number of particle histories followed (NHS= NS+ISPLC), the number of collisions (NCOL), and the number of delayed particles (NDEL).

Moments of spatial distributions of collisions in various regions are printed for ANDY1R2 in the NCON=1 option.

In Appendices F and G are listed parts of the program outputs for the problems with inputs listed in Appendices D and E.

ACKNOWLEDGEMENTS

The author is grateful to R. M. Frank, J. I. Sackett, and G. D. Turner for assistance in various phases of this development.

REFERENCES

1. H. Rief and H. Kschwendt, "Reactor Analysis by Monte Carlo," Nucl. Sci. Eng. 30, 395 (1967).

2. E. D. Ceshwell and C. J. Everett, A Practical Manual on the Monte Carlo Method for Random Walk Problems, Pergamon Press, New York, 1959.

3. P. J. Hemmings, "The Gem Code," AHSB(s)R105. United Kingdom Atomic Energy Authority (1967).

4. G. E. Whitesides and N. F. Cross, "KENO-a Multigroup Monte Carlo Criticality Program," CTC-5. Union Carbide Corp. Nuclear Division (1969).

5. D. C. Irving, R. M. Freestone, Jr., and F. B. K. Kam, "Ø5R, a General-Purpose Monte Carlo Neutron Transport Code," ORNL-3622, Oak Ridge National Laboratory (1965).

