X-661-73-218

70428

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

(NASA-TM-X-70428) DETERMINATION OF THE GEOMETRICAL FACTOR AND OF THE SPATIAL DISTRIBUTIONS OF PARTICLES FOR A COSMIC RAY TELESCOPE USING A MONTE CARLO APPROACH (NASA) 45 p HC $4.25 CSCL 14B G3/14

N73-28498

Unclas 10542

MICHAEL HEMMER
CAROL JO CRANNELL

JULY 1973

GSFC

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

†Work supported in part by NASA Grant Number NGR-09-050-001.
*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,[1] 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[2] spatial coordinates (X0, Y0) 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 coordinates of a new simulated particle in the uppermost layer of the detector and the zenith and azimuthal trajectory angles are chosen by four separate calls to subroutine RANDU.[2] An isotropic distribution 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 2π radians.

After arrays for the graphing output are incremented, the next major, nested loop of the program is entered. The detector configuration 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 parallepiped, 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 94.6cm^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 assistnace with this program and Dr. Jonathan Ormes for his interest and constructive comments.

REFERENCES

1) C. J. Crannell, and J. F. Ormes, Nucl. Instr. and Methods 94, 179 (1971) and CUA-70-27 (1970).

2) Random, in this context, means that the value of the variable was set using RANDU from the System/360 Scientific Subroutine Package (360A-CM-03X) Version III Programmer's Manual, 77 (1968).

HYPOTHETICAL DETECTOR CONFIGURATION

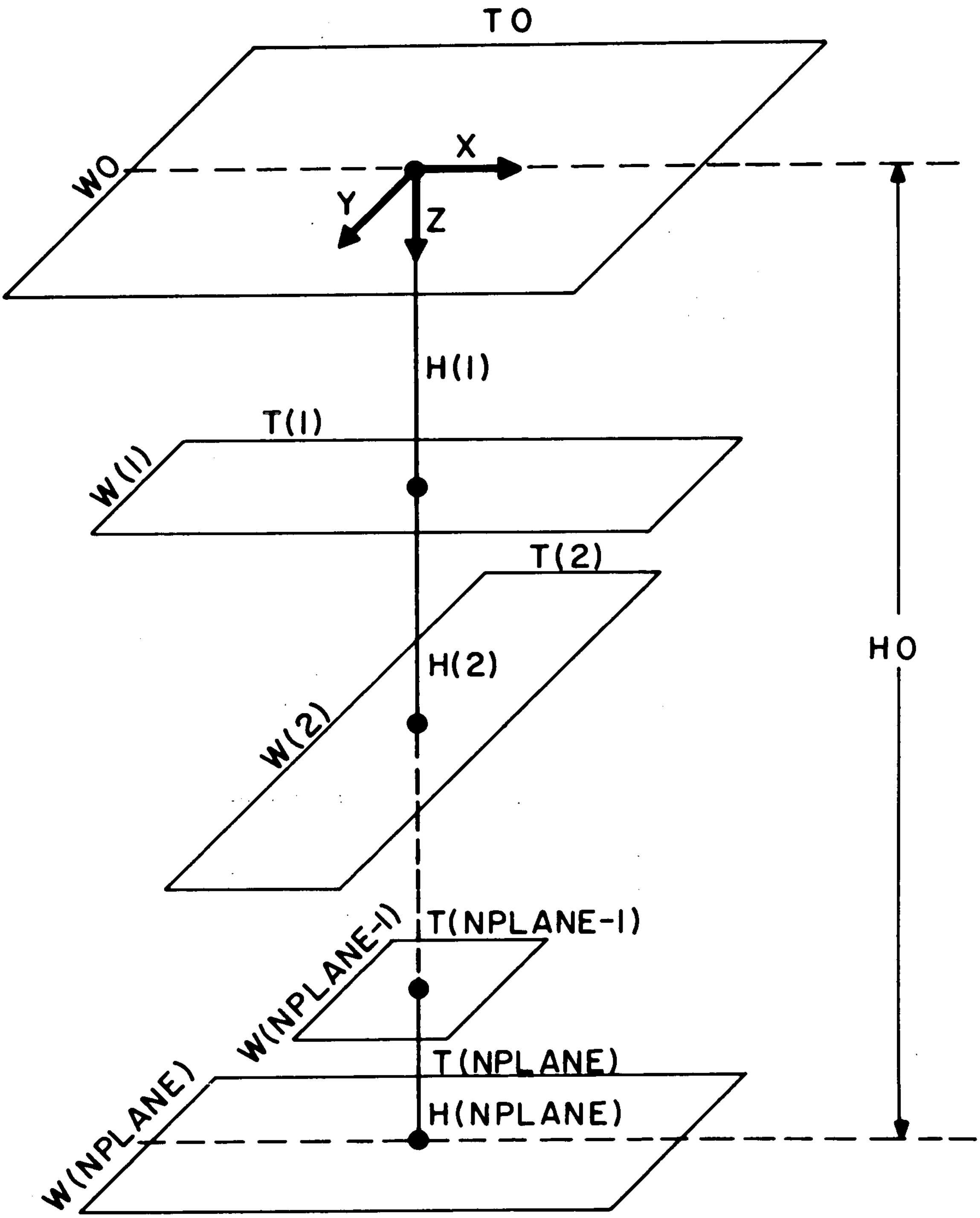


FIGURE 1

SIMULATED PARTICLE TRAJECTORY

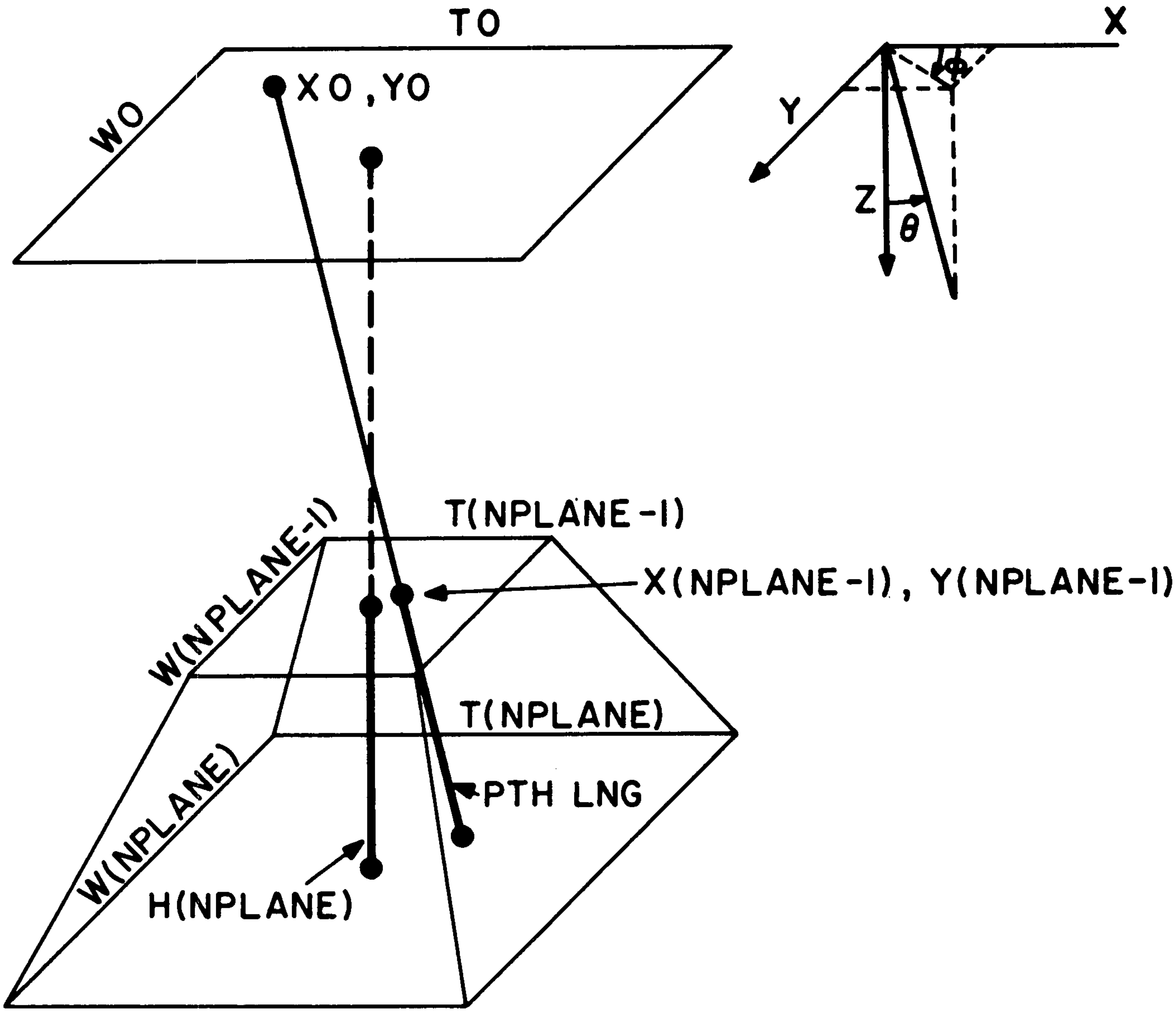


FIGURE 2

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 angle 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 |real| 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 |real| the average path length of all particles that enter the last detector level.

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

CPHI |real| cos(PHI)

CPSD |real| the particle counts per square degree.

CTHETA |real| cos(THETA)

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

DEG2 |real| 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.*LPPTH, 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 PATH1 or PATH2. It has no significnace 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; 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:

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

HEAD(5) |real*8| should contain the heading for the plot.

A(N) |real|, N between 1 and 100, contains the X coordinates of the data.

B(N) |real|, N between 1 and 100, contains the Y coordinates of the data.

SIZE |int| contains the dimensions of A and B.

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

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

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

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.

H0 |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.

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

IIOS |int| saves the previous value of IIO.

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 reference.

