DETERMINATION OF THE GEOMETRICAL FACTOR AND OF THE SPATIAL DISTRIBUTIONS OF PARTICLES FOR A COSMIC RAY TELESCOPE USING A MONTE CARLO APPROACH

MICHAEL HEMMER
CAROL JO CRANNELL

JULY 1973

GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND
DETERMINATION OF THE GEOMETRICAL FACTOR AND OF THE SPATIAL DISTRIBUTIONS OF PARTICLES FOR A COSMIC RAY TELESCOPE USING A MONTE CARLO APPROACH

Michael Hemmer*
NASA/Goddard Space Flight Center
Greenbelt, Maryland 20771

and

Carol Jo Crannell
Federal City College
Washington, D.C. 20001

*Co-Op student trainee, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061
INTRODUCTION

The program presented here uses a Monte Carlo technique to compute the geometrical factor of a detector and the spatial and angular distributions of simulated particles at designated layers within the detector. As the program is written, each layer is assumed to be rectangular, parallel to every other layer with its edges parallel to the edges of every other layer, and centered about a common axis. See Figure 1. In a previous publication, the physical significance of the geometrical factor, and of the effective geometrical factor as a function of path length, is described.

The distributions are computed by accumulating data from a large number of individual, simulated events. Each simulated particle is assigned independent, random spatial coordinates \((X_0, Y_0)\) in the plane of the uppermost layer of the detector and independent, random trajectory angles \((\Theta, \Phi)\). A pictorial representation of a simulated event in a hypothetical detector configuration is shown in Figure 2.

This program is written in Fortran IV for the IBM 360 computers and is currently being run under the Fortran IV (H) compiler. Some features such as the subroutine ERRSET, RANDU, and REMTIM and the type declarations REAL *8 and LOGICAL *1 may need to be changed for use with a different computer system.

The operation of the program is described in detail in the next section. In the third section the use of input data cards and a sample
data set are presented. The symbol definition list is given in Appendix A. Card images of the Fortran object deck and the graphing subroutine are listed in Appendix B. A flow diagram of the program is presented in Appendix C.

DESCRIPTION OF THE PROGRAM

The program begins by finding the CPU and I/O times used in the computations for the previous detector configuration by a call to the subroutine REMTIM. Then the initial values for a new data set are read. A data set consists of the numerical specifications of a particular detector configuration for NPAR number of simulated particles. Next, the variables NPLANE and TIME are tested. The program is terminated if NPLANE is set equal to 99 on the input data card. If insufficient CPU and I/O time remain, TIME = 1 and the program is terminated. If execution continues, the remainder of the input data cards for the present configuration are read and the internal variables are initialized. The longest possible path length within the detector, LPPTH, is determined for later reference.

In the main loop of the program, the trajectory of a simulated particle is generated and tested at successively deeper layers until the trajectory passes outside the detector. At the start of the loop, after every 3000th simulated particle, the remaining CPU and I/O times are again determined. If sufficient time does not remain for both an additional 3000 events and the output summary for the detector configuration, the program branches to the final computations, resetting.
NPAR to the number of events presently accumulated. If sufficient
time does remain, TOPT and TOPW are set equal to the dimensions of
the uppermost or 0TH layer of the detector, T0 and W0. The coordi-
nates of a new simulated particle in the uppermost layer of the
detector and the zenith and azimithal trajectory angles are chosen
by four separate calls to subroutine RANDU.² An isotropic distri-
bution of incident flux is simulated by the algorithms in the
program as it is presented in the next section. The change, indicated
in the comment cards will generate a $\cos^2(\theta)$ distribution of
incident flux. Advantage is taken of the four-fold symmetry in
azimuth angle, by varying $\phi$ over the range 0 to $\pi/2$ rather than a
full $2\pi$ radians.

After arrays for the graphing output are incremented, the next
major, nested loop of the program is entered. The detector configu-
ration is treated as a series of boxes with rectangular tops and
bottoms and trapizoidal sides. These boxes, or levels, are numbered
by the DO variable K the same as the number of the bottom layer or
plane. The surface through which the particle exits the detector
level is determined. If the particle has exited the detector, data
for that particle are stored, and the program returns to generate a
new simulated particle. If the particle has passed through the bottom
of the present detector level into the next level, the parameters are
set to the values for the next detector level and the loop is repeated.
In the last detector level ($K = NPLANE$) the path length of the trajectory
within that level is computed. The distribution of path lengths and
the effective geometrical factor as a function of path length are output at the completion of the data set.

After NPAR simulated particles have been generated and their trajectories through the detector levels determined, the output summary for the detector configuration is prepared. Path length data are computed and written. If the detector has the form of a rectangular parallelepiped, the geometrical factor is computed analytically and written. Finally the distributions of simulated particles by path lengths, by spatial coordinates at specified detector layers, and by zenith angle at specified detector layers are printed numerically and as printer plots through the use of the graphing subroutine. Following the output execution, the program returns to the beginning.

DESCRIPTION OF THE INPUT DATA

One data set, including a heading, the detector dimensions, and other numerical parameters, is specified for each detector configuration. All dimensions are in centimeters. The data cards to be read for each data set are as follows.

Card 1 a heading. Any combination of 80 characters and blanks on the first card will be read and written out at the start of the output.

Card 2 detector and particle data. The input data are read in the following order: the number of detector layers after the first, NPLANE, in format I2; the X dimension of the 0th plane, T0, in format F10.4; the Y dimension of the 0th plane, W0, in format F10.4; the number of simulated particles to be generated, NPAR, in format
I10; the initial value for the random integer, IY, in format I5; and the numbers of those detector layers for which distribution data are required, up to 10 plane numbers in ascending order, in format 10I2.

Card 3 and subsequent cards up to a total of NPLANE +2. One card is prepared for each detector layer after the topmost, giving in order the X dimension of the layer, T(*); the Y dimension of the layer, W(*); and the distance along the Z axis between the present and the previous layers, H(*), in format 3F10.4.

Note that after the data set for the last detector configuration, two more data cards are needed. The first card, a fake heading, is followed by another card with the digits 99 in the first two columns. This sets NPLANE to 99 which terminates the program. A sample data set follows:

```
THIS DETECTOR IS A THREE-ELEMENT COSMIC RAY TELESCOPE
2  94.6  94.6  100000  37 1 2
 76.5  76.5  18.2
 76.5  76.5  57.3
THIS IS A FAKE HEADING CARD
99
```

This data set specifies a detector configuration with three layers. The top layer is 94.6 x 94.6 cm$^2$. A total of 100000 simulated particles will be generated using the random number generator initialized with the integer, 37. Spatial and angular distribution data will be generated for the first and second layers as well as for the topmost. The first layer following the top has dimensions 76.5 x 76.5 cm$^2$ and is spaced 18.2 cm below the top. The second layer has dimensions 76.5 x 76.5 cm$^2$ and is spaced 57.3 cm below the first. The program is terminated after finishing the output summary for this detector configuration.
ACKNOWLEDGMENTS

The authors wish to thank Mr. Norman Grabowski for helpful suggestions and assistance with this program and Dr. Jonathan Ormes for his interest and constructive comments.

REFERENCES


HYPOTHETICAL DETECTOR CONFIGURATION

FIGURE 1
SIMULATED PARTICLE TRAJECTORY

W(NPLANE)

H(NPLANE)

X(NPLANE-1), Y(NPLANE-1)

W(NPLANE-1)

T(NPLANE)

T(NPLANE-1)

XO, YO

WO

TO

Y

Z

θ
A. SYMBOL DEFINITION LIST
AND(30) | int| the number of particles that pass through the last plane of 
the detector divided into 30 three-degree bins in THETA.