APPENDIX A. NOTATION, TIME BOX SELECTION, AND SURFACE DESCRIPTIONS

```
C NOTATION
C      A1,B1,C1,....ETC=TEMPORARIES IN GEOMETRY ROUTINE.  OTHER TEMPORARIES
C           ARE FORMED BY ADDING SUFFIX T
C      AS(IS),BS(IS),....=PARAMETERS DEFINING SURFACE WITH ID NUMBER IS
C      BETA=DELAYED PARTICLE FRACTION
C      C(IM)=NUMBER OF SECONDARIES PER COLLISION WITH MIX NUMBER IM
C      D=PARTICLE FLIGHT PATH LENGTH TO SURFACE OR TO COLLISION POINT
C      DELT1=WIDTH OF TIME BOX IT (2.LE.IT.LE.ITB1). DELT1I IS INVERSE
C      DELT2=WIDTH OF TIME BOX IT (IT.GT.ITB1.AND.IT.LE.NT). DELT2I IS INVERSE
C      FRN=REAL NUMBER IN (0.,1.) FORMED BY PSEUDORANDOM NUMBER GENERATOR
C      IAR(IR,ISS)=ID NUMBER OF REGION ON OTHER SIDE OF ISS-TH SURFACE SEGMENT
C           FROM REGION IR. IAR(IR,ISS) EXCEEDS NREG IF ISS IS ON SYSTEM BOUNDARY
C           IAR(IR,ISS) IS NEGATIVE IF ISS IS ON CELL BOUNDARY
C      IAS(IR,ISS)=ID NUMBER OF ISS-TH SURFACE SEGMENT ADJACENT TO REGION WITH ID
C           NUMBER IR. (IR=1,NREG),(ISS=1,NAS(IR)).
C      IDEN(ISS,IS1)=ID NUMBER OF THE IS1-TH SURFACE BOUNDING SURFACE SEGMENT ISS
C      IDS(ISS)=ID NUMBER OF SURFACE CONTAINING SURFACE SEGMENT WITH ID ISS
C      IRSPL=REGION OF BANKED PARTICLE
C      IMIX(IR)=CROSS SECTION MIX NUMBER IN REGION IR
C      INDICES/IT FOR TIME BOX, IB FOR BLOCK, IS FOR SURFACE, ISS FOR SURFACE
C           SEGMENT, IST FOR STARTER, IR FOR REGION
C      IRMULT= MULTIPLIER IN PSEUDORANDOM NUMBER GENERATOR
C      IRS=REGION  OF STARTING PARTICLES
C      ISEN(ISS,IS1)=SENSE OF IS1-TH SURFACE BOUNDING SURFACE SEGMENT ISS
C      ISPL=NUMBER OF SECONDARY PARTICLES IN BANK
C      ISPLC=CUMULATIVE NUMBER OF SECONDARY PARTICLES
C      ISPLO=NUMBER OF SECONDARIES OVERFLOWING BANK
C      ITB1=LAST TIME BOX OF WIDTH DELT1 (1.LE.ITB1.LE.NT.LE.100)
C      ITP(IS)=TYPE NUMBER OF SURFACE IS
C           WHETHER PLANE (1), SPHERE (2), CYLINDER (3), ELLIPSOID (4) , CONE (5)
C      KRN=INTEGER PSEUDORANDOM NUMBER
C      MORE+1=NUMBER OF CASES IN THIS JOB
C      NAS(IR)=NUMBER OF SURFACE SEGMENTS ADJACENT TO (BOUNDING) REGION IR
C      NCOL=NUMBER OF COLLISIONS IN THIS JOB
C      NCON=0 IF A PARTICLE IS REFLECTED BACK INTO THE PARENT CELL FROM A SURFACE
C           SEGMENT ACROSS WHICH A NEGATIVELY NUMBERED REGION IS ENCOUNTERED.
C           WHEN NCON=0 NO SPATIAL COLLISION MOMENTS ARE COMPUTED
C      NCON=1 IF A PARTICLE IS DISPLACED ACROSS PARENT CELL FROM A SURFACE SEG-
C           MENT ACROSS WHICH A NEGATIVELY NUMBERED REGION IS ENCOUNTERED. WHEN
C           NCON=1 SPATIAL COLLISION MOMENTS ARE COMPUTED USING XCUM,YCUM, ZCUM.
C      NDEL=NUMBER OF DELAYED PARTICLES FOLLOWED IN JOB
C      NIT=NUMBER OF INTEGRAL TALLIES
C      NKRN=FACTOR (1,3,5,7,....) FOR STARTING PSEUDORANDOM NUMBER GENERATOR
C      NMIX=NUMBER OF CROSS SECTION MIXES
C      NPUNCH=0 IF NO INPUT AND OUTPUT PUNCHING. NON-ZERO OTHERWISE
C      NREG=NUMBER OF REGIONS (1.LENREG.LE.20)
C      NS=NUMBER OF SOURCE PARTICLES
C      NSEG=NUMBER OF SURFACE SEGMENTS (1.LE.NSEG.LE.40)
C      NSEN(ISS)=NUMBER OF SURFACES BOUNDING SURFACE SEGMENT ISS
C      NSUR=NUMBER OF SURFACES (1.LE.NSUR.LE.20)
C      NT=NUMBER OF TIME BOXES (1,.LE.ITB1.LE.NT.LE.100)
C      OFFSET=UPPER LIMIT OF FIRST TIME BOX (0.LE.OFFSET.LT.DELT1).  IF
C           OFFSET=0., TIME BOX IT=1 WILL BE EMPTY, IE, FIRST NON-ZERO TALLY
C           WILL BE IN TIME BOX IT=2
C      RMOD=PERIOD OF PSEUDORANDOM NUMBER GENERATOR
C      RNS=NUMBER OF SOURCE PARTICLES IN PROBLEM (RNS=NS*NB)
C      SIG(IM)=MACROSCOPIC CROSS SECTION FOR COLLISION WITH MIX IM. SIGI(IM) IS
C           INVERSE
C      T=CUMULATIVE PARTICLE FLIGHT TIME=REAL TIME AGE OF PARTICLE AT EVENT TIME
C      TB(J)=UPPER BOUND OF TIME BOX J=1,2,...,NT
C      TD=DELAY TIME OF DELAYED PARTICLE
C      TSPL=REAL TIME AGE OF BANKED PARTICLE
C      TSPLT=A PARTICLE SPLITS IF IT LIVES LONGER THAN TSPLT
C      UX,UY,UZ=DIRECTION COSINES OF PARTICLE
C      V=PARTICLE SPEED.  VI IS INVERSE SPEED
C      WCO=A PARTICLE HISTORY IS TERMINATED IF ITS WEIGHT DROPS BELOW WCO
C      WCOC=CUMULATIVE WEIGHT OF TERMINATED PARTICLES
C      WCOCE=VALUE WHICH IF EXCEEDED BY WCOC PERMITS A LOW WEIGHT (W.LT.WCO)
C           PARTICLE TO CONTINUE WITH WEIGHT WCOC
```