POS |int| used to determine a particles 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 or the plane,
(4) to both the front or back and right or left of the plane, that is, off a corner.

PTHLNG |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 T0 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 REMTIM.

SIZE |int| the last argument to G which contains the dimension of GAND and GTAN.

SPHI |real| sin(PHI)

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. PTHLNG''.

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) |int| 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 |int| 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 |int| 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.

T0 |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.

WIIOS |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 parellel 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) |real| the coordinate of a particle in the Y dimension at the Nth plane.

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

YDIS(50,11) |int| the distribution of particles along the Y axis, that is in strips parellel 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 |real| an argument to RANDU, it is returned with a random real number between 0.0 and 1.0.

Y0 |real| the Y coordinate of a random particle at the 0th plane.

ZERO |int| 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),HEAD(20),TGR(11),WGR(11),
     AGPRI(100),GEFF(100),GPOL(30),GCPS(30),GAND(30),GTOT(100),GTAN(30),
     BGANG(30),GXB(50),GYB(50),GXDI(50),GYDI(50),LAPTH,LPPTH,LP
      INTEGER EXITPT(99,3),POLAR,AND(30),SUMC(100),TOTC(4,100),OX,OY,
     AXDIS(50,11),YDIS(50,11),GRAPH(11),ANGX,ANGY,ANGT,ANGDIS(4,4,30,11)
     B,ZERO,TANG,ANGD1(4),ANGD2(4),TTANG,TIME,POS,OI,REM,SIZE
COMMENT   (SUBH1-7) CONTAIN THE HEADINGS FOR TRANSMISSION TO SUBROUTINE
C         (G) THEY ARE REAL*8 FOR CONVINENCE ONLY.
      REAL*8 SUBH1(5),SUBH2(5),SUBH3(5),SUBH4(5),SUBH5(5),SUBH6(5),
     ASUBH7(5)
      DATA SUBH1,SUBH2,SUBH3,SUBH4,SUBH5,SUBH6,SUBH7/'EFFGE VS','. PTHLN
     AG', 3*'        ','TOTAL CO','UNTS VS.',' PATH-LE','NGTH    ',' 
     B   ','COUNTS O','UT BOTTO','M VS. PO','LAR ANGL','E       ','CPSD
     C VS.',' POLAR A','NGLE    ',2*'        ','NO. OF P','ARTICLES',' V
     DS. X B','IN      ','        ','NO. OF P','ARTICLES',' VS. Y B','IN
     E     ','        ','TOTAL PA','RTICLES ','VS. ANGL','E       ','
     F     '/
COMMENT   (PATH1,PATH2) ARE STATMENT FUNCTION SUBPROGRAMS WHICH
C         CALCULATE A COEFFICIENT FOR THE PATH LENGTH OF THE PARTICLES.
      PATH1(A,B,C,D)=(A-2.*B)/(A-C+2.*(D-B))
      PATH2(A,B,C,D)=(2.*A+B)/(B-C+2.*(A-D))
COMMENT   (TIME) IS A FLAG, WHEN SET TO ONE THE PROGRAM STOPS
      TIME=0
COMMENT   THIS DO ALLOWS THE PROGRAM TO OPPERATE ON AS MANY AS 100000
C         DATA SETS.
      DO 10 I=1,100000
COMMENT   (REMTIM) IS AN IBM SUBROUTINE THAT RETURNS THE SECONDS
C         REMAINING OF CPU AND I/O TIME AS INTEGERS.
      CALL REMTIM(ICPU,IIO)
      IF(I.EQ.1)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.
      WICPUS=(ICPUS-ICPU)/60.
      WIIOS=(IIOS-IIO)/60.
      WRITE(6,1000)WICPUS,WIIOS
  101 ICPUS=ICPU
      IIOS=IIO
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(M,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 (IY), THE PLANES FOR WHICH DISTRUBUTION DATA IS
C         REQUESTED IN ASSENDING ORDER.  THE FORMAT IS, (20A4/I2,2F10.4,
C         I10,I5,10I2).
```

```
      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   INITLIZE (QS,LAPTH,H0,XDIS,YDIS,TOPT,EXITPT,GAND,GANG,GCPS
C         GPOL,GTAN,AND,TOTC,GEFF,GPBI,GTOT,SUMC) TO ZERO
      QS=0.
      LAPTH=0.
      H0=0.
      DO 13 J=1,11
      DO 13 K=1,50
      XDIS(K,J)=0
   13 YDIS(K,J)=0
      TOTPTH=0.
      DO 14 J=1,99
      DO 14 K=1,3
   14 EXITPT(J,K)=0
      DO 36 J=1,30
      GAND(J)=0.
      GANG(J)=0.
      GCPS(J)=0.
      GPOL(J)=0.
      GTAN(J)=0.
   36 AND(J)=0
      DO 15 J=1,100
      DO 16 K=1,4
   16 TOTC(K,J)=0
      GEFF(J)=0.
      GPBI(J)=0.
      GTOT(J)=0.
   15 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)
   17 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   DETERMINE 100./(THE LONGEST POSSIBLE PATH LENGTH), (LPPTH)
  100 IF(NPLANE.EQ.1)GO TO 104
      LPPTH=100./SQRT(((T(NPLANE)/2+T(NPLANE-1)/2)**2)+
```

```
   A((W(NPLANE)/2+W(NPLANE-1)/2)**2)+H(NPLANE)**2)
      LP=100./SQRT(W(NPLANE-1)**2+T(NPLANE-1)**2)
      GO TO 105
  104 LPPTH=100./SQRT(((T(1)/2+T0/2)**2)+((W(1)/2+W0/2)**2)+H(1)**2)
      LP =100./SQRT(T0*T0+W0*W0)
  105 IF(LPPTH.GT.LP)LPPTH=LP
      REM=0
COMMENT   THIS DO LOOP FINDS A RANDOM PARTICLE AND SENDS IT THROUGH
C         EACH DETECTOR LEVEL WHILE COLLECTING INFORMATION ABOUT IT
      DO 19 J=1,NPAR
COMMENT   AT EVERY 3000TH PARTICLE CALL (REMTIM) AND TEST TO SEE IF THERE
C         IS ENOUGH TIME FOR ANOTHER 3000 PARTICLES AND TO FINISH THE
C         OUTPUT SECTION OF THIS PROGRAM. IF NOT GO DIRECTLY TO THE
C         OUTPUT SECTION USING ONLY AS MANY PARTICLES AS HAVE ALREADY
C         BEEN PROCESSED.  THE INDICATED TIMES, 14 AND 5, ARE APPROPIATE
C         TO MOST DATA SETS ON AN IBM 360/91 AND SHOULD BE MODIFIED FOR
C         DIFFERENT MACHINES.
      REM=REM+1
      IF(REM.NE.3000)GO TO 103
      CALL REMTIM(ICPU,IIO)
      IF((ICPU.LE.14).OR.(IIO.LE.5))GO TO 106
COMMENT   SET THE VALUES OF (TOPW,TOPT,IX); INITLIZE (PTHLNG)
      REM=0
  103 TOPW=W0
      TOPT=T0
      PTHLNG=0.
      IX=IY
      CALL RANDU(IX,IY,YFL)
COMMENT   FIND A RANDOM (Y0) ANYWHERE WITHIN THE 0TH PLANE
      Y0=W0*(YFL-.5)
      IX=IY
      CALL RANDU(IX,IY,YFL)
COMMENT   FIND A RANDOM (X0) ANYWHERE WITHIN THE 0TH PLANE
      X0=T0*(YFL-.5)
      IX=IY
      CALL RANDU(IX,IY,YFL)
COMMENT   FIND A RANDOM (THETA,STHETA,CTHETA) FOR AN ISOTROPIC
C         DISTURBUTION OF INCIDENT FLUX.  THE FOLLOWING EXPRESSIONS FOR
C         (STHETA,CTHETA), WHICH ARE LISTED AS COMMENTS, MAY REPLACE
C         THOSE LISTED AS FORTRAN, TO OBTAIN A COS-SQ DISTRIBUTION OF
C         INCIDENT FLUX.
C     STHETA=SQRT(.999999-.99999*SQRT(YFL))
C     CTHETA=SQRT(SQRT(.000001+.99999*YFL))
      CTHETA=SQRT(.999999-.99999*YFL)
      STHETA=SQRT(.000001+.99999*YFL)
      THETA=ARSIN(STHETA)
      IX=IY
      CALL RANDU(IX,IY,YFL)
COMMENT   FIND A RANDOM (PHI,SPHI,CPHI) FOR ANGLES UP TO 90. DEGREES
      PHI=1.570796*YFL+.000001
```