ANGDIS(4,4,30,11) | int| the distribution of particles by position and by 
age with respect to the Z axis at 11 different planes. The 
subscripts represent: the displacement from the X axis divided into 
4 equal bins; the displacement from the Y axis divided into 4 equal 
bins; the angle made with the Z axis, THETA, divided into 30 three-
degree bins; the Nth plane number given by GRAPH(N), N=1 is always 
the 0th plane.

ANGD1(4) | int| every other 9-degree bin, 3 three-degree bins in ANGDIS, 
starting at 9 degrees. After each set of its values are written out, 
it is filled with the next bin corresponding to the 2nd subscript of 
ANGDIS, rotating 1, 2, 3, 4.

ANGD2(4) | int| the same as ANGD1 except it represents the alternate 
9-degree bin starting at 18 degrees.

ANGLE | reall| set to every integer multiple of 3., up to 90., for reference 
in output.

ANGT | int| used as the 3rd subscript to ANGDIS to divide the angle made 
with the Z axis, THETA, into 30 three-degree bins.

ANGX | int| used as the 1st subscript to ANGDIS to divide a particle's 
displacement from the X axis, plus or minus, into 4 equal bins.

ANGY | int| used as the 2nd subscript to ANGDIS to divide a particle's 
displacement from the Y axis, plus or minus, into 4 equal bins.

AVEPTH | reall| the average path length of all particles that enter the last 
detector level.

CGF | reall| the geometrical factor which can be computed analytically for 
the special case that the detector shape is a rectangular 
parallelepiped.

CPhi | reall| cos(PHI)

CPSD | reall| the particle counts per square degree.

CTHETA | reall| cos(THETA)

DEG1 | reall| set to every other multiple of 9., starting at 9., up to 81., 
for output reference.

DEG2 | reall| set to every other multiple of 9., starting at 18., up to 90., 
for output reference.
EFFGF |real| the effective geometrical factor for path lengths greater than \( J/100 \times \bar{L}PPTH \), \( J=1,100 \).

ERRSET an IBM system subroutine that suppresses the output which would otherwise be printed for error number IHC210. It is possible to receive this error when calculating PTHLNG in PATHI or PATH2. It has no significance in this program because the standard fixup sets the result to zero, which is ignored. See "IBM System/360 Operating System, FORTRAN IV (G & H) Programmers' Guide", (GC28-6217-2), pp. 124-125.

EXITPT(99,3) |int| gives the plane through which the particle leaves the detector. The subscripts indicate the detector level and which surface within the level is the exit plane of the particle. The subscripts represent: the detector level where the particle exits; and the plane within the level that the particle exits from as follows:

1. if this is the last detector level and the particle passes through the bottom detector layer. If this is any other level, the particle does not exit the detector by passing through the lower layer.
2. the particle pass out of either side between the detector layers.
3. the particle passes through the front or back between the detector layers.

G a subroutine that will do printer plots of up to 100 points. The arguments that must be passed to it are as follows:

1. TYPE |int| with a value greater than 0 as follows,
2. a linear plot.
3. a semi-log plot in X, and a linear plot in Y.
4. a semi-log plot in Y, and a linear plot in X.
5. a log-log plot.
6. (5 or greater) the subroutine will decide what kind of plot to make. 5 is used in all calls in this program.

1. HEAD(5) |real*8| should contain the heading for the plot.
2. A(N) |real|, \( N \) between 1 and 100, contains the X coordinates of the data.
3. B(N) |real|, \( N \) between 1 and 100, contains the Y coordinates of the data.
4. SIZE |int| contains the dimensions of \( A \) and \( B \).

1. GAND(30) |real| a variable that transmits the values of AND to subroutine G.
2. GANG(30) |real| a variable that transmits the values of ANGLE to subroutine G.
3. GCPS(30) |real| a variable that transmits the values of CPSD to subroutine G.
4. GEFF(100) |real| a variable that transmits the values of EFFGF to subroutine G.
GF |real| the geometrical factor of this detector for particles passing through every detector layer.

GPBI(100) |real| a variable that transmits the values of PBIN to subroutine G.

GPOL(30) |real| a variable that transmits the values of POLANG to subroutine G.

GRAPH(11) |int| contains the plane numbers for the last subscripts of ANGDIS, XDIS, and YDIS. GRAPH(1) always has the value of 0, GRAPH must be filled left to right in ascending order.

GTAN(30) |real| a variable that transmits the value of TANG to subroutine G.

GTOT(30) |real| a variable that transmits the value of TOTC to subroutine G.

GXB(50) |real| a variable that transmits the values of XB to subroutine G.

GXDI(50) |real| a variable that transmits the values of XDIS to subroutine G.

GYB(50) |real| a variable that transmits the values of YB to subroutine G.

GYDI(50) |real| a variable that transmits the values of YDIS to subroutine G.

H(99) |real| the Z dimension of the Nth detector level.

HEAD(20) |real| reads and writes the data set heading.

I0 |real| is the total height of the detector in the Z dimension in the case that the detector has shape of a rectangular parallelepiped.

I |int| a DO variable.

ICPU |int| the first argument to REMTIM, it returns the seconds of CPU time remaining.

ICPUS |int| saves the previous value of ICPU.

I10 |int| the 2nd argument to REMTIM, it returns the seconds of I/O time remaining.

I1OS |int| saves the previous value of I10.

IX |int| is the 1st argument to REMTIM, it should contain the previous IY.
IY | int| the 2nd argument to RANDU, it is a random integer that is returned by RANDU.

I1 | int| a DO variable.

I2 | int| initial value for I1 in a DO loop.

I3 | int| the maximum value for I1 in a DO loop.

J | int| a DO variable.

JS | int| saves the current value of J for later use.

K | int| a DO variable.

KS | int| saves the current value of K for later use.

L | int| a DO variable.

LAPTH | real| the longest actual path length of any particle that enters the last detector level.

LP | real| used to calculate LAPTH.

LPPTH | real| the longest possible path length in the last detector level.

LS | int| saves the current value of L for later use.

M | int| a DO variable.

N | int| a DO variable.

NPAR | int| the number of particles to be simulated in this data set.

NPLANE | int| the number of planes after the 0th. NPLANE=99 stops the program.

PATH1 | real| a statement function subprogram which is used to calculate PTHLNG.

PATH2 | real| a statement function subprogram which is used to calculate PTHLNG.

PBIN | real| the longest possible path length, LPPTH, divided into 100 equal bins.

PHI | real| the azimuthal angle.

POLAR | int| set to even multiples of 3, from 3, up to 90, for output
POS [int] used to determine a particle's position with respect to the edges of the bottom of the present detector level:
(1) inside the edges of the plane,
(2) to the right or left of the plane,
(3) to the front or back of the plane,
(4) to both the front or back and right or left of the plane, that is, off a corner.

PTH.LN.G [real] the length of the path of a particle in the last detector level.

QC [real] an internal variable for calculating X and Y.

QI [int] the subscript to AND. It divides THETA into 30 three-degree bins.

QPTH [real] an internal variable used to calculate PTHLNG.

QPTH1 [real] an internal variable used to calculate PTHLNG.

QPTH11 [real] an internal variable used to calculate PTHLNG.

QPTH2 [real] an internal variable used to calculate PTHLNG.