APPENDIX A. (Continued)

```
C      WCUM=CUMULATIVE WEIGHT OF PARTICLES EVER COLLIDING
C      WDEL=DELAYED PARTICLE WEIGHT
C      WDELF=A DELAYED PARTICLE IS FOLLOWED IF WDEL EXCEEDS WDELF
C      WIR(IT,IR)=CUMULATIVE WEIGHT OF PARTICLES COLLIDING IN TIME BOX IT IN,
C           REGION IR
C      WISN(IT,ISS)=CUMULATIVE WEIGHT OF PARTICLES CROSSING SURFACE SEGMENT ISS
C           FROM REGION WITH HIGHER ID NUMBER TO REGION WITH LOWER ID NUMBER
C      WISP(IT,ISS)=CUMULATIVE WEIGHT OF PARTICLES CROSSING SURFACE SEGMENT ISS
C           FROM REGION WITH LOWER ID NUMBER TO REGION WITH HIGHER ID NUMBER
C      WIT(IT,N)=CUMULATIVE WEIGHT IN TIME BOX IT=1,2,...,NT FOR INTEGRAL TALLY
C           OF TYPE N=1,2,...,NIT (NIT.LE.10).  THESE TALLIES ARE COMPUTED BY
C           STATEMENTS INSERTED AT WILL INTO THE PROGRAM AND ARE PARTICULAR TO
C           THE GEOMETRY, APPLICATION, ETC.
C      WLC(IT)=CUMULATIVE WEIGHT OF PARTICLES LEAKING FROM SYSTEM IN TIME BOX IT
C      WS=CUMULATIVE WEIGHT OF SOURCE PARTICLES
C      WSPL=WEIGHT OF BANKED PARTICLE
C      WSPLT=PARTICLE WEIGHT MINIMUM FOR SPLITTING
C      WX1,WY1,WZ1=CUMULATIVE VALUES OF XCUM,YCUM,ZCUM AT COLLISION POINTS TIMES
C           WEIGHT BEFORE COLLISION
C      WX2,WY2,WZ2=CUMULATIVE VALUES OF XCUM**2,YCUM**2,ZCUM**2 AT COLLISION
C           POINTS TIMES WEIGHT BEFORE COLLISION
C      WX1S,WY1S,WZ1S=CUMULATIVE VALUES OF X,Y,Z AT SOURCE PARTICLE LOCATIONS
C      WX2S,WY2S,WZ2S= CUMULATIVE VALUES OF X**2,Y**2,Z**2 AT SOURCE LOCATIONS
C      X,Y,Z=RECTANGULAR COORDINATES OF PARTICLE IN PARENT CELL
C      XCELL,YCELL,ZCELL=CELL DIMENSIONS, NOT USED IF NCON=0
C      XCUM,YCUM,ZCUM=RECTANGULAR COORDINATES OF PARTICLE IN ARRAY
C      XSPL,YSPL,ZSPL=RECTANGULAR COORDINATES OF BANKED PARTICLE
C      X1,Y1,Z1=MEAN COLLISION COORDINATES
C      X2,Y2,Z2=MEAN SQUARE COLLISION COORDINATES
C           SUFFICES ON CUMULATIVE WEIGHTS AND MOMENTS REFER TO COLLISIONS IN
C           CERTAIN REGIONS ONLY
C      X1S,Y1S,Z1S=MEAN SOURCE COORDINATES
C      X2S,Y2S,Z2S=MEAN SQUARE SOURCE COORDINATES
C      RS,R2U,XYU,AS,BS,CS,ZCA=TEMPORARIES IN GEOMETRY CALCULATIONS
C
C  TIME BOXES
C      AN EVENT AT TIME T IS TALLIED IN TIME BOX IT IF TB(IT-1).LE.T.LT.TB(IT),
C      WHERE TB(IT) IS THE UPPER LIMIT OF TIME BOX IT.  TIME BOX IT=1 COVERS THE
C      INTERVAL (0.,OFFSET). TIME BOXES IT=2,ITB1 ARE OF WIDTH DELT1, AND TIME
C      BOXES IT=ITB1+1,NT ARE OF WIDTH DELT2
C
C  SURFACE DESCRIPTIONS
C      PLANE (1), A VECTOR (AS,BS,CS) IS NORMAL TO PLANE AND IS DIRECTED GENER-
C      ALLY OUT FROM ORIGIN.  LEAST DISTANCE FROM ORIGIN TO PLANE IS
C      DS/SQRT(AS**2+BS**2+CS**2).
C      SPHERE (2), RADIUS DS CENTERED AT (AS,BS,CS)
C      CIRCULAR CYLINDER (3), RADIUS GS WITH AXIS PASSING THRU (AS,BS,CS) IN
C      DIRECTION (DS,ES,FS)
C      ELLIPSOID (4), CENTERED AT (AS,BS,CS) WITH AXES PARALLEL TO X,Y,Z COOR-
C      DINATE AXES AND WITH RADII DS,ES,FS, RESPECTIVELY.