```
      SPHI=SIN(PHI)
      CPHI=COS(PHI)
COMMENT   DETERMINE IF THERE ARE ANY VALUES IN (GRAPH), IF NOT SKIP THIS
C         SECTION WHERE DISTRIBUTION DATA IS COLLECTED FOR THE 0TH PLANE
      IF(GRAPH(2).EQ.0)GO TO 107
COMMENT   (ANGT,ANGY,ANGX) ARE USED AS SUBSCRIPTS, (ANGT) DIVIDES THETA
C         INTO 30 EQUAL PARTS, 19.09859=NUMBER OF DEGREES IN A RADIAN/3.
C         (ANGY,ANGX) EACH PUT ALL OF THE PARTICLES INTO 4 EQUAL BINS ON
C         THE BASIS OF THEIR DISTANCES FROM THEIR RESPECTIVE AXES.
      ANGT=THETA*19.09859+1
      ANGY=ABS(Y0)*8/W0+1
      ANGX=ABS(X0)*8/T0+1
COMMENT   (ANGDIS) ACCUMULATES THE ANGULAR DISTURBUTION BY X AND Y BIN
C         AT 11 DIFFERENT PLANES, IT INCLUDES ALL PARTICLES THAT PASS
C         THROUGH EACH PLANE AND, IN THIS CASE, THROUGH THE 0TH PLANE.
      ANGDIS(ANGX,ANGY,ANGT,1)=ANGDIS(ANGX,ANGY,ANGT,1)+1
COMMENT   (QX,QY) ARE USED AS SUBSCRIPTS TO (XDIS,YDIS).  THEY DIVIDE
C         THEIR RESPECTIVE AXES INTO 50 BINS NUMBERED FROM THE AXES OUT,
C         SO THAT THE BINS ARE STRIPS, RUNNING PERPENDICULAR TO THEIR
C         RESPECTIVE AXES.
      QX=ABS(X0)*100/T0+1
      QY=ABS(Y0)*100/W0+1
      XDIS(QX,1)=XDIS(QX,1)+1
      YDIS(QY,1)=YDIS(QY,1)+1
COMMENT   IN THE LOOP STARTING AT 'DO 20' EACH LEVEL IS TREATED AS EVERY
C         OTHER, BECAUSE THE 0TH PLANE HAS ITS DIMENSIONS GIVEN IN THE
C         VARIABLES (T0,W0), NEW VARIABLES ARE TEMPORARLY USED.  THEY
C         ARE (TOPX,TOPY,TOPT,TOPW) WHICH RESPECTIVELY STAND FOR THE X
C         CORDINATE OF THE PARTICLE AT THE TOP OF THE LEVEL, THE Y
C         CORDINATE, THE X DIMENSION, AND THE Y DIMENSION.
  107 TOPX=X0
      TOPY=Y0
COMMENT   THIS LOOP IS THE HEART OF THE PROGRAM, IT SENDS THE PARTICLE
C         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)=TOPX+CPHI*QC
      Y(K)=TOPY+SPHI*QC
COMMENT   DETERMINE THE POSITION OF THE PARTICLE RELATIVE TO THE EDGES
C         OF LAYER K.
C         (POS)=1, INSIDE THE EDGES OF THE PLANE
C         (POS)=2, TO THE RIGHT OR LEFT OF THE PLANE
C         (POS)=3, TO THE FRONT OR BACK OF THE PLANE
C         (POS)=4, TO BOTH THE RIGHT OR LEFT AND THE FRONT OR BACK OF
C         THE PLANE
      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 PLANE.
C         IF THERE IS ONE SAVE 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
      GO TO 113
COMMENT   ACCUMULATE  (XDIS,YDIS,ANGDIS) AS IN THE PREVIOUS CASE
  112 OX=ABS(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=ABS(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
      QI=THETA*19.09859+1
      AND(QI)=AND(QI)+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)))
     APTHLNG=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(TOPW,TOPY,W(K),Y(K))
      QPTH22=PATH2(TOPY,TOPW,W(K),Y(K))
```

```
COMMENT   IT IS POSSIBLE THAT WHEN CALCULATING IN (PATH1,PATH2) A
C         DIVISION BY ZERO WILL BE ATTEMPTED.  THIS IS POSSIBLE BECAUSE
C         A PARTICLE MAY BE PARALLEL TO ONE OF THE SIDES OF THE DETECTOR
C         TO WITHIN THE ACCURACY OF THE MACHINE. THIS HAS NO DETRIMENTAL
C         EFFECT ON THE RESULTS, AND A CALL TO (ERRSET), CANCELS THE
C         ERROR MESSAGE, EXECPT 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
C         THE APPROPIATE VALUE TO REFLECT WHERE THE PARTICLE ACTUALLY
C         LEFT THE DETECTOR.
      IF(QPTH1.LT.QPTH2)GO TO 116
      IF(K.EQ.NPLANE)PTHLNG=QPTH2
      POS=2
      GO TO 115
  116 IF(K.EQ.NPLANE)PTHLNG=QPTH1
      POS=3
      GO TO 115
COMMENT   RESET (TOPX,TOPY,TOPW,TOPT) TO VALUES FOR THE NEXT DETECTOR
C         LEVEL.
  114 TOPX=X(K)
      TOPY=Y(K)
      TOPW=W(K)
      TOPT=T(K)
   20 CONTINUE
COMMENT   IF (PTHLNG) WAS NOT ASSIGNED A VALUE FOR THIS PARTICLE THEN
C         INCREMENT (EXITPT) AND CONTINUE.  IF A VALUE WAS ASSIGNED,
C         FIND THE ACTUAL PHYSICAL PATH LENGTH, INCREMENT (TOTC), AND
C         TEST TO SEE IF THIS WAS THE LONGEST PATH LENGTH SO FAR.
  115 IF(PTHLNG.EQ.0.)GO TO 19
      PTHLNG=PTHLNG*H(KS)/CTHETA
  123 QI=LPPTH*PTHLNG+1
      TOTC (POS,QI)=TOTC(POS,QI)+1
      IF(PTHLNG.GT.LAPTH)LAPTH=PTHLNG
   19 EXITPT(KS,POS)=EXITPT(KS,POS)+1
      GO TO 117
COMMENT   IF INSUFFICIENT TIME REMAINED WHEN (REMTIM) WAS CALLED, SET
C         (NPAR) TO THE ACTUAL NUMBER OF PARTICLES PROCESSED AND SET
C         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
C         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 PARALLEPIPED,
C         COMPUTE THE GEOMETRICAL FACTOR ANALYTICALLY AND WRITE IT
C         OUT.
      DO 22 J=1,NPLANE
      IF((T(J).NE.T0) .OR.(W(J).NE.W0))GO TO 118
   22 H0=H(J)+H0
      SQX=SQRT(H0*H0+W0*W0)
      SQY=SQRT(H0*H0+T0*T0)
      CGF=2*T0*SQX*ATAN(T0/SQX)+2*W0*SQY*ATAN(W0/SQY)-2*H0*T0*
     AATAN(T0/H0)-2*W0*H0*ATAN(W0/H0)-H0*H0*ALOG((H0*H0*(H0*H0+
     BT0*T0+W0*W0))/(SQX*SQX*SQY*SQY))
      WRITE(6,1007)CGF
COMMENT   FIND THE GEOMETRICAL FACTOR AND WRITE IT OUT.
  118 GF=3.141593*T0*W0*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 23 J=1,NPLANE
   23 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 24 J=1,100
   24 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*W0/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
```

```
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
C         AND THE (CPSD), SUPPRESS ZEROS BY TESTING WITH (QS), SAVE THE
C         VALUE OF (J).
      WRITE(6,1013)POLAR,AND(J),CPSD
      JS=J
      IF(QS.EQ.EXITPT(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
C         BY THE BIN FROM THEIR RESPECTIVE AXIS, (XB,YB). SUPRESS 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
      I3=GRAPH(J)
      I2=GRAPH(J-1)+1
      DO 29 I1=I2,I3
   29 ZERO=ZERO-EXITPT(I1,2)-EXITPT(I1,3)
  121 WRITE(6,1014)GRAPH(J),ZERO
      DO 30 K=1,50
      XB=K*TGR(J)/100.
      YB=K*WGR(J)/100.
COMMENT   FILL (GXB,GYB,GXDI,GYDI) FOR PLOTTING.
      GXB(K)=XB
      GYB(K)=YB
      GXDI(K)=XDIS(K,J)
      GYDI(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,GXB,GXDI,50)
      CALL G(5,SUBH6,GYB,GYDI,50)
      WRITE(6,1016)GRAPH(J)
      TTANG=0
COMMENT   IN THIS LOOP FIND THE TOTAL NUMBER OF PARTICLES, (TANG), THAT
C         WERE IN EACH ANGULAR BIN, (ANGLE), AT EACH PLANE LISTED IN
C         (GRAPH(K)) AND FILL (GANG,GTAN) FOR PLOTTING.  ZEROS ARE
C         SUPPRESSED WHEN WRITING OUT (ANGLE,ANGDIS,TANG)..
      DO 31 K=1,30
      TANG=0
      ANGLE=3.*K
      GANG(K)=ANGLE
      DO 32 L=1,4
      DO 32 M=1,4
```