QPTH22 [real] an internal variable used to calculate PTHLNG.

QS [real] an internal variable used to accumulate the values of AND in order to suppress zeros.

QX [int] used as the 1st subscript to XDIS, it divides TO into 50 equal bins.

QY [int] used as the 1st subscript to YDIS, it divides W0 into 50 equal bins.

RANDU a subroutine which is used to generate random numbers. The arguments are as follows:
IX [int] should be set to the previous IY, RANDU uses IX to generate another IY and YFL.
IY [int] is the random integer returned. It is read into the program as input data. Its initial value is any odd integer of 9 or less digits.
YFL [real] is a random number between 0.0 and 1.0 which is used to generate particle parameters.

REM [int] used to count 3000 particles between calls to REMTIN.

SIZE [int] the last argument to G which contains the dimension of GAND and GTAN.
SPIII | real | sin(PIII)
SQX | real | an internal variable used to calculate CGF.
SQY | real | an internal variable used to calculate CGF.
STHETA | real | sin(THETA)

SUBH1(5) | real*8 | contains the heading for the 1st call to G, "EFFGF VS. PATH-LNGTH".
SUBH2(5) | real*8 | contains the heading for the 2nd call to G, "TOTAL COUNTS VS. PATH-LENGTH".
SUBH3(5) | real*8 | contains the heading for the 3rd call to G, "COUNTS OUT BOTTOM VS. POLAR ANGLE".
SUBH4(5) | real*8 | contains the heading for the 4th call to G, "CPSD VS. POLAR ANGLE".
SUBH5(5) | real*8 | contains the heading for the 5th call to G, "NO. OF PARTICLES VS. XBIN".
SUBH6(5) | real*8 | contains the heading for the 6th call to G, "NO. OF PARTICLES VS. YBIN".
SUBH7(5) | real*8 | contains the heading for the 7th call to G, "TOTAL PARTICLES VS. ANGLE".

SUMC(100) | lint | the number of particles which have a path length in the last detector of N/100.*LPPTH or greater.
T(99) | real | the X dimension of the Nth plane.
TANG | lint | the total number of particles that are within a specific 3-degree angular bin at a given plane.
TGR(11) | real | the X dimension of the Nth plane in GRAPH(N).
THETA | real | the zenith angle of a simulated particle trajectory.
TIME | lint | a flag, when set to one, the program stops.
TOPT | real | the X dimension of the top of the detector level that the particle is presently passing through.
TOPW | real | the Y dimension of the top of the detector level that the particle is presently passing through.
TOPX |real| the X coordinate of a particle at the top to the present detector level.

TOPY |real| the Y coordinate of a particle at the top of the present detector level.

TOTC(4,100) |int| used to determine the path length and exit place of the particles from the detector. The subscripts represent: the exit plane,
(1) bottom of the detector,
(2) side of the detector,
(3) front or back of the detector,
(4) the total of 1, 2, and 3; path length divided into 100 equal bins.

TOTPTH |real| the sum of all of the path lengths of all the particles that enter the last detector level.

TTANG |int| the total of all of the particles in a given plane. TTANG is used to test against ZERO in order to suppress zeros.

TO |real| the X dimension of the 0th plane.

W(99) |real| the Y dimension of the Nth plane.

WGR(11) |real| the Y dimension of the Nth plane in GRAPH(N).

WICPUS |real| used to write out, in minutes, the CPU time required by the previous data set.

WIOOS |real| used to write out, in minutes, the I/O time required by the previous data set.

WLPPTH |real| used to write out the longest possible path length.

W0 |real| the Y dimension of the 0th plane.

X(99) |real| the coordinate of a particle in the X dimension at the Nth plane.

XB |real| set to multiples of the X dimension of a plane divided by 100, for output reference.

XDIS(50,11) |int| the distribution of particles along the X axis, that is in strips parallel to the Y axis, the subscripts represent: the displacement from the X axis, plus or minus, divided into 50 equal bins; the Nth plane in GRAPH(N).

X0 |real| the X coordinate of a particle at the 0th plane.
Y(99) is the coordinate of a particle in the Y dimension at the Nth plane.

YB is set to a multiple of the X dimension of a plane divided by 100, for output reference.

YDIS(50,11) is the distribution of particles along the Y axis, that is in strips parallel to the X axis, the subscripts represent: the displacement from the Y axis, plus or minus, divided into 50 equal bins; the Nth plane in GRAPH(N).

YFL is an argument to RANDU, it is returned with a random real number between 0.0 and 1.0.

YO is the Y coordinate of a random particle at the 0th plane.

ZERO is used to find the number of particles at each plane in GRAPH(N) and to suppress zeros in the output.
B. LISTING OF THE
GEOMETRICAL FACTOR PROGRAM
AND
GRAPHING SUBROUTINE
REAL T(99), W(99), H(99), X(99), Y(99), HFAD(20), TGR(11), WGR(11),
AGPBI(100), GFF(100), GPOI(30), GCP(30), GAND(30), GTOT(100), GTAN(30),
BGANG(50), GXR(50), GYR(50), GXDI(50), GYDI(50), LAPTH, LPPTH, LP
INTEGR EXTPT(99, 3), PO, AND(30), SUMC(100), TOTC(4, 100), OX, OY,
AXDIS(50, 11), YDIS(50, 11), GRA(11), ANG, ANG, ANG, ANG, ANGGDIS(4, 4, 30, 11)
R, ZERO, TANG, ANG(4), ANG(4), TANG, TIME, POS, OI RFIM, SIZE

COMMENT (SUBH1-7) CONTAIN THE HEADINGS FOR TRANSMISSION TO SUBROUTINE
C (6) THEY ARE REAL*8 FOR CONVENIENCE ONLY.
ASURH7(5)
DATA SUBH1, SUBH2, SUBH3, SUBH4, SUBH5, SUBH6, SUBH7/'** FF ** F VS', PTHLN
AG', 3*' ', 'TOTAL CO', 'INTS VS', 'PATH-LE', 'NGTH', 'R', 'COUNTS', 'UT DOTT', 'M VS. PO', 'LAR ANG', 'F', 'CPSD
E', 'TOTAL PA', 'RTICLES', 'VS. ANG', 'F', 'F /

COMMENT (PATH1, PATH2) ARE STATEMENT FUNCTION SUBPROGRAMS WHICH
C CALCULATE A COEFFICIENT FOR THE PATH LENGTH OF THE PARTICLES.
PATH2(A, B, C, D) = (2.*A + B)/(C - C + 2.*(A - D))

COMMENT (TIME1) IS A FLAG, WHEN SET TO ONE THE PROGRAM HALTS
C TIM1 = 0

COMMENT THIS DO ALLOWS THE PROGRAM TO OPERATE ON AS MANY AS 100000
C DATA SETS.
DO 10 I = 1, 100000

COMMENT (RFMTIM) IS AN IRM SUBROUTINE THAT RETURNS THE SECONDS
C REMAINING OF CPU AND I/O TIME AS INTEGERS.
CALL RFMTIM(IPCPU, 110)
IF (.F.0) GO TO 101

COMMENT THIS DETERMINES THE TIME REQUIRED FOR THE PREVIOUS DATA SET
C AND WRITES THIS OUT AS MINUTES OF CPU AND I/O TIME.
WIPUS = (ICPUS - IPCPU)/60.
WIOS = (I1OS - I10)/60.
WRITE (6, 1000) WIPUS, WIOS
101 IPCPU = IPCPU
I1OS = I10