C      CONE(5), APEX AT (AS,BS,CS) WITH AXIS PARALLEL TO (DS,ES,FS) AND OPENING
C           ANGLE 2.*GS WHERE GS IS IN UNITS OF RADIANS
```

```
  505 D=1000000.
C     NAST IS THE NUMBER OF SURFACE SEGMENTS ADJACENT TO REGION IR
      NAST=NAS(IR)
      DO 500 ISS=1,NAST
C     IAST IS THE ID NO. OF THE SURFACE SEGMENT WITH ORDER NO. ISS ADJACENT TO
C     REGION IR
      IAST=IAS(IR,ISS)
C     IDST IS THE ID NO. OF THE SURFACE CONTAINING SURF SEG WITH ID NO. IAST
      IDST=IDS(IAST)
C     ITPT IS THE TYPE NO. OF SURFACE WITH ID NO. IDST
      ITPT=ITP(IDST)
      GO TO (501,502,503,504,506                              ),ITPT
C     PLANE SURFACE WITH NORMAL (AS,BS,CS) AND WITH CLOSEST DISTANCE TO
C     ORIGIN DS/SQRT(AS**2+BS**2+CS**2)
  501 A1=AS(IDST)*UX+BS(IDST)*UY+CS(IDST)*UZ
      IF (ABS(A1).LT.1.E-07) GO TO 500
      B1=DS(IDST)-AS(IDST)*X-BS(IDST)*Y-CS(IDST)*Z
      DIRY=B1/A1
      IF (DIRY.LT.0.0) GO TO 500
      NPRT=1
      GO TO 520
C     SPHERE OF RADIUS DS CENTERED AT (AS,BS,CS)
  502 A1=1.
      B1=(X-AS(IDST))*UX+(Y-BS(IDST))*UY+(Z-CS(IDST))*UZ
      C1=(X-AS(IDST))**2+(Y-BS(IDST))**2+(Z-CS(IDST))**2-DS(IDST)**2
      GO TO 510
C     CIRCULAR CYLINDER OF RADIUS GS WITH AXIS PASSING THRU (AS,BS,CS) IN
C          DIRECTION (DS,ES,FS).  NOTE (DS,ES,FS) WAS NORMALIZED IN INITIALIZE
  503 V2=UX*DS(IDST)+UY*ES(IDST)+UZ*FS(IDST)
      V3=(X-AS(IDST))*UX+(Y-BS(IDST))*UY+(Z-CS(IDST))*UZ
      V4=(X-AS(IDST))**2+(Y-BS(IDST))**2+(Z-CS(IDST))**2
      V5=(X-AS(IDST))*DS(IDST)+(Y-BS(IDST))*ES(IDST)+(Z-CS(IDST))*
     1FS(IDST)
      A1=1.-V2**2
      B1=V3-V5*V2
      C1=V4-V5**2-GS(IDST)**2
      GO TO 510
C     ELLIPSOID CENTERED AT (AS,BS,CS) WITH AXES PARALLEL TO X,Y,Z AXES WITH
C          RADII DS,ES,FS, RESPECTIVELY.  THEN IN INITIALIZE DS,ES,FS ARE
C          REPLACED BY THEIR RECIPROCALS SQUARED
  504 A1=UX**2*DS(IDST)+UY**2*ES(IDST)+UZ**2*FS(IDST)
      B1=UX*(X-AS(IDST))*DS(IDST)+UY*(Y-BS(IDST))*ES(IDST)
     1+UZ*(Z-CS(IDST))*FS(IDST)
      C1=(X-AS(IDST))**2*DS(IDST)+(Y-BS(IDST))**2*ES(IDST)
     1+(Z-CS(IDST))**2*FS(IDST)
      GO TO 510
C     CONE WITH APEX AT (AS,BS,CS) AND AXIS PARALLEL TO (DS,ES,FS) WITH
C          OPENING ANGLE 2.*GS.  NOTE (DS,ES,FS) WAS NORMALIZED AND (COSGS)**2
C          WAS STORED IN GS IN INITIALIZE
  506 V2=UX*DS(IDST)+UY*ES(IDST)+UZ*FS(IDST)
      V3=(X-AS(IDST))*UX+(Y-BS(IDST))*UY+(Z-CS(IDST))*UZ
      V4=(X-AS(IDST))**2+(Y-BS(IDST))**2+(Z-CS(IDST))**2
      V5=(X-AS(IDST))*DS(IDST)+(Y-BS(IDST))*ES(IDST)+(Z-CS(IDST))*
     1FS(IDST)
      A1=V2**2-GS(IDST)
      B1=V2*V5-GS(IDST)*V3
      C1=V5**2-GS(IDST)*V4
  510 B2=B1**2-A1*C1
C     ARE THE DISTANCE ROOTS COMPLEX
      IF (B2.LT.0.0) GO TO 500
      B3=SQRT(B2)
      IF (ABS(A1).LT.1.E-20) GO TO 500
      D1(IAST)=(-B1+B3)/A1
C     ARE THE DISTANCE ROOTS BOTH NEGATIVE
      IF (D1(IAST).LT.0.0) GO TO 500
      D2(IAST)=(-B1-B3)/A1
      DIRY=D2(IAST)
      NPRT=2
      IF (DIRY.LT.0.0) GO TO 519
```