```
   32 TANG=TANG+ANGDIS(M,L,K,J)
      WRITE(6,1017)ANGLE,((ANGDIS(M,L,K,J),L=1,4),M=1,4),TANG
      TTANG=TTANG+TANG
      GTAN(K)=TANG
      KS=K
      IF(TTANG.EQ.ZERO)GO TO 122
   31 CONTINUE
COMMENT   FIND THE SIZE OF THE ARAY TO BE TRANSMITED 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         BINS INSTEAD OF THE USUAL 3-DEGREE BINS, 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(M)=0
   34 ANGD2(M)=0
      DO 35 M=1,4
      DO 35 N=1,3
      ANGD1(M)=ANGD1(M)+ANGDIS(M,L,N+(K-1)*3,J)
   35 ANGD2(M)=ANGD2(M)+ANGDIS(M,L,N+(K-1)*3+3,J)
   33 WRITE(6,1020)L,(ANGD1(M),M=1,4),L,(ANGD2(M),M=1,4)
   28 CONTINUE
   10 CONTINUE
  102 STOP
 1000 FORMAT('0THE TIME IN MIN. REQUIRED FOR THIS DATA SET WAS',F6.2,
     A' CPU ',F4.2,' I/O')
 1001 FORMAT( 20A4/I2,2F10.4,I10,I5,I012)
 1002 FORMAT('1DATA SET NO. ',I5//1X,20A4/'0NUMBER OF PLANES BESIDES FIR
     AST = ',I2/'0NUMBER OF PARTICLES = ',I10/'0IY = ',I5/'0I =     T(I)
     B =      W(I) =      H(I) ='/'   0',2(2X,F10.4))
 1003 FORMAT(3F10.4)
 1004 FORMAT(I4,3(2X,F10.4))
 1005 FORMAT('0THERE IS NOT ENOUGH TIME FOR MORE PARTICLES OR DATA SETS,
     A NUMBER OF PARTICLES = ',I10)
 1006 FORMAT('0LONGEST POSSIBLE PATH = ',F10.4/'0LONGEST ACTUAL PATH = '
     A,F10.4/'0AVE. PATH = ',F10.4)
 1007 FORMAT('0COMPUTED GEOMETRICAL FACTOR = ',1PE12.6)
 1008 FORMAT('0GEOMETRICAL FACTOR = ',1PE12.6/'0DETECTOR   BOTTOM    SI
     ADE    FRONT')
 1009 FORMAT(4X,I2,5X,3(I7,2X))
 1010 FORMAT('1 PATH-LENGTH  SIDECTS    FRTCTS    BTMCTS    TOTCTS',4X,
     A'EFFGF')
```

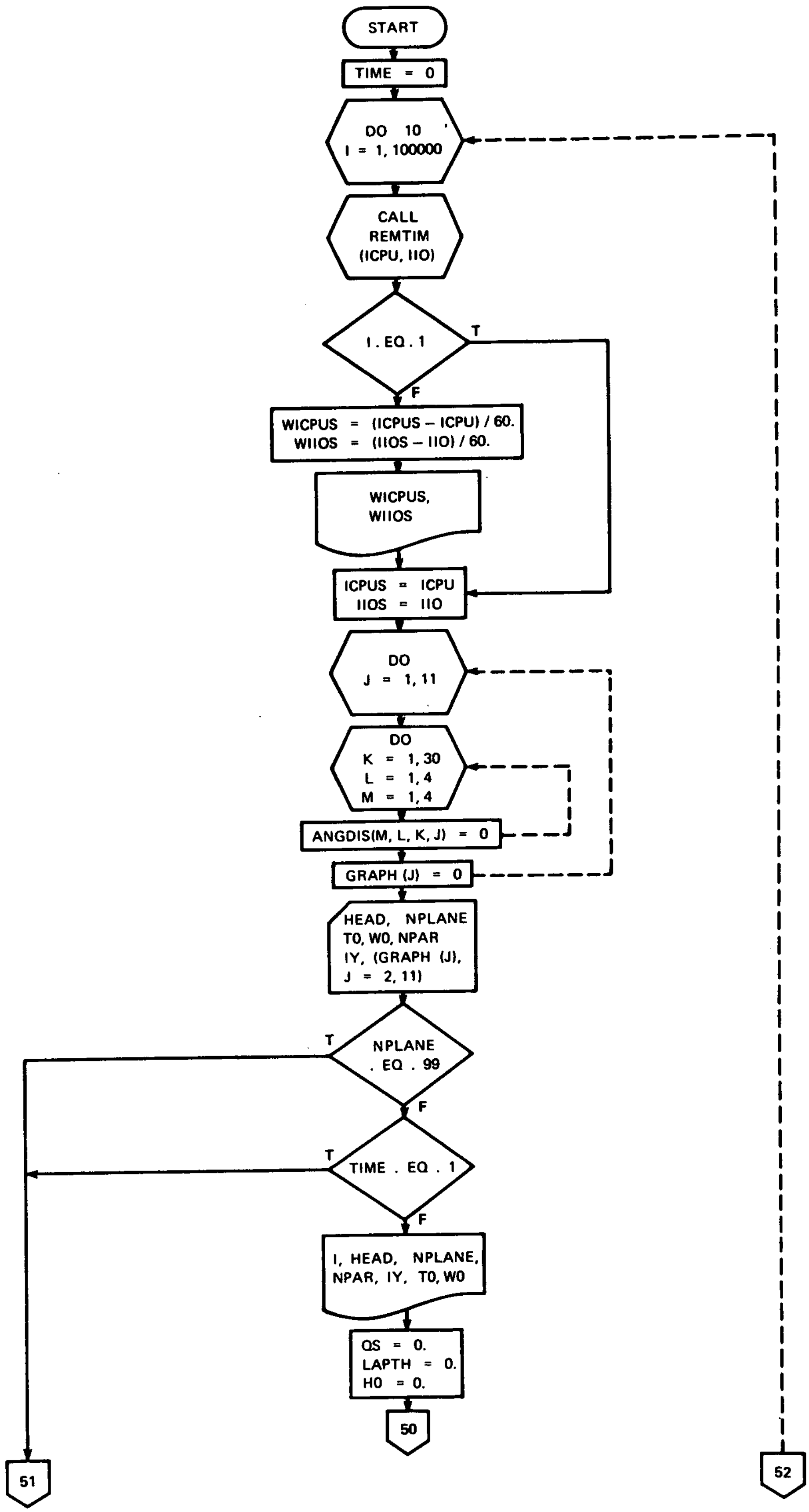
```
1011 FORMAT(1H ,F8.2,4I10,F13.2)
1012 FORMAT('1POLAR ANGLE       COUNTS OUT BOTTOM    COUNTS PER SQ-DEG')
1013 FORMAT(1H ,I7,14X,I6,12X,1PE12.6)
1014 FORMAT('1DISTURBUTION OF INCIDENT PARTICLES AT PLANE NO. ',I2/
    A'0TOTAL PARTICLES AT THIS PLANE = ',I10/'0BIN    X BIN    NO. OF PAR
    BTITCLES    Y BIN    NO. OF PARTICLES')
1015 FORMAT(I4,2(2X,F8.2,4X,I10,3X))
1016 FORMAT('1ANGULAR DISTURBUTION OF INCIDENT PARTICLES AT PLANE NO. '
    A,I2/'0      ',51X,'CORDINATES (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,I6),2X,I8)
1018 FORMAT('1ANGULAR DISTURBUTION AT 9. 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 ,I1,' *',4(1X,I7,1X),'*',13X,I1,' *',4(1X,I7,1X),'*')
     END
```