COMMENT SET (ANGDIS, GRAPH) TO ZERO.
DO 11 J = 1, 11
DO 12 K = 1, 30
DO 12 L = 1, 4
DO 12 M = 1, 4
12 ANGDIS(N, L, K, J) = 0
11 GRAPH(J) = 0

COMMENT READ THE DATA SET HEADING, THE NUMBER OF PLANES AFTER THE 0TH,
C LENGTH OF THE 0TH PLANE IN THE X DIMENSION, THE LENGTH OF THE
C 0TH PLANE IN THE Y DIMENSION, THE NUMBER OF PARTICLES, THE
C INITIAL (1Y), THE PLANES FOR WHICH DISTRIBUTION DATA IS
C REQUESTED IN ASCENDING ORDER. THE FORMAT IS, (20A4/12, 2F10.4,
C 110, 15, 1012).
READ(5,1001) HEAD, NPLANE, T0, W0, NPAR, IY, GRAPH(J), J=2,11

COMMENT TEST TO SEE IF THERE ARE NO MORE DATA SETS, IF (NPLANE)=99,
C OR IF (TIME) HAS BEEN SET TO ONE. IF SO BRANCH TO THE STOP
C STATEMENT.
If((NPLANE.EQ.99).OR.(TIME.EQ.1)) GO TO 102
WRITE(6,1002) I, HEAD, NPLANE, NPAR, IY, T0, W0

COMMENT INITIALIZE (QS, LAPTH, HO, XDIS, YDIS, TOPT, EXITPT, GAND, GANG, GCPS
C GPOL, GTAN, AND, TOTC, GEFF, GPBI, GTOT, SUMC) TO ZERO
QS=0.
LAPTH=0.
HO=0.
DO 13 J=1,11
DO 13 K=1,50
XDIS(K,J)=0
YDIS(K,J)=0
TOTPTH=0.
DO 14 J=1,99
DO 14 K=1,3
EXITPT(J,K)=0
DO 36 J=1,30
GAND(J)=0.
GANG(J)=0.
GCPS(J)=0.
GPOL(J)=0.
GTAN(J)=0.
AND(J)=0
DO 15 J=1,100
DO 16 K=1,4
TOTC(K,J)=0
GEFF(J)=0.
GPBI(J)=0.
GTOT(J)=0.
SUMC(J)=0

COMMENT READ AND WRITE (T, W, H), THE X, Y, AND Z DIMENSIONS OF THE
C DETECTOR RESPECTIVELY. FOR EACH PLANE AFTER THE 0TH, (H) IS
C THE DISTANCE FROM THE PREVIOUS PLANE. THE FORMAT IS, (F10.4).
DO 17 J=1,NPLANE
READ(5,1003) T(J), W(J), H(J)
WRITE(6,1004) J, T(J), W(J), H(J)

COMMENT SET THE VALUES OF (TGR, WGR) EQUAL TO (T, W) AT EACH PLANE
C LISTED IN (GRAPH), AS WELL AS TO THE 0TH PLANE.
TGR(1)=T0
WGR(1)=W0
DO 18 J=2,11
IF(GRAPH(J).EQ.0) GO TO 100
TGR(J)=T(GRAPH(J))
18 WGR(J)=W(GRAPH(J))

COMMENT DETERMIN 100./THE LONGEST POSSIBLE PATH LENGTH), (LPPTH)
100 IF(NPLANE.EQ.1) GO TO 104
LPPTH=100./SORT(1/2*(NPLANE-1)/2)**2)

R-2
\[
A(\left(\frac{W(\text{NPLANF})}{2} + \frac{W(\text{NPLANF}-1)}{2}\right)^2 + H(\text{NPLANF})^2) + H(\text{NPLANF})^2)
\]

\[
\text{LP} = 100. / \sqrt{\text{CI(\text{NPLANE}-1)}^2 + \text{T(\text{NPLANE}-1)}^2}
\]

\[
\text{LPPTH} = 100. / \sqrt{\left(\frac{\text{T}(1)}{2} + \text{T0}/2\right)^2 + \left(\frac{\text{W}(1)}{2} + \text{WO0}/2\right)^2 + \text{H}(1)^2}
\]

\[
\text{I.P} = 100. / \sqrt{\text{T0}^2 + \text{WO}^2}
\]

\[
\text{IF}(\text{LPPTH} \gt \text{LP}) \text{LPPTH} = \text{LP}
\]

\[
\text{REM} = 0
\]

\[
\text{COMMENT THIS DO LOOP FINDS A RANDOM PARTICLE AND SENDS IT THROUGH EACH DETECTOR LEVEL WHILE COLLECTING INFORMATION ABOUT IT}
\]

\[
\text{DO } J = 1, \text{NPAR}
\]

\[
\text{COMMENT AT EVERY 3000TH PARTICLE CALL (RFMTIM) AND TEST TO SEE IF THERE IS ENOUGH TIME FOR ANOTHER 3000 PARTICLES AND TO FINISH THE OUTPUT SECTION OF THIS PROGRAM. IF NOT GO DIRECTLY TO THE OUTPUT SECTION USING ONLY AS MANY PARTICLES AS HAVE ALREADY BEEN PROCESSED. THE INDICATED TIMES, 14 AND 5, ARE APPROPRIATE TO MOST DATA SETS ON AN IBM 360/91 AND SHOULD BE MODIFIED FOR DIFFERENT MACHINES.}
\]

\[
\text{REM} = \text{RFMTIM} + 1
\]

\[
\text{IF}(\text{REM} \neq 3000) \text{GO TO 103}
\]

\[
\text{CALL RFMTIM(\text{ICPU}, \text{IIO})}
\]

\[
\text{IF}(\text{ICPU} \leq 14) \text{OR} (\text{IIO} \leq 5) \text{GO TO 106}
\]

\[
\text{COMMENT SET THE VALUES OF (TOPW, TOPT, IX); INITLIZE (PTHLNG)}
\]

\[
\text{REM} = 0
\]

\[
\text{103 TOPW} = \text{WO}
\]

\[
\text{TOPT} = \text{T0}
\]

\[
\text{PTHLNG} = 0.
\]

\[
\text{IX} = \text{IY}
\]

\[
\text{CALL RANDU(IX, IY, YFL)}
\]

\[
\text{COMMENT FIND A RANDOM (YO) ANYWHERE WITHIN THE OTH PLANE}
\]

\[
\text{YO} = \text{WO} \times (\text{YFL} - .5)
\]

\[
\text{IX} = \text{IY}
\]

\[
\text{CALL RANDU(IX, IY, YFL)}
\]

\[
\text{COMMENT FIND A RANDOM (XO) ANYWHERE WITHIN THE OTH PLANE}
\]

\[
\text{XO} = \text{T0} \times (\text{YFL} - .5)
\]

\[
\text{IX} = \text{IY}
\]

\[
\text{CALL RANDU(IX, IY, YFL)}
\]

\[
\text{COMMENT FIND A RANDOM (THETA, STHETA, CTHETA) FOR AN ISOTROPIC DISTRIBUTION OF INCIDENT FLUX. THE FOLLOWING EXPRESSIONS FOR (THETA, STHETA, CTHETA), WHICH ARE LISTED AS COMMENTS, MAY REPLACE THOSE LISTED FOR FORTRAN, TO OBTAIN A COS-SQ DISTRIBUTION OF INCIDENT FLUX.}
\]

\[
\text{SSTETHA} = \text{SQRT(0.9999999\text{-}0.99999\text{*SQRT(YFL))}}
\]

\[
\text{CSTETHA} = \text{SQRT(SQRT(0.000001\text{*}0.99999\text{*YFL))}}
\]

\[
\text{CSTETHA} = \text{SQRT(SQRT(0.000001\text{*}0.99999\text{*YFL}})
\]

\[
\text{STETHA} = \text{ARSIN(THETHA)}
\]

\[
\text{IX} = \text{IY}
\]

\[
\text{CALL RANDU(IX, IY, YFL)}
\]

\[
\text{COMMENT FIND A RANDOM (PHI, SPHI, CPHI) FOR ANGLES UP TO 90. DEGREES}
\]

\[
\text{PHI} = 1.570796 \times \text{YFL} + .000001
\]

B-3
\[ \text{Sphi} = \sin(\text{phi}) \]
\[ \text{Cphi} = \cos(\text{phi}) \]

**COMMENT** Determine if there are any values in (graph), if not skip this section where distribution data is collected for the 0th plane.