APPENDIX B. (Continued)

```
      GO TO 520
  519 DTRY=D1(IAST)
      NPRT=1
  520 IF (DTRY.GT.D) GO TO 500
C
C     IF THIS POINT IS REACHED THEN WE HAVE A CANTIDATE DTRY FOR THE SHORTEST
C     FLIGHT TO A SURFACE SEGMENT ADJACENT TO REGION IR. NOW TEST SENSES TO
C     FIND IF R+DTRY*U IS ACTUALLY IN THE SURFACE SEGMENT WITH ID IAST.
      IF (NSEN(IAST).EQ.0) GO TO 550
      X1=X+DTRY*UX
      Y1=Y+DTRY*UY
      Z1=Z+DTRY*UZ
      NSENT=NSEN(IAST)
      DO 530 IS1=1,NSENT
      IDENT=IDEN(IAST,IS1)
      ITPT=ITP(IDENT)
      GO TO (531,532,533,534                                    ) ITPT
  531 SENSE=AS(IDENT)*X1+BS(IDENT)*Y1+CS(IDENT)*Z1-DS(IDENT)
      GO TO 540
  532 SENSE=(X1-AS(IDENT))**2+(Y1-BS(IDENT))**2+(Z1-CS(IDENT))**2
     1-DS(IDENT)**2
      GO TO 540
  533 V1=DS(IDENT)**2+ES(IDENT)**2+FS(IDENT)**2
      V11=1./V1
      SENSE=(X1-AS(IDENT))**2*(1.-DS(IDENT)**2*V11)
     1+(Y1-BS(IDENT))**2*(1.-ES(IDENT)**2*V11)
     2+(Z1-CS(IDENT))**2*(1.-FS(IDENT)**2*V11)-GS(IDENT)**2
      GO TO 540
  534 SENSE=(X1-AS(IDENT))**2/DS(IDENT)**2+(Y1-BS(IDENT))**2/
     1ES(IDENT)**2+(Z1-CS(IDENT))**2/FS(IDENT)**2-1.
  540 JSEN=1
      IF (SENSE.LT.0.0) JSEN=-1
      IF (ISEN(IAST,IS1)*JSEN.EQ.0) GO TO 541
C     THIS SENSE TEST HAS BEEN PASSED
      GO TO 530
C     THIS SENSE TEST HAS BEEN FAILED
  541 IF (NPRT.EQ.1) GO TO 500
      DTRY=D1(IAST)
      NPRT=1
      GO TO 520
  550 CONTINUE
C     ALL TESTS HAVE BEEN PASSED SO IHIT IS THE ID NUMBER OF THE CLOSEST STRUCK
C     SURFACE SEGMENT OF THOSE YET TESTED
  552 IHIT=IAST
      ISST=ISS
      D=DTRY
  500 CONTINUE
```