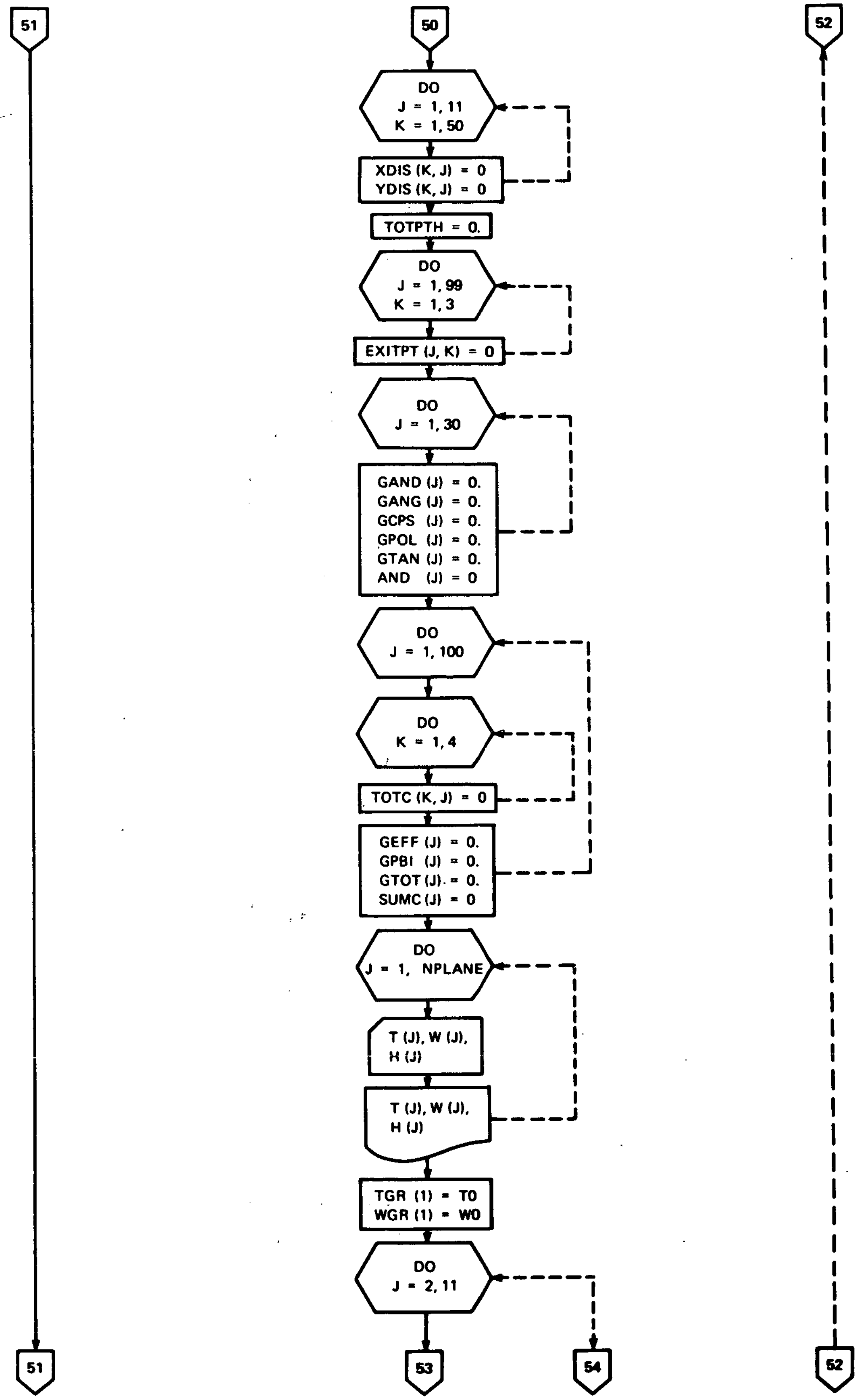
```
      SUBROUTINE G(TYPE,HEAD,A,B,SIZE)
      REAL*8 HEAD(5)
      LOGICAL*1 CWR(103)
      REAL ARAY(101),BARAY(101),A(SIZE),B(SIZE)
      INTEGER IARAY(101),IBARAY(101),QW,Q(11),TYPE,WA,EXPB,ADIF,BDIF,
     ATYP,SIZE,HIST
COMMENT   IF (HIST) EQUALS TO ONE THEN THIS SUBROUTINE MAKES HISTOGRAMS,
C         IF (HIST) DOSE NOT EQUAL TO ONE THEN THIS SUBROUTINE MAKES
C         GRAPHS.
      HIST=1
      J=101
      AMAX=0.
      BMAX=0.
      DO 10 I=1,101
      ARAY(I)=0.00001
      BARAY(I)=0.00001
      IF((I.LE.SIZE).AND.(A(I).GT..00001))ARAY(I)=A(I)
      IF((I.LE.SIZE).AND.(B(I).GT..00001))BARAY(I)=B(I)
   10 CONTINUE
      DO 11 I=1,101
      IF((ARAY(101-I).LE..00001).OR.(BARAY(101-I).LE..00001))GO TO 100
   11 J=101-I
  100 DO 12 I=1,J
      IF(ARAY(I).GT.AMAX)AMAX=ARAY(I)
      IF(BARAY(I).GT.BMAX)BMAX=BARAY(I)
   12 CONTINUE
      IF((AMAX.LE.5.).OR.(BMAX.LE.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(AMAX.GE.1.E5 )TYP =TYP +1
  106 GO TO (102,103,104,105),TYP
  102 TYP=1
      ADIF=AMAX/100+.995
      BDIF=BMAX/100+.995
      CWR(102)=64
      CWR(103)=64
      EXPB=1
      WRITE(6,1001)
      DO 13 I=1,J
      IARAY(I)=(ARAY(I)+ADIF/2.)/ADIF+1.
   13 IBARAY(I)=(BARAY(I)+BDIF/2.)/BDIF+1
      GO TO 107
  103 TYP=2
      ADIF=1
      BDIF=BMAX/100+.995
      CWR(102)=64
      EXPB=1
      CWR(103)=197
```

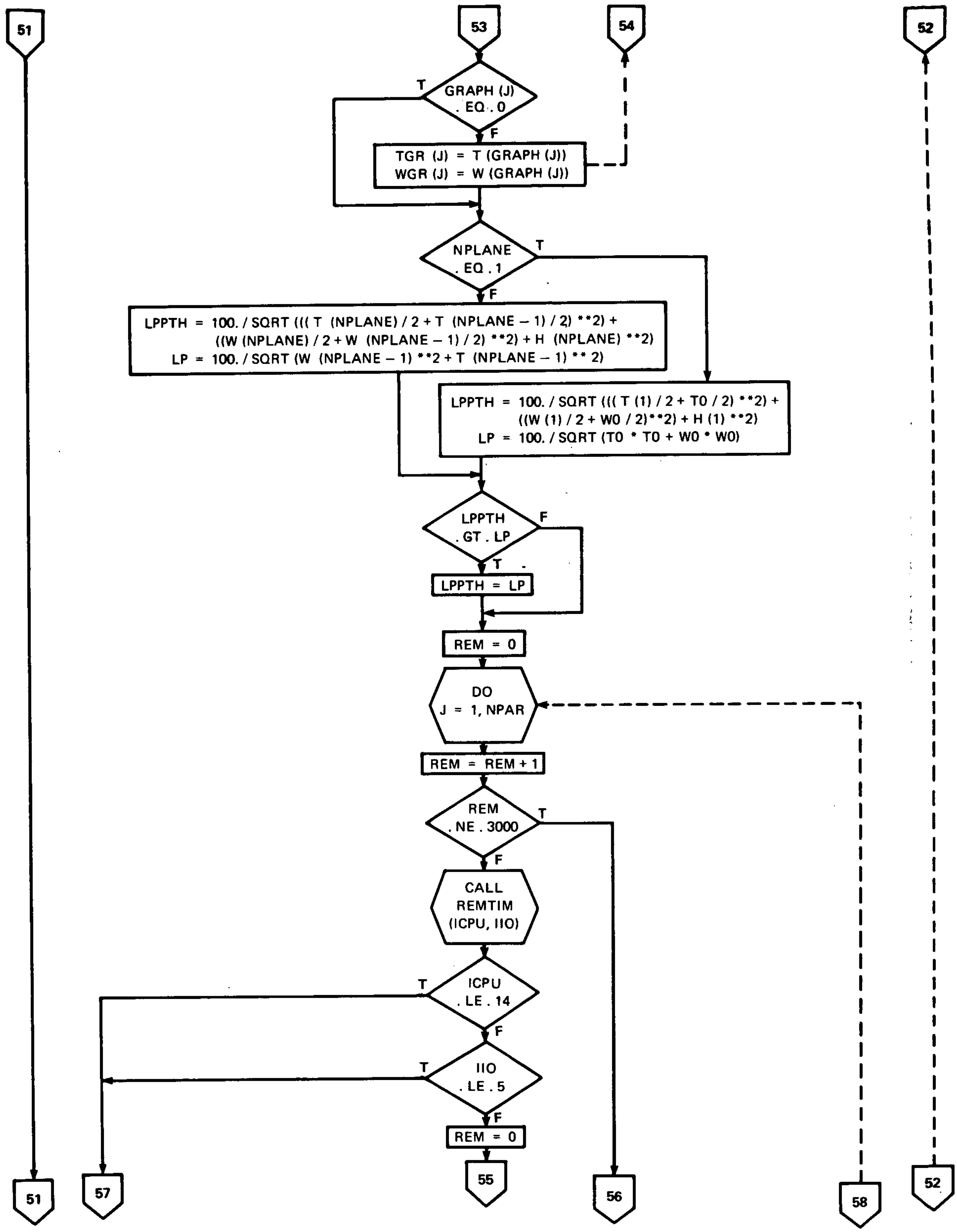
```
      WRITE(6,1002)
      DO 14 I=1,J
      IARAY(I)=(ALOG10(ARAY(I))+.05)*10+1
   14 IBARAY(I)=(BARAY(I)+BDIF/2.)/BDIF+1
      GO TO 107
  104 TYP=3
      ADIF=AMAX/100+.995
      BDIF=1
      CWR(102)=197
      CWR(103)=64
      EXPB=10
      WRITE(6,1003)
      DO 15 I=1,J
      IARAY(I)=(ARAY(I)+ADIF/2.)/ADIF+1.
   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 CWR(I)=64
  115 DO 17 I=1,101
      IF(HIST.EQ.1)GO TO 119
      DO 18 J=1,101
   18 CWR(J)=64
  119 WA =BDIF*(101-I)/EXPB
      DO 19 K=1,11
      IF(I.EQ.10*(K-1)+1) GO TO 113
   19 CONTINUE
      WRITE(6,1005)
      GO TO 114
  113 WRITE(6,1006)CWR(102),WA
  114 DO 20 J=1,101
      QW=0
      IF(IBARAY(J).EQ.102-I)QW=IARAY(J)
      IF(QW.GT.0)CWR(QW)=92
   20 CONTINUE
   17 WRITE(6,1007)(CWR(J),J=1,101)
      DO 21 I=1,11
      IF(TYP/2.EQ.TYP/2.)GO TO 116
      Q(I)=ADIF*(I-1)*10
      GO TO 21
```