If(graph(2).eq.0) go to 107

**COMMENT** \((\text{angt,angy,angx})\) are used as subscripts, \((\text{angt})\) divides theta into 30 equal parts, \(19.09859 = \text{number of degrees in a radian}/3\). \((\text{angy,angx})\) each put all of the particles into 4 equal bins on the basis of their distances from their respective axes.

\[ \text{angt} = \text{theta} \times 19.09859 + 1 \]
\[ \text{angy} = \text{abs}(y) \times 8/\text{wo} + 1 \]
\[ \text{angx} = \text{abs}(x) \times 8/\text{to} + 1 \]

**COMMENT** \((\text{angdis})\) accumulates the angular distribution by \(x\) and \(y\) bin at 11 different planes, it includes all particles that pass through each plane and, in this case, through the 0th plane.

\[ \text{angdis(angx,angy,angt,1)} = \text{angdis(angx,angy,angt,1)} + 1 \]

**COMMENT** \((\text{qx,qy})\) are used as subscripts to \((\text{xdis,ydis})\). They divide their respective axes into 50 bins numbered from the axes out, so that the bins are strips, running perpendicular to their respective axes.

\[ \text{qx} = \text{abs}(x) \times 100/\text{to} + 1 \]
\[ \text{qy} = \text{abs}(y) \times 100/\text{wo} + 1 \]
\[ \text{xdis(qx,1)} = \text{xdis(qx,1)} + 1 \]
\[ \text{ydis(qy,1)} = \text{ydis(qy,1)} + 1 \]

**COMMENT** In the loop starting at 'do 20' each level is treated as every other, because the 0th plane has its dimensions given in the variables \((\text{to,wo})\), new variables are temporarily used. They are \((\text{topx,topy,ktop,typw})\) which respectively stand for the \(x\) coordinate of the particle at the top of the level, the \(y\) coordinate, the \(x\) dimension, and the \(y\) dimension.

107 topx=xo
topy=y0

**COMMENT** This loop is the heart of the program, it sends the particle through each detector level while accumulating data.

do 20 k=1,nplane

**COMMENT** Save the value of \((k)\)

ks=k
qc=stheta/ctheta*h(k)

**COMMENT** Find the coordinates of the particle in plane \(k\).

\[ x(k) = \text{topx} + \text{cphi} \times \text{qc} \]
\[ y(k) = \text{topy} + \text{sphi} \times \text{qc} \]

**COMMENT** Determine the position of the particle relative to the edges of layer \(k\).

\( \text{pos}=1\), inside the edges of the plane
\( \text{pos}=2\), to the right or left of the plane
\( \text{pos}=3\), to the front or back of the plane
\( \text{pos}=4\), to both the right or left and the front or back of the plane

\( \text{pos}=4 \)
if(abs(x(k)).le.t(k)/2.) pos=pos-2
IF(ABS(Y(K)).LE.W(K)/2.)POS=POS-1
GO TO(108,109,110,111),POS

COMMENT SEARCH FOR A VALUE IN (GRAPH) THAT MATCHES THIS PLANF.
C IF THERE IS ONE SAVF THE SUBSCRIPT OF (GRAPH) THAT IT
C MATCHES.
108 DO 21 L=2,11
   LS=L
   IF(GRAPH(L).EQ.K)GO TO 112
21 CONTINUE

COMMENT ACCUMULATF (XDIS,YDIS,ANGDIS) AS IN THE PREVIOUS CASE
112 OX=ARS(X(K))*100/T(K)+1
   QY=ABS(Y(K))*100/W(K)+1
   XDIS(OX,LS)=XDIS(OX,LS)+1
   YDIS(QY,LS)=YDIS(QY,LS)+1
   ANGX=ARS(X(K))*8/T(K)+1
   ANGY=ABS(Y(K))*8/W(K)+1
   ANGDIS(ANGX,ANGY,ANGT,LS)=ANGDIS(ANGX,ANGY,ANGT,LS)+1

COMMENT IF THERE ARE MORE PLANES, REINITIALIZE TO START ANOTHER LEVEL.
C IF NOT, COMPUTE THE PATH LENGTH, ADD THAT TO THE TOTAL PATH,
C LENGTH, AND FIND THE CORRECT BIN IN WHICH TO INCREMENT (AND).
113 IF(K.NE.NPLANE)GO TO 114
   PTHLNG=H(K)/CTHETA
   TOTPTH=TOTPTH+PTHLNG
   Q1=THETA*19.09859+1
   AND(Q1)=AND(Q1)+1
   GO TO 123

COMMENT IF THIS IS NOT THE LAST DETECTOR LEVEL, INCREMENT (EXITPT)
C WHICH TELLS WHERE THE PARTICLES LEFT THE DETECTOR, THEN GET
C ANOTHER PARTICLE.
109 IF(K.NE.NPLANE)GO TO 19

COMMENT FIND THE PATH LENGTH BY TAKING THE LEAST POSITIVE VALUE OF
C (PTHLNG,QPTH).
   PTHLNG=PATH1(TOPW,TOPY,W(K),Y(K))
   QPTH=PATH2(TOPY,TOPW,W(K),Y(K))
   GO TO 124

110 IF(K.NE.NPLANE)GO TO 19

COMMENT FIND THE PATH LENGTH BY TAKING THE LEAST POSITIVE VALUE OF
C (PTHLNG,QPTH).
   PTHLNG=PATH1(TOPT,TOPX,T(K),X(K))
   QPTH=PATH2(TOPX,TOPT,T(K),X(K))
124 IF((PTHLNG.LT.0.).OR.((QPTH.LT.PTHLNG).AND.(0..LE.QPTH)))
   PTHLNG=QPTH
   GO TO 115