APPENDIX C. REGIONS, SURFACES AND SURFACE SEGMENTS IN MONTE CARLO COMPUTATIONS

We wish to use such concepts as region, surface segment, etc., without more precise definition than geometrical intuition requires. By the neighborhood of a point $(x_0,y_0,\cdots)$ in real Euclidean space, we mean the set of all points $(x,y,\ldots)$ in the space such that $\sqrt{(x - x_0)^2 + (y - y_0)^2 + \cdots} < \varepsilon$, where ε is some positive number. The point $(x_0,y_0,\cdots)$ is a limit point of a set of points S in the space if every neighborhood of $(x_0,y_0,\cdots)$ contains at least one point of S. If the set contains all its limit points it is a closed set. A limit point of a set is an interior point of the set if there exists a neighborhood of the limit point which consists entirely of points of the set. A limit point of the set which is not an interior point is a boundary point of the set. A set which consists entirely of interior points is said to be an open set.

A straight line segment is the set of points with coordinates $x = a_x s + b_x$, $y = a_y s + b_y$, $\cdots$, where the real variable, s, is in the range $(s_1 \leq s \leq s_2)$. We define a multiple segment as an ordered set of straight-line segments $S_1,S_2,S_3,\cdots$ of which segment S_n and segment S_{n+1} have common end points, $n = 1,2,\cdots$. An S-connected set of points in space is such that every pair of points in the set can be joined by a multiple segment. A region is an open, S-connected set of points in space.

We take a polynomial surface (P-surface) of order N to be a set of points satisfying a relation $f_N(x,y,\cdots) = 0$, where f_N is a polynomial of order N. Each point $(x,y,\cdots)$ in space has a unique sense 1 or -1 with respect to a particular P-surface of order N according as $f_N(x,y,\cdots)$ is non-negative or is negative. A P-surface segment is a non-empty set of points all of which lie in a particular P-surface and which have assigned senses with respect to other P-surfaces. Because the latter P-surfaces may not intersect the first one it follows that a P-surface is itself a P-surface segment. A P-surface segment may be open or closed.

The closure of a region is its surface, and we shall restrict ourselves henceforth to regions, P-regions, such that their surfaces are joins of disjoint (except possibly at their boundaries) P-surface segments, called PR-surface segments. We assume an analogue of the Jordan theorem that any set of P-surfaces divides all space into disjoint fundamental P-regions. Any P-region is the join of one or more fundamental P-regions together with the P-surface segments separating them (see Figure C-1).