```
 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,'I'    ,10(9X,'I'))
1006 FORMAT(1H ,A1,I7,101('-')/10H+         I,10(9X,'I'))
1007 FORMAT(9H+        ,101A1)
1008 FORMAT(1H ,5X,11(A1,I3,6X))
     END
```

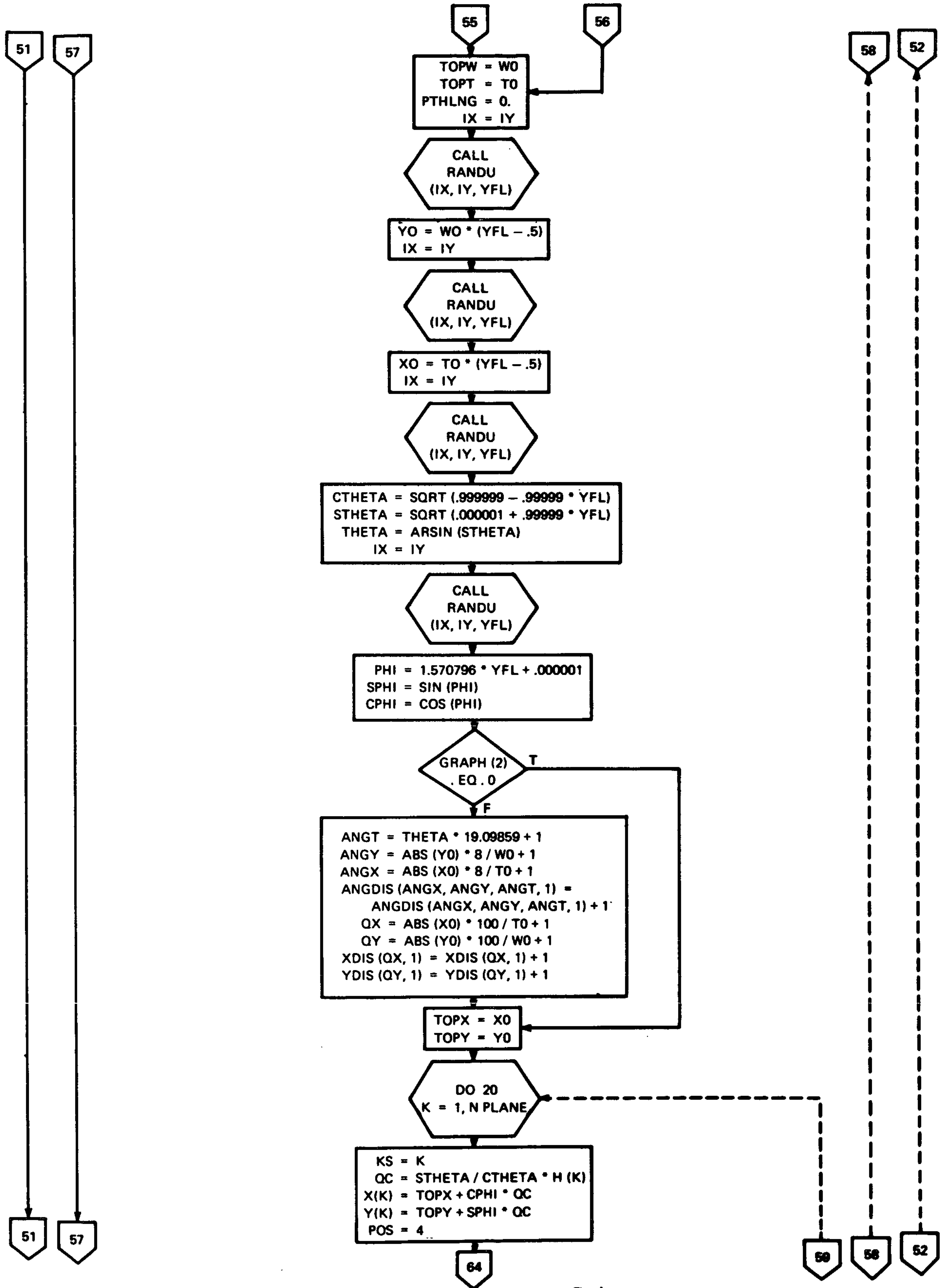
C. FLOW DIAGRAM OF THE GEOMETRICAL FACTOR PROGRAM


START
TIME = 0
DO 10
I = 1, 100000
CALL
REMTIM
(ICPU, IIO)
I . EQ . 1
T
F
WICPUS = (ICPUS – ICPU) / 60.
WIIOS = (IIOS – IIO) / 60.
WICPUS,
WIIOS
ICPUS = ICPU
IIOS = IIO
DO
J = 1, 11
DO
K = 1, 30
L = 1, 4
M = 1, 4
ANGDIS(M, L, K, J) = 0
GRAPH (J) = 0
HEAD, NPLANE
T0, W0, NPAR
IY, (GRAPH (J),
J = 2, 11)
NPLANE
. EQ . 99
T
F
TIME . EQ . 1
T
F
I, HEAD, NPLANE,
NPAR, IY, T0, W0
QS = 0.
LAPTH = 0.
H0 = 0.
50
51
52

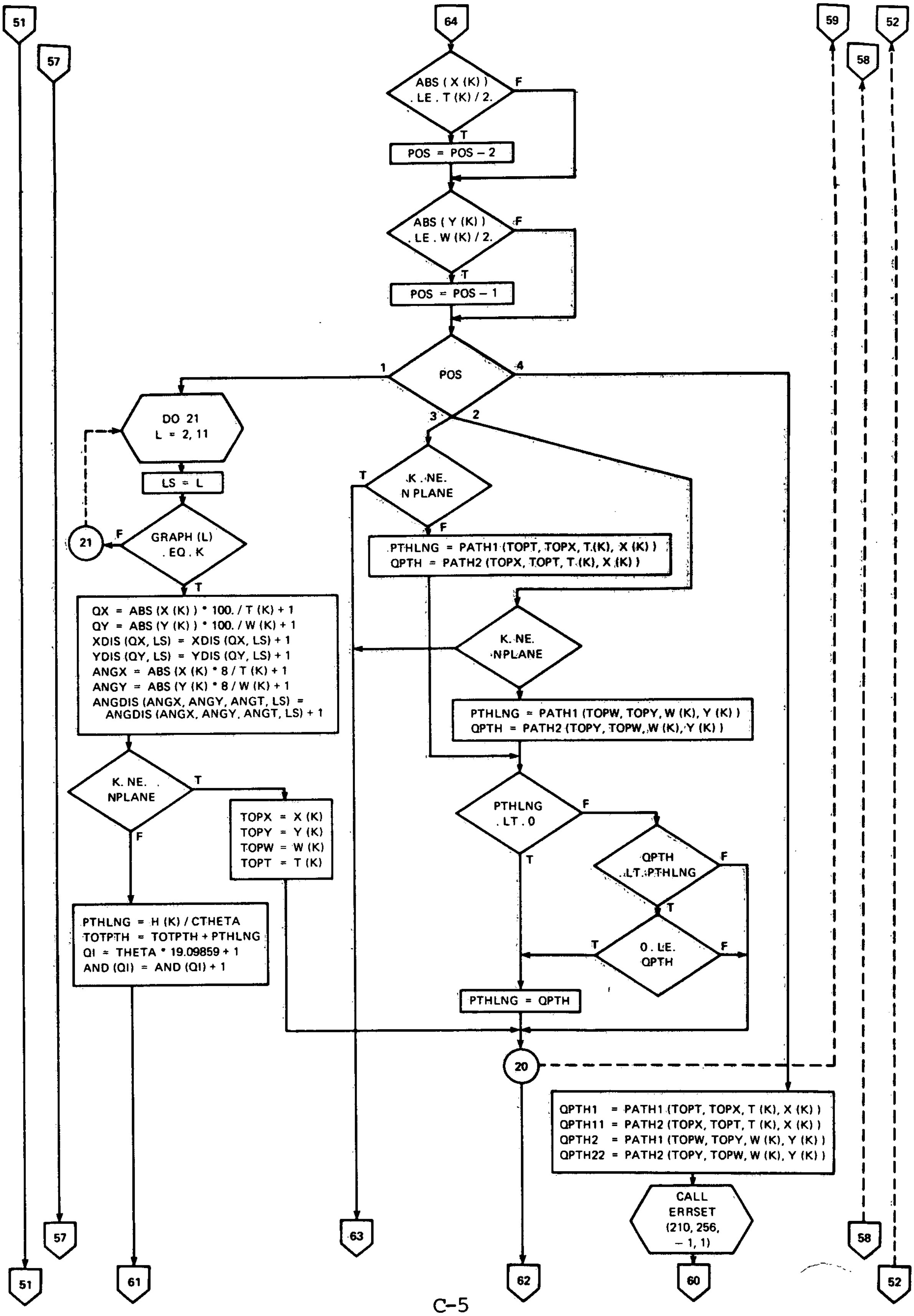

51
50
52
DO
J = 1, 11
K = 1, 50
XDIS (K, J) = 0
YDIS (K, J) = 0
TOTPTH = 0.
DO
J = 1, 99
K = 1, 3
EXITPT (J, K) = 0
DO
J = 1, 30
GAND (J) = 0.
GANG (J) = 0.
GCPS (J) = 0.
GPOL (J) = 0.
GTAN (J) = 0.
AND (J) = 0
DO
J = 1, 100
DO
K = 1, 4
TOTC (K, J) = 0
GEFF (J) = 0.
GPBI (J) = 0.
GTOT (J) = 0.
SUMC (J) = 0
DO
J = 1, NPLANE
T (J), W (J),
H (J)
T (J), W (J),
H (J)
TGR (1) = T0