COMMENT FIND THE PATH LENGTH BY TAKING THE LEAST, POSITIVE VALUE
C OF (QPTH1,QPTH11,QPTH2,QPTH22)
111 QPTH1=PATH1(TOPT,TOPX,T(K),X(K))
   QPTH11=PATH2(TOPX,TOPT,T(K),X(K))
   QPTH2=PATH1(TOPY,TOPY,W(K),Y(K))
   QPTH22=PATH2(TOPY,TOPW,W(K),Y(K))
COMMENT IT IS POSSIBLE THAT WHEN CALCULATING IN (PATH1,PATH2) A
DIVISION BY ZERO WILL BE ATTEMPTED. THIS IS POSSIBLE BECAUSE
A PARTICLE MAY BE PARALLEL TO ONE OF THE SIDES OF THE DETECTOR
TO WITHIN THE ACCURACY OF THE MACHINE. THIS HAS NO DETRIMENTAL
EFFECT ON THE RESULTS, AND A CALL TO (ERRSET), CANCELS THE
ERROR MESSAGE, EXCEPT FOR A STATEMENT AT THE END OF THE OUTPUT
CALL ERRSET(210,256,-1,1)
IF((QPTH1.LE.0.).OR.((QPTH11.LT.QPTH1).AND.(0..LT.QPTH11)))
AQPTH1=QPTH11
IF((QPTH2.LE.0.).OR.((QPTH22.LT.QPTH2).AND.(0..LT.QPTH22)))
AQPTH2=QPTH22
COMMENT WHILE FINDING THE LEAST OF (QPTH1,QPTH2), ALSO SET (POS) TO
LEFT THE DETECTOR.
IF(QPTH11.LT.QPTH2)GO TO 116
IF(K.EQ.NPLANE)PTHLNQ=QPTH2
POS=2
GO TO 115
116 IF(K.EQ.NPLANE)PTHLNQ=QPTH1
POS=3
GO TO 115
COMMENT RESET (TOPX,TOPY,TOW1,TOPT) TO VALUES FOR THE NEXT DETECTOR
LEVEL.
114 TOPX=X(K)
TOPY=Y(K)
TOP1=!(K)
TOPT=T(K)
20 CONTINUE
COMMENT IF (PTHLNQ) WAS NOT ASSIGNED A VALUE FOR THIS PARTICLE THEN
FIND THE ACTUAL PHYSICAL PATH LENGTH, INCREMENT (TOTC), AND
TEST TO SEE IF THIS WAS THE LONGEST PATH LENGTH SO FAR.
115 IF(PTHLNQ.EQ.0.)GO TO 19
PTHLNQ=PTHLNQ*H(KS)/CTIHTA
123 QI=LPPTH*PTHLNQ+1
TOTC (POS,QI)=TOTC(POS,QI)+1
IF(PTHLNQ.GT.LAPTH)LAPTH=PTHLNQ
19 EXITPT(KS,POS)=EXITPT(KS,POS)+1
GO TO 117
COMMENT IF INSUFFICIENT TIME REMAINED WHEN (REMTIM) WAS CALLED, SET
(NPAR) TO THE ACTUAL NUMBER OF PARTICLES PROCESSED AND SET
THE FLAG, (TIME), TO ONE.
106 NPAR=J-1
TIME=1
WRITE(6,1005)NPAR
COMMENT WRITE OUT THE LONGEST POSSIBLE PATH LENGTH, THE LONGEST ACTUAL
PATH LENGTH, AND THE AVERAGE PATH LENGTH.
117 WLPPTH=100./LPPTH
AVEPTH=TOTPTH/EXITPT(NPLANE,1)
WRITE(6,1006)WLPPTH,LAPTH,AVEPTH
COMMENT IF THE DETECTOR HAS THE FORM OF A RECTANGULAR PARALLELEPIPED,
C COMPUTE THE GEOMETRICAL FACTOR ANALYTICALLY AND WRITE IT OUT.
DO 22 J=1,NPLANE
IF((T(J).NE.T0) .OR.(W(J).NE.W0))GO TO 118
22 H0=H(J)+HO
SQX=SQRT(H0*H0+W0*W0)
SQY=SQRT(H0*H0+TO*TO)
CGF=2*TO*SQX*ATAN(TO/SQX)+2*WO*SQY*ATAN(W0/SQY)-2*HO*TO*
AATAN(TO/H0)-2*WO*ATAN(W0/H0)-HO*HO*ALOG((HO*HO*(HO+HO+
BT0*TO+W0*W0))/(SQX*SQX*SQY*SQY))
WRITE(6,1007)CGF
COMMENT FIND THE GEOMETRICAL FACTOR AND WRITE IT OUT.
118 GF=3.141593*T0*WO*EXITPT(NPLANE,1)/NPAR
WRITE(6,1008)GF
COMMENT WRITE OUT THE NUMBER OF PARTICLES THAT LEFT THE DETECTOR, BY
C PLANE, AT EACH LEVEL.
DO 25 J=1,NPLANE
25 WRITE(6,1009)J,(EXITPT(J,K),K=1,3)
WRITE(6,1010)
COMMENT FIND THE EFFECTIVE GEOMETRICAL FACTOR FOR ALL PARTICLES THAT
C HAVE A PATH LENGTH OF (PBIN) OR GREATER.
DO 26 J=1,100
26 TOTC(4,J)=TOTC(1,J)+TOTC(2,J)+TOTC(3,J)
SUMC(100)=TOTC(4,100)
DO 25 J=1,99
25 SUMC(100-J)=TOTC(4,100-J)+SUMC(101-J)
DO 26 J=1,100
EFFGF=SUMC(J)*3.141593*T0*WO/NPAR
IF(EFFGF.EQ.0.)GO TO 119
PBIN=J/LPPTH
COMMENT FILL (GPBI,GTOT,GEFF), WRITE OUT THE PATH LENGTH BIN, (PBIN),
C THE NUMBER OF PARTICLES HAVING (PBIN) PATH LENGTH ACCORDING TO
C WHERE THEY EXIT THE LAST DETECTOR LEVEL AND THE EFFECTIVE
C GEOMETRICAL FACTOR FOR PARTICLES OF (PBIN) PATH LENGTH OR
C GREATER.
GPBI(J)=PBIN
GTOT(J)=TOTC(4,J)
GEFF(J)=EFFGF
26 WRITE(6,1011)PBIN,TOTC(2,J),TOTC(3,J),TOTC(1,J),TOTC(4,J),EFFGF
COMMENT PLOT (EFFGF,TOTC(4,*) VS. (PBIN).
119 CALL G(5,SUBH1,GPBI,GEFF,J)
CALL G(5,SUBH2,GPBI,GTOT,J)
WRITE(6,1012)
ZERO=NPAR
COMMENT FIND THE COUNTS PER SQUARE DEGREE, (CPSD), AT EACH ANGULAR
C BIN, (POLAR).
DO 27 J=1,30
QS=QS+AND(J)
POLAR=3*J

B-7
COMMENT FILL (GAND, GPOL, GCPS) FOR PLOTTING.
GAND(J) = AND(J)
GPOL(J) = POLAR
CPSD = AND(J)/(20626.48*COS((J-1)*5.235988E-2) - COS(J*5.235988E-2))
GCPS(J) = CPSD

COMMENT WRITE OUT THE ANGULAR BIN, THE NUMBER OF PARTICLES IN THAT BIN
AND THE (CPSD), SUPPRESS ZEROS BY TESTING WITH (QS), SAVE THE
VALUE OF (J).
WRITE(6,1013) POLAR, AND(J), CPSD
JS = J
IF(QS.EQ.EXITP(NPLANE,1)) GO TO 120
27 CONTINUE

COMMENT PLOT (AND, CPSD) VS. (POLAR)
120 CALL G(5,SUBH3,GPOL,GAND,JS)
CALL G(5,SUBH4,GPOL,GCPS,JS)

COMMENT WRITE OUT (XDIS, YDIS) AT EACH PLANE WHERE THEY WERE ACCUMULATED
BY THE BIN FROM THEIR RESPECTIVE AXIS, (XB,YB). SUPPRESS ZEROS.
DO 28 J=1,11
IF(((J.EQ.1).AND.(GRAPH(2).EQ.0)).OR.((J.NE.1).AND.(GRAPH(J).EQ.0)
A)) GO TO 10
IF(J.EQ.1) GO TO 121
13 = GRAPH(J)
12 = GRAPH(J-1)+1
DO 29 I=12,13
29 ZERO = ZERO - EXITPT(I,2) - EXITPT(I,3)
121 WRITE(6,1014) GRAPH(J), ZERO
DO 30 K=1,50
XB = K*GRAPH(J)/100.
YB = K*GRAPH(J)/100.