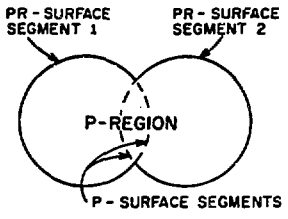

Fig. C-1. Illustration of Surface Segments.

A straight line segment which intersects one PR-surface segment once has its end points in two district and uniquely defined P-regions. Thus if in a Monte Carlo transport calculation we have determined that a particle trajectory hits a certain PR-surface segment then we know the unique P-region entered on the other side. For example, in penetrating PR-surface segment 2 in Figure C-2, the particle trajectory A goes from P-region I into P-region II.

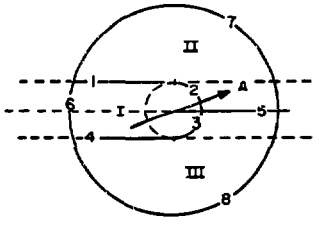

Fig. C-2. Illustration of P-Regions (Roman Numerals), Surface Segments (Arabic Numerals), and Trajectories.

In a Monte Carlo transport calculation there is a hierarchy of tests which establish the PR-surface segment first intersected by a particle trajectory. First, given the set of PR-surface segments adjacent to the P-region of trajectory origin (as PR-surface segments 1, 2, 3, 4, and 6 are adjacent to P-region I in Figure C-2), one computes distance to each of the P-surfaces containing these PR-surface segments. Complex and negative distances indicate misses and these are not considered further. Beginning with the least, real, positive distance we test to see if each point of intersection with a P-surface

satisfies the sense relations which establish that an actual PR-surface segment has been hit. In the example of Figure C-2,trajectory A intersects two P-surface segments which are not PR-surface segments before it hits PR-surface segment 2.

APPENDIX D. SAMPLE INPUT FOR ANDY1G2

```
POINT SOURCE NEAR TWO INTERSECTING SPHERES
 4 5100      1       0       0       0
   100      10
.?              .2              .?              1000.           1.
?.              .0001           .001            50.             10.
        ?       ?       5       2       0
        ?
0.              0.              0.              100.            0.              0.
0.              0.
        ?
50.             0.              0.              20.             0.              0.
0.              0.
        ?
50.             10.             0.              15.             0.              0.
0.              0.
        1       0
        ?       1
        ?
        1
        ?       1
        ?
        1
        ?       1
        1       ?       ?
        4       ?       ?
        2       ?
        ?       ?
        1       1
.0?             .9              .0
10.             .5              .0
```

APPENDIX E. SAMPLE INPUT FOR ANDYIR2

```
TRX  5/1  GROUP 3  SOURCE UNIFORM AND ISOTROPIC IN COOLANT
  4000     59      0       0       0
     3     2
1.0      E+501.0      E+50 0.0          1.0      E+501.0      E+50
1000.         .0005          1000.          1.             4.
1.806         3.128          2.
     6     9     9      2     1
     3
0.             0.             0.             0.             0.             1.
.575306        0.
     3
-.903          1.564          0.             0.             0.             1.
.575306        0.
     3
.903           1.564          0.             0.             0.             1.
.575306        0.
     3
.903           -1.564         0.             0.             0.             1.
.575306        0.
     3
-.903          -1.564         0.             0.             0.             1.
.575306        0.
     1
0.             1.             0.             1.564          0.             0.
0.             0.
     1
1.             0.             0.             .903           0.             0.
0.             0.
     1
0.             -1.            0.             1.564          0.             0.
0.             0.
     1
-1.            0.             0.             .903           0.             0.
0.             0.
     1    0
     2    0
     3    0
     4    0
     5    0
     6    0
     7    0
     8    0
     9    0
     9    1
     1    2      3      4      5      6      7      8      9
     2    3      4      5      6     #1     #1     #1     #1
     1    2
     1
     1
     3    2
     2    6      9
     1   -6     -4
     3    2
     3    6      7
     1   -5     -3
     3    2
     4    7      8
     1   -6     -4
     3    2
     5    8      9
     1   -3     -5
.56           .77            .0
.41           .87            .0
```