WGR (1) = W0
DO
J = 2, 11
51
53
54
52

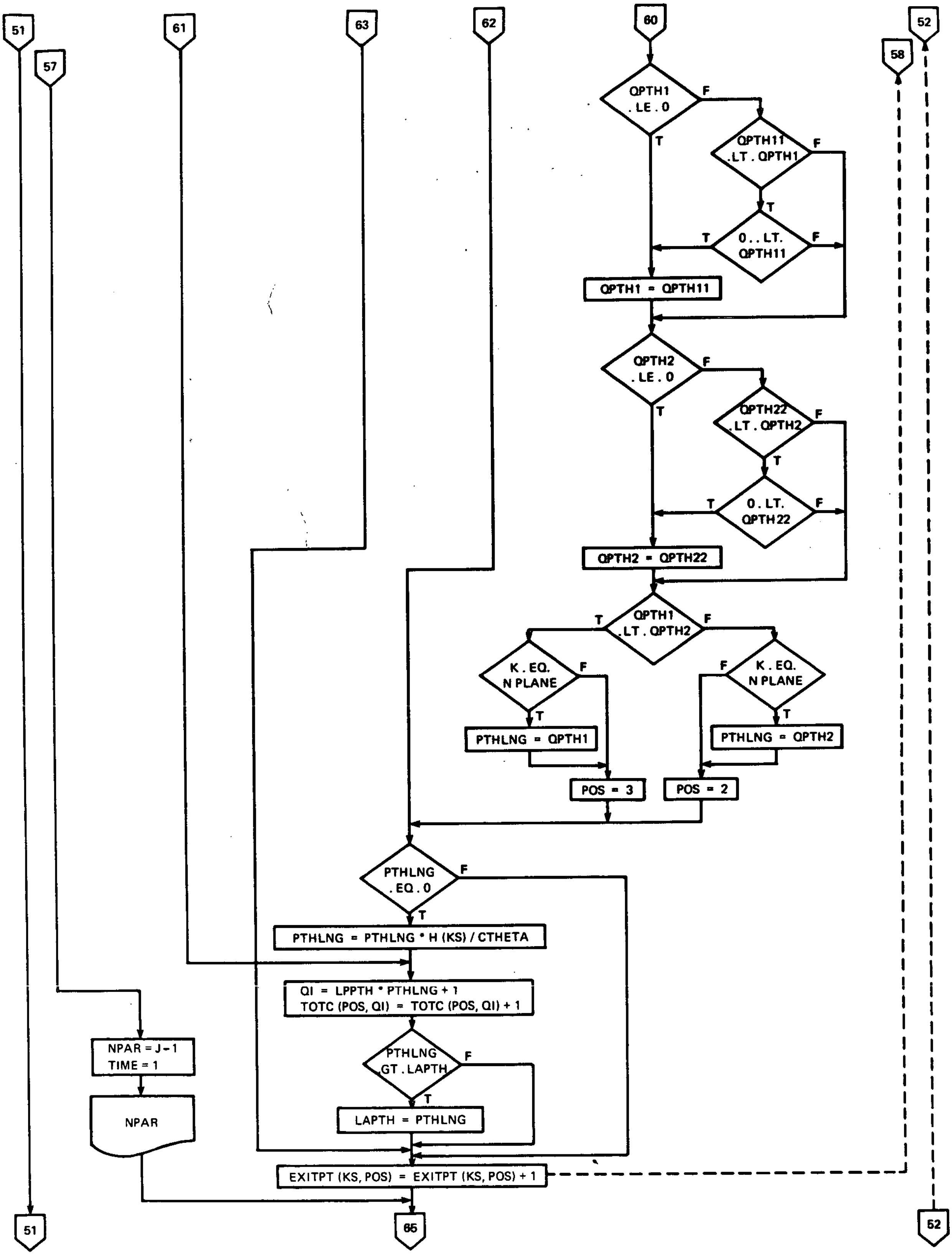


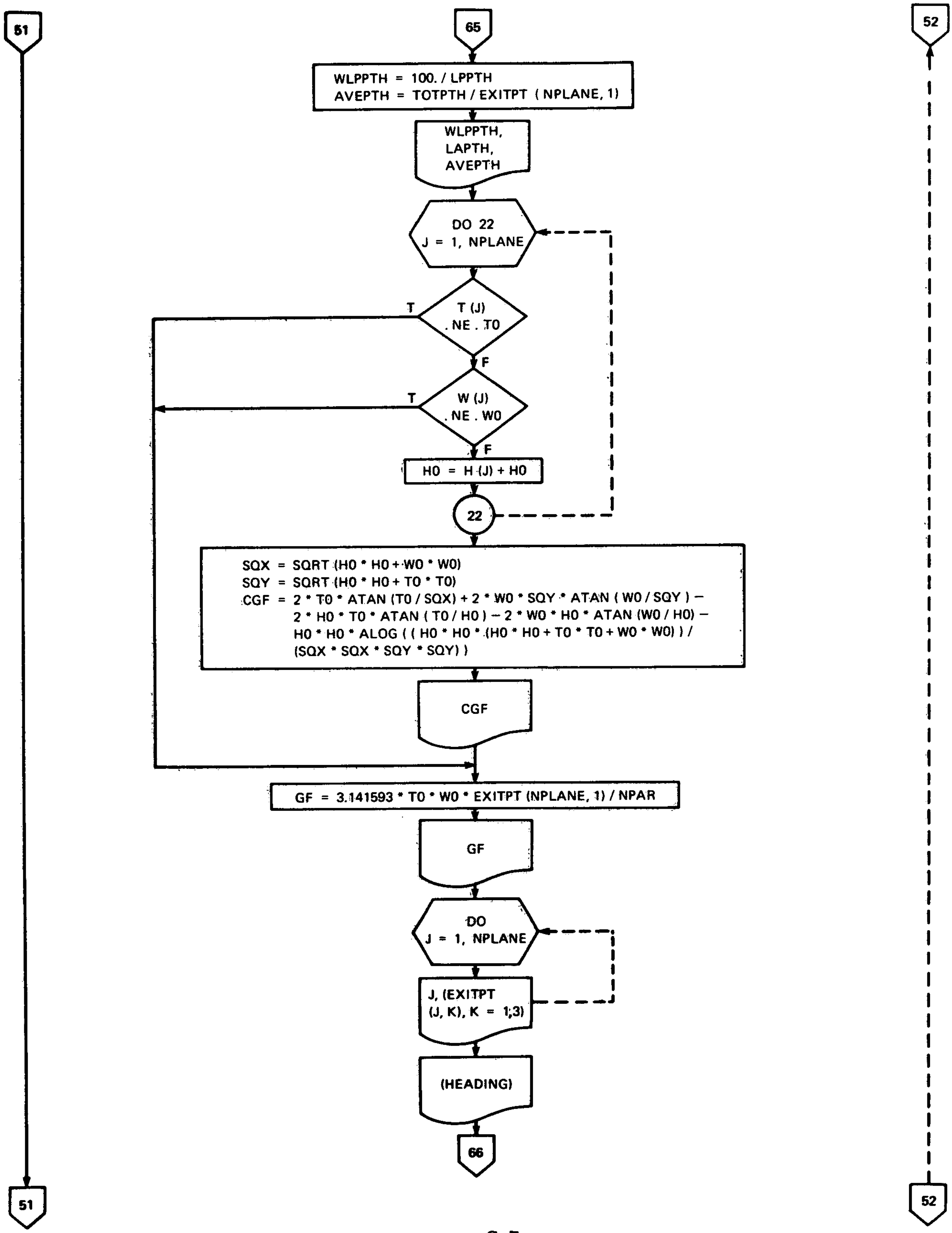
51
53
54
52
GRAPH (J) . EQ . 0
T
F
TGR (J) = T (GRAPH (J))
WGR (J) = W (GRAPH (J))
NPLANE . EQ . 1
T
F
LPPTH = 100. / SQRT (((T (NPLANE) / 2 + T (NPLANE – 1) / 2) **2) +
((W (NPLANE) / 2 + W (NPLANE – 1) / 2) **2) + H (NPLANE) **2)
LP = 100. / SQRT (W (NPLANE – 1) **2 + T (NPLANE – 1) ** 2)
LPPTH = 100. / SQRT (((T (1) / 2 + T0 / 2) **2) +
((W (1) / 2 + W0 / 2)**2) + H (1) **2)
LP = 100. / SQRT (T0 * T0 + W0 * W0)
LPPTH . GT . LP
F
T
LPPTH = LP
REM = 0
DO J = 1, NPAR
REM = REM + 1
REM . NE . 3000
T
F
CALL REMTIM (ICPU, IIO)
ICPU . LE . 14
T
F
IIO . LE . 5
T
F
REM = 0
51
57
55
56
58
52

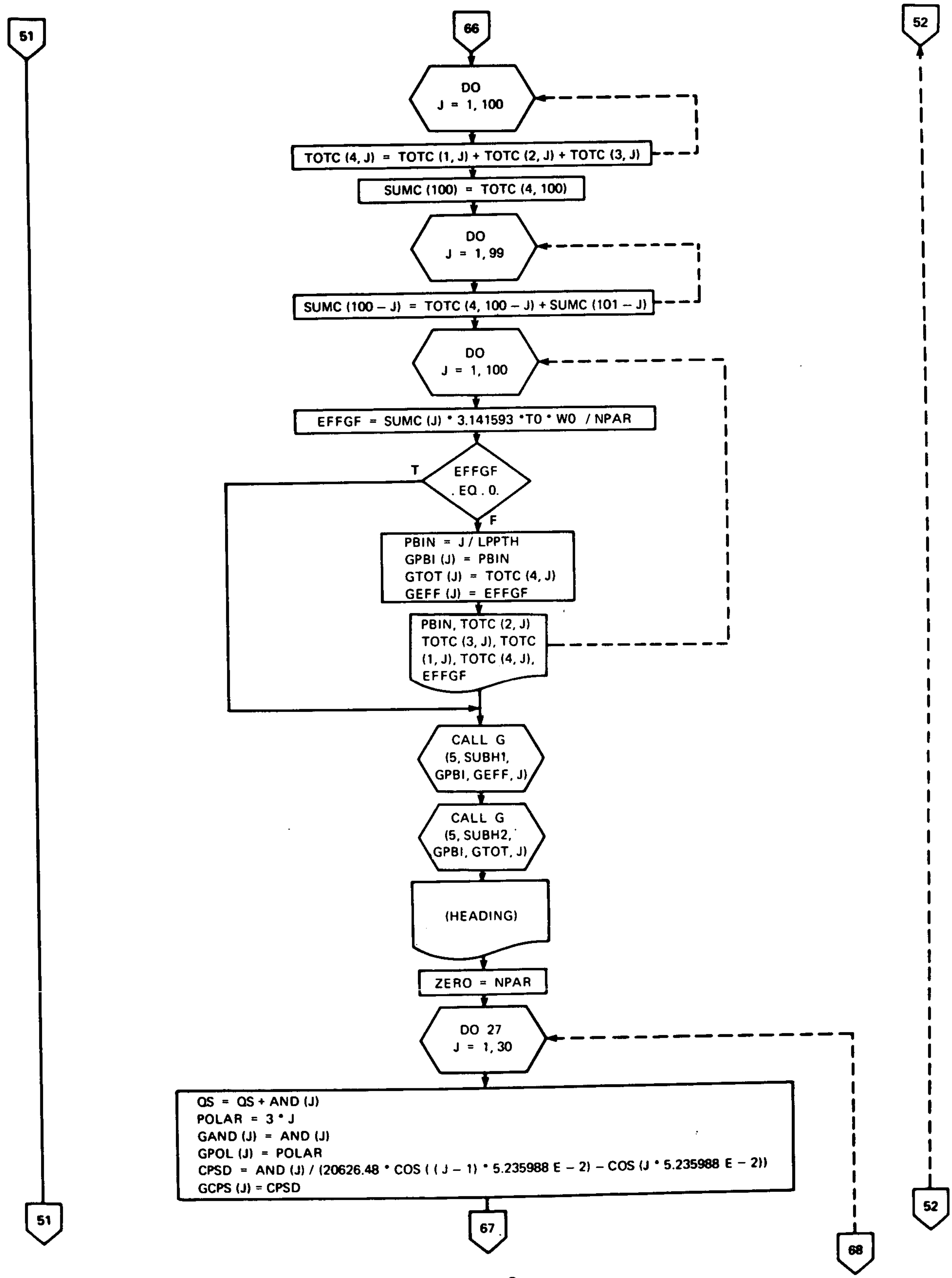


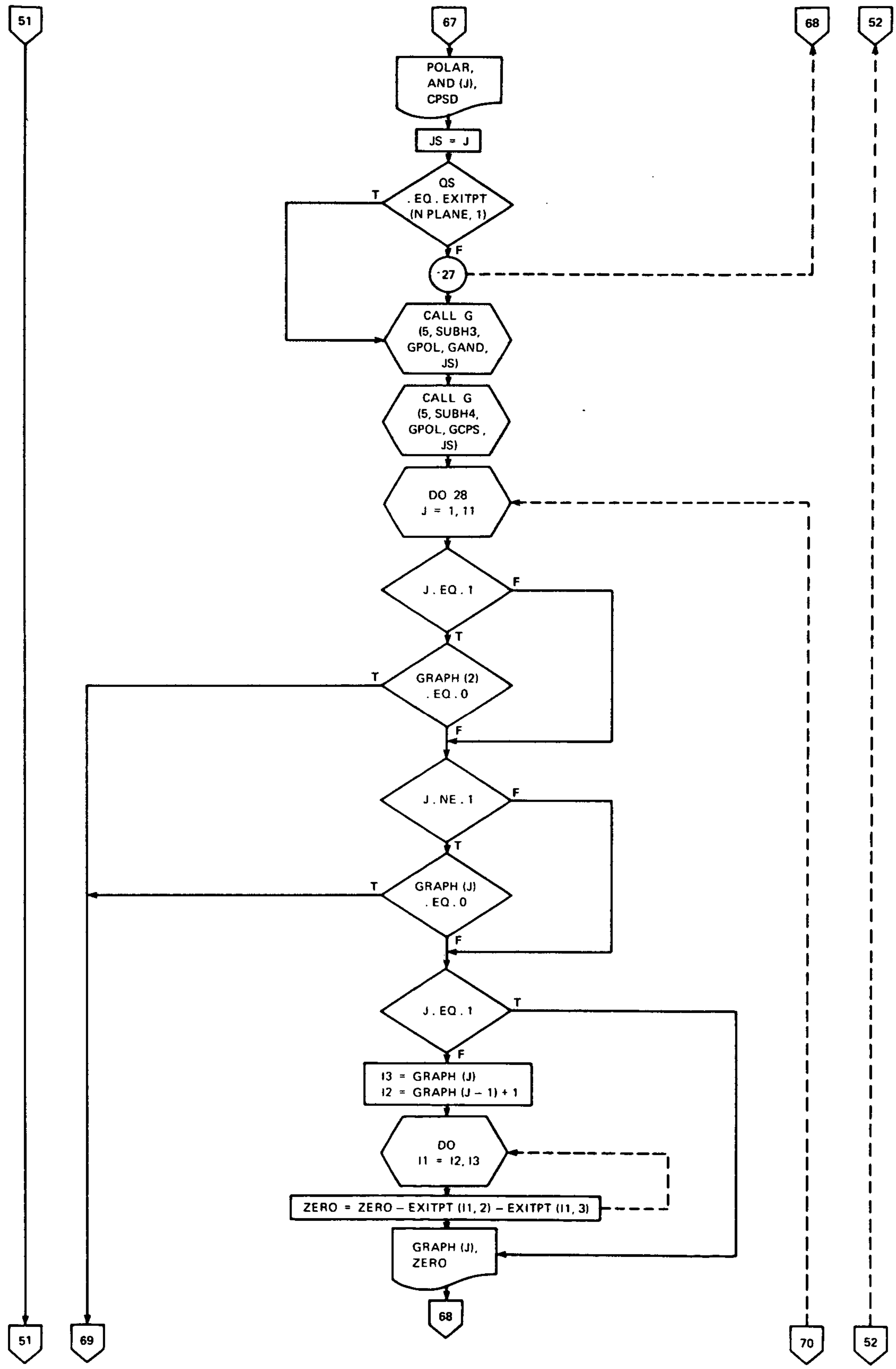
55
56
51
57
58
52
TOPW = W0
TOPT = T0
PTHLNG = 0.
IX = IY
CALL RANDU (IX, IY, YFL)
Y0 = W0 * (YFL – .5)
IX = IY
CALL RANDU (IX, IY, YFL)
X0 = T0 * (YFL – .5)
IX = IY
CALL RANDU (IX, IY, YFL)
CTHETA = SQRT (.999999 – .99999 * YFL)
STHETA = SQRT (.000001 + .99999 * YFL)
THETA = ARSIN (STHETA)
IX = IY
CALL RANDU (IX, IY, YFL)