COMMENT FILL (GX, GYR, GXD1, GYD1) FOR PLOTTING.
GX(K) = XB
GYR(K) = YB
GXD1(K) = XDIS(K,J)
GYD1(K) = YDIS(K,J)
30 WRITE(6,1015) K, XB, XDIS(K,J), YB, YDIS(K,J)

COMMENT PLOT (XDIS(*,J)) VS. (XB) AND (YDIS(*,J)) VS. (YB)
CALL G(5, SUBH5, GX, GXD1, 50)
CALL G(5, SUBH6, GYR, GYD1, 50)
WRITE(6,1016) GRAPH(J)

TTANG = 0

COMMENT IN THIS LOOP FIND THE TOTAL NUMBER OF PARTICLES, (TANG), THAT
WERE IN EACH ANGULAR BIN, (ANGLE), AT EACH PLANE LISTED IN
(GRAPH(K)) AND FILL (GANG, GTAN) FOR PLOTTING. ZEROS ARE
SUPPRESSED WHEN WRITING OUT (ANGLE, ANGDIS, TANG).
DO 31 K=1,30
TANG = 0
ANGLE = 3.*K
GANG(K) = ANGLE
DO 32 I=1,4
DO 32 M=1,4
32 TANG = TANG + ANGDIS(N, L, K, J)
WRITE (6, 1017) ANGLE, ((ANGDIS(N, L, K, J), L=1, 4), M=1, 4), TANG
TTANG = TTANG + TANG
GTAN(K) = TANG
KS = K
IF (TTANG .EQ. ZER0) GO TO 122
31 CONTINUE

COMMENT FIND THE SIZE OF THE ARRAY TO BE TRANSMITTED INTO (G), THEN PLOT
C (TANG) VS. (ANGLE).
122 SIZE = ANGLE/3.
CALL G(5, SUBH7, GANG, GTAN, SIZE)
WRITE (6, 1018)

COMMENT IN THIS LOOP WRITE OUT (ANGDIS) AS (ANGD1, ANGD2) IN 9-DEGREE
C RINGS INSTEAD OF THE USUAL 3-DEGREE RINGS, IN A MATRIX FASHION,
C SUPPRESSING ZEROS.
DO 33 K = 1, 10, 2
IF (3*K-1) .GE. KS GO TO 28
DEG1 = K*9.
DEG2 = (K+1)*9.
WRITE (6, 1019) DEG1, DEG2
DO 33 L = 1, 4
DO 34 M = 1, 4
ANGD1(L) = 0
DO 35 M = 1, 14
DO 35 N = 1, 3
ANGD1(N) = ANGD1(N) + ANGDIS(N, L, N+(K-1)*3, J)
35 ANGD2(M) = ANGD2(M) + ANGDIS(L, N+(K-1)*3, J)
33 WRITE (6, 1020) L, (ANGD1(L), L, (ANGD2(M), M = 1, 4)
28 CONTINUE
10 CONTINUE
102 STOP

1000 FORMAT ('THE TIME IN MIN. REQUIRED FOR THIS DATA SET WAS', F6.2,
A' CPU ', F4.2, ' 1/0'
1001 FORMAT (20A4/12, 10F10.4, 15, 1012)
1002 FORMAT ('DATA SET NO. ', 15/'1X, 20A4/10NUMBER OF PLANES BEsIDES FIr
AST = ', 12/1X, 20A4/10NUMBER OF PARTICLES = ', 1F10.4, '01Y = ', 15/'0I = 'T(I)
B = W(I) = H(I) = '/', 2(2X, F10.4))
1003 FORMAT (3F10.4)
1004 FORMAT (14, 3(2X, F10.4))
1005 FORMAT (' THERE IS NOT ENOUGH TIME FOR MORE PARTICLES OR DATA SETS,
A NUMBER OF PARTICLES = ', 10)
1006 FORMAT ('LONGEST POSSIBLE PATH = ', F10.4/10LONGEST AChAL PATH = '
A, F10.4/10AVE. PATH = ', F10.4)
1007 FORMAT ('COMPUTED GEOMETRICAL FACTOR = ', F10.4)
1008 FORMAT (' GEOMETRICAL FACTOR = ', F10.4/10DETECTOR BOTTOM SI
ADE FRONT')
1009 FORMAT (4X, 12, 5X, 3(17, 2X))
1010 FORMAT (' PATH-LENGTH SIDECTS BMTCTS TOTCTS ', 4X,
A'EFFGF')
1011 FORMAT(1H,F8.2,4I10,F13.2)
1012 FORMAT('1POLAR ANGLE COUNTS OUT BOTTOM COUNTS PER 50-DEG')
1013 FORMAT(1H,17,14X,16,12X,1PF12.6)
1014 FORMAT('1DISTURBANCE OF INCIDENT PARTICLES AT PLANE NO. ',12/
A'TOTAL PARTICLES AT THIS PLANE = ',110/'0BIN X BIN NO. OF PAR
BITICLES Y BIN NO. OF PARTICLES')
1015 FORMAT(14,2(2X,F8.2,4X,I10,3X))
1016 FORMAT('1ANGULAR DISTURBANCE OF INCIDENT PARTICLES AT PLANE NO. ',
A,12/'0 ',51X,'ORDINATES (X,Y) '/' ANGLE (1,1) (1,2) (1,3)
B(1,4) (2,1) (2,2) (2,3) (2,4) (3,1) (3,2) (3,3) (3,4) (4,
C1) (4,2) (4,3) (4,4) TOTAL')
1017 FORMAT(2H,F4.1,16(1X,16),2X,18)
1018 FORMAT('1ANGULAR DISTURBANCE AT 90 DEG BINS')
1019 FORMAT(1H0,2(14X,F3.0,' DEG'),32X)/1H:2('Y/X 1 2
A 3 4',16X)/3H:38('*',15X,38('*'))
1020 FORMAT(1H,11,' *',4(1X,17,1X),'*',13X,11,' *',4(1X,17,1X),'*')
END
SUBROUTINE G(TYPE, HEAD, A, R, SIZE)
REAL*8 HEAD(5)
LOGICAL*1 CWR(103)
REAL ARAY(101), BARAY(101), A(SIZE), B(SIZE)
INTEGER IARAY(101), BARAY(101), QN, N(11), TYPE, WA, EXPB, ADIF, BDIF, ATYP, SIZE, HIST

COMMENT IF (HIST) EQUALS TO ONE THEN THIS SUBROUTINE MAKES HISTOGRAMS,
C IF (HIST) DOSE NOT EQUAL TO ONE THEN THIS SUBROUTINE MAKES
C GRAPHS.