APPENDIX F. PART OF OUTPUT FOR SAMPLE ANDY1G2 PROBLEM*

```
      RESULTS FOR TIME INTERVAL      5

      WTR(IT,IR),IR=1,NREG
 .723124E+04       .445461E+03
      WTSP(IT,ISS),ISS=1,NSEG
 .                 .241687E+03     0.
      WTS(IT,ISS),ISS=1,NSEG
 .                 .433249E+02     0.
      WLC(IT)
 .

      RESULTS FOR TIME INTERVAL      6

      WTR(IT,IR),IR=1,NREG
 .699300E+04       .243761E+03
      WTSP(IT,ISS),ISS=1,NSEG
 .                 .172653E+03      .900000E+00
      WTS(IT,ISS),ISS=1,NSEG
 .                 .242404E+02      .225000E+00
      WLC(IT)
 .

      RESULTS FOR TIME INTERVAL      7

      WTR(IT,IR),IR=1,NREG
 .677426E+04       .253838E+03
      WTSP(IT,ISS),ISS=1,NSEG
 .                 .141624E+03      .660510E+01
      WTS(IT,ISS),ISS=1,NSEG
 .                 .213362E+02      .347737E+00
      WLC(IT)
 .
```

*In addition, input, time box boundaries, and special tallies are listed. Running (CP) time 96 sec for 40,000 starters and 167,000 collisions.

APPENDIX G. PART OF OUTPUT FOR SAMPLE ANDY1R2 PROBLEM*

WCUM

.199478E+05

WX1 ,WY1 ,WZ1 ,WX2 ,WY2 ,WZ2

-.105458E+04 .648419E+03 -.475754E+03 .265131E+06 .275221E+06 .253554E+06

WX1C,WY1C,WZ1C,WX2C,WY2C,WZ2C

-.127691E+04 .686386E+03 -.434714E+03 .183325E+06 .190851E+06 .175848E+06

WX1R,WY1R,WZ1R,WX2R,WY2R,WZ2R

-.377769E+03 -.379677E+02 -.410405E+02 .818062E+05 .843701E+05 .777066E+05

WS

.400000E+04

WX1S,WY1S,WZ1S,WX2S,WY2S,WZ2S

-.902823E+02 .11.805E+03 0. .112371E+04 .297877E+04 0.

X1S,Y1S,Z1S,X2S,Y2S,Z2S

-.215700E-01 .277011E-01 0. .280928E+00 .744692E+00 0.

X1 ,Y1 ,Z1 ,X2 ,Y2 ,Z2

-.829457E-01 .325059E-01 -.238500E-01 .132913E+02 .137971E+02 .127109E+02

X1C,Y1C,Z1C,X2C,Y2C,Z2C

-.909756E-01 .488528E-01 -.304402E-01 .130479E+02 .135836E+02 .125158E+02

X1R,Y1R,Z1R,X2R,Y2R,Z2R

-.640541E-01 -.643777E-02 -.695879E-02 .138710E+02 .143057E+02 .131759E+02

VARX ,VARY ,VARZ ,ANIZX ,ANIZY ,WCUM

.130077E+02 .130521E+02 .127109E+02 .977184E+00 .973857E+00 .199478E+05

VARXC,VARYC,VARZC,ANIZXC,ANIZYC,WIR(2,1)

.127640E+02 .128377E+02 .125158E+02 .980550E+00 .974920E+00 .140501E+05

VARXR,VARYR,VARZR,ANIZXR,ANIZYR,WCUMR

.135862E+02 .135629E+02 .131759E+02 .969653E+00 .971461E+00 .589765E+04

NS	ISPLC	ISPLO	NHS	NCOL	NDEL
4000	0	0	4000	137311	0

*In addition, input, time box boundaries and tallies, and other special tallies are listed. Running (CP) time 277 seconds.