HIST=1
J=101
ANAX=0.
BMAX=0.
DO 10 I=1,101
ARAY(I)=0.00001
BARAY(I)=0.00001
IF((I.LE.SIZE).AND.(A(I).GT.0.00001))ARAY(I)=A(I)
IF((I.LE.SIZE).AND.(B(I).GT.0.00001))BARAY(I)=B(I)
10 CONTINUE
DO 11 I=1,101
IF((ARAY(101).LT.0.00001).OR.(BARAY(101).LT.0.00001))GO TO 100
11 J=101-I
100 DO 12 I=1,J
IF(ARAY(I).GT.ANAX)ANAX=ARAY(I)
IF(BARAY(I).GT.BMAX)BMAX=BARAY(I)
12 CONTINUE
IF((ANAX.LE.5.).OR.(BMAX.LT.5.))GO TO 101
WRITE(6,1000)HEAD
IF(TYPE.LT.5)GO TO 106
TYP =1
IF(BMAX.GE.1.E5 )TYP =TYP +2
IF(ANAX.GE.1.E5 )TYP =TYP +1
106 GO TO (102,103,104,105), TYP
102 TYP=1
ADIF=ANAX/100+.995
BDIF=BMAX/100+.995
CWR(102)=64
CWR(103)=64
EXPR=1
WRITE(6,1001)
DO 13 I=1,J
IARAY(I)=(ARAY(I)+ADIF/2.)/ADIF+1.
13 IARAY(I)=(BARAY(I)+BDIF/2.)/BDIF+1
GO TO 107
103 TYP=2
ADIF=1
BDIF=BMAX/100+.995
CWR(102)=64
EXPR=1
CWR(103)=197
WRITE(6, 1002)
NO 14 I=1, J
IARAY(I)= (ALOG10(ARAY(I)) + .05) * 10 + 1
14 IBARAY(I)= (BARAY(I) + BDIF/2.)/BDIF + 1
GO TO 107
104 TYP=3
ADIF=AMAX/100+.995
BDIF=1
CWR(102)=197
CWR(103)=64
EXPB=10
WRITE(6, 1003)
DO 15 I=1, J
IARAY(I)= (ARAY(I) + ADIF/2.)/ADIF + 1.
15 IBARAY(I)= (ALOG10(BARAY(I)) + .05) * 10 + 1
GO TO 107
105 TYP=4
ADIF=1
BDIF=1
CWR(102)=197
CWR(103)=197
EXPB=10
WRITE(6, 1004)
DO 16 I=1, J
IARAY(I)= (ALOG10(ARAY(I)) + .05) * 10 + 1
16 IBARAY(I)= (ALOG10(BARAY(I)) + .05) * 10 + 1
107 IF(HIST.NE.1)GO TO 115
DO 22 I=1, 101
22 CVWR(I)=64
115 DO 17 J=1, 101
17 IF(HIST.EQ.1)GO TO 119
DO 18 J=1, 101
18 CIWR(J)=64
119 WA =BDIF*(101-I)/EXPB
DO 19 K=1, 11
19 IF(J.EQ.10*(K-1)+1) GO TO 113
CONTINUE
WRITE(6, 1005)
GO TO 114
113 WRITE(6, 1006) CWR(102), WA
114 DO 20 J=1, 101
20 CONTINUE
WRITE(6, 1007) (CWR(J), J=1, 101)
DO 21 I=1, 11
17 IF(TYP/2.EQ.TYP/2.) GO TO 116
Q(I)=ADIF*(I-1)*10
GO TO 21

R-12
116  Q(I)=I-1
21  CONTINUE
    WRITE (6,1008)(CWR(103),Q(I),I=1,11)
101  RETURN
1000  FORMAT(1H1,5A8)
1001  FORMAT('0THIS IS A SCALER PLOT')
1002  FORMAT('0THIS IS A SEMI-LOG PLOT IN X')
1003  FORMAT('0THIS IS A SEMI-LOG PLOT IN Y')
1004  FORMAT('0THIS IS A LOG-LOG PLOT')
1005  FORMAT(9X,'|',10(9X,'|'))
1006  FORMAT(1H+,A1,17,101('-')/10H+,1,10(9X,'|'))
1007  FORMAT(9H+,101A1)
1008  FORMAT(1H+,5X,11(A1,13,6X))
END
C. FLOW DIAGRAM OF THE
GEOMETRICAL FACTOR PROGRAM

C.1
DO i = 1, 100000
CALL REMTIM (ICPU, IIO)

I, EQ. 1

GRAPH (J) = 0

HEAD, NPLANE T0, W0, NPAR IY, (GRAPH (J), J = 2, 11)

TIME, EQ. 1

QS = 0.
LAPTH = 0.
H0 = 0.

C-1 - Q
WLPPTH = 100. / LPPTH
AVEPTH = TOTPTH / EXITPT (NPLANE, 1)

WLPPTH, LAPTTH, AVEPTH

DO 22
J = 1, NPLANE

T (J) NE TO

F

T W (J) NE W0

F

H0 = H (J) + H0

22

SQX = SQRT (H0 * H0 + W0 * W0)
SQY = SQRT (H0 * H0 + T0 * T0)
CGF = 2 * T0 * ATAN (T0 / SQX) + 2 * W0 * SOY * ATAN (W0 / SQY) -
2 * H0 * T0 * ATAN (T0 / H0) - 2 * W0 * H0 * ATAN (W0 / H0) -
H0 * H0 * ALOG ((H0 * H0 * (H0 * H0 + T0 * T0 + W0 * W0)) /
(SQX * SQX * SQY * SQY))

CGF

GF = 3.141593 * T0 * W0 * EXITPT (NPLANE, 1) / NPAR

GF

DO
J = 1, NPLANE

J, (EXITPT (J, K), K = 1,3)

(HEADING)

66
\[ \text{TOTC}(4, J) = \text{TOTC}(1, J) + \text{TOTC}(2, J) + \text{TOTC}(3, J) \]

\[ \text{SUMC}(100) = \text{TOTC}(4, 100) \]

\[ \text{SUMC}(100 - J) = \text{TOTC}(4, 100 - J) + \text{SUMC}(101 - J) \]

\[ \text{DO} \]

\[ J = 1, 100 \]

\[ \text{EFFGF} = \text{SUMC}(J) \times 3.141593 \times T0 \times W0 / \text{NPAR} \]

\[ \text{T} \]

\[ \text{EFFGF}, \text{EQ. 0.} \]

\[ F \]

\[ \text{PBIN} = J / \text{LPPTH} \]

\[ \text{GPBI}(J) = \text{PBIN} \]

\[ \text{GTOT}(J) = \text{TOTC}(4, J) \]

\[ \text{GEFF}(J) = \text{EFFGF} \]

\[ \text{PBIN, TOTC}(2, J), \text{TOTC}(3, J), \text{TOTC}(1, J), \text{TOTC}(4, J), \text{EFFGF} \]

\[ \text{CALL G} \]

\[ (5, \text{SUBH}1, \text{GPBI}, \text{GEFF}, J) \]

\[ \text{CALG} \]

\[ (5, \text{SUBH}2, \text{GPBI}, \text{GTOT}, J) \]

\[ \text{(HEADING)} \]

\[ \text{ZERO} = \text{NPAR} \]

\[ \text{DO 27} \]

\[ J = 1, 30 \]

\[ \text{QS} = \text{QS} + \text{AND}(J) \]

\[ \text{POLAR} = 3 \times J \]

\[ \text{GAND}(J) = \text{AND}(J) \]

\[ \text{GPOL}(J) = \text{POLAR} \]

\[ \text{CPSD} = \text{AND}(J) / (20626.48 \times \cos((J - 1) \times 5.235988 \times 10^2) - \cos(J \times 5.235988 \times 10^2)) \]

\[ \text{GCPS}(J) = \text{CPSD} \]