PHI = 1.570796 * YFL + .000001
SPHI = SIN (PHI)
CPHI = COS (PHI)
GRAPH (2) . EQ . 0
T
F
ANGT = THETA * 19.09859 + 1
ANGY = ABS (Y0) * 8 / W0 + 1
ANGX = ABS (X0) * 8 / T0 + 1
ANGDIS (ANGX, ANGY, ANGT, 1) =
ANGDIS (ANGX, ANGY, ANGT, 1) + 1
QX = ABS (X0) * 100 / T0 + 1
QY = ABS (Y0) * 100 / W0 + 1
XDIS (QX, 1) = XDIS (QX, 1) + 1
YDIS (QY, 1) = YDIS (QY, 1) + 1
TOPX = X0
TOPY = Y0
DO 20 K = 1, N PLANE
KS = K
QC = STHETA / CTHETA * H (K)
X(K) = TOPX + CPHI * QC
Y(K) = TOPY + SPHI * QC
POS = 4
51
57
59
58
52
64

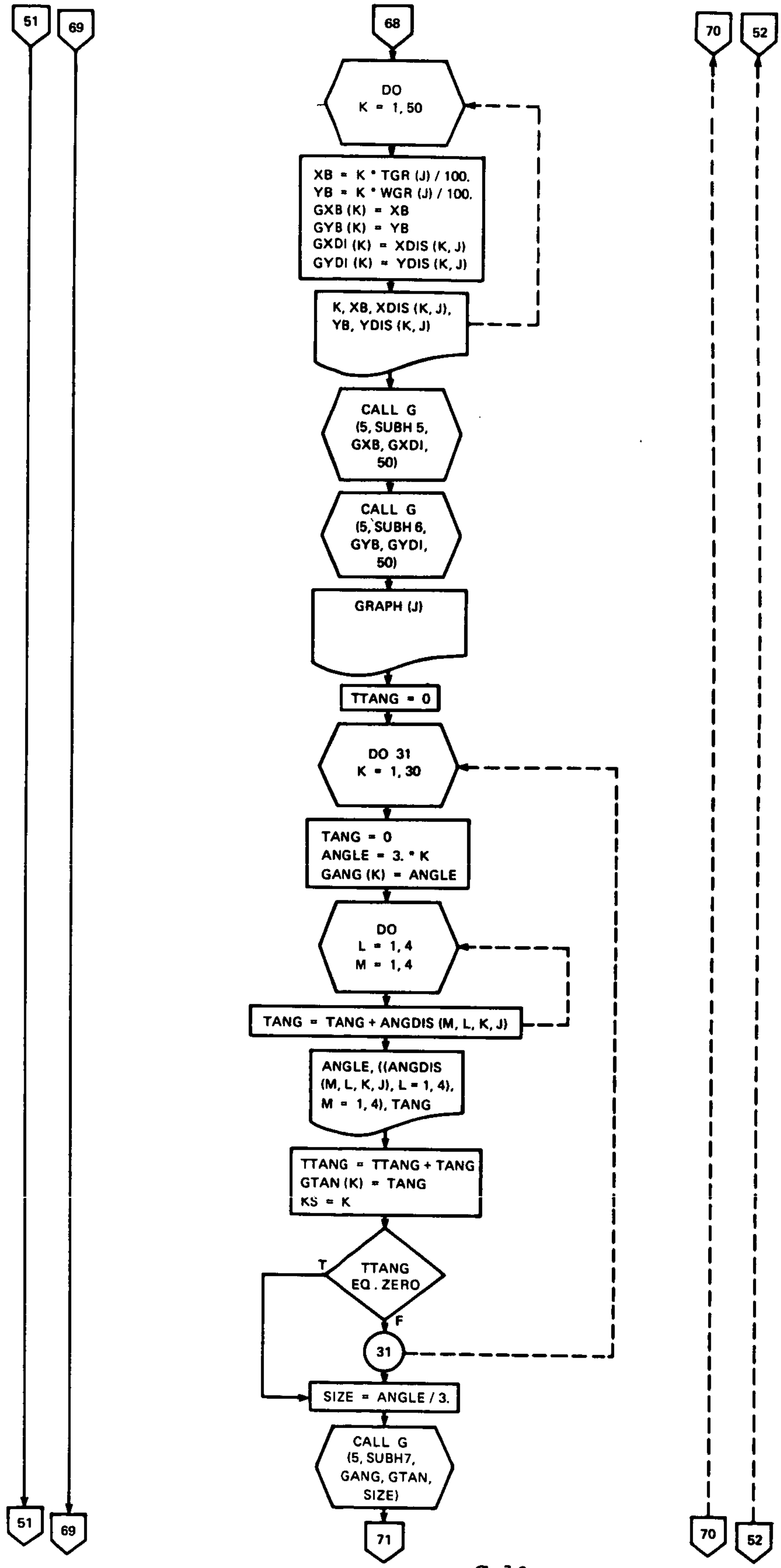

51
57
64
59
52
58
ABS (X (K)) . LE . T (K) / 2.
POS = POS – 2
ABS (Y (K)) . LE . W (K) / 2.
POS = POS – 1
POS
DO 21 L = 2, 11
LS = L
GRAPH (L) . EQ . K
21
QX = ABS (X (K)) * 100. / T (K) + 1
QY = ABS (Y (K)) * 100. / W (K) + 1
XDIS (QX, LS) = XDIS (QX, LS) + 1
YDIS (QY, LS) = YDIS (QY, LS) + 1
ANGX = ABS (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
K. NE. NPLANE
TOPX = X (K)
TOPY = Y (K)
TOPW = W (K)
TOPT = T (K)
PTHLNG = H (K) / CTHETA
TOTPTH = TOTPTH + PTHLNG
QI = THETA * 19.09859 + 1
AND (QI) = AND (QI) + 1
K . NE. N PLANE
PTHLNG = PATH1 (TOPT, TOPX, T (K), X (K))
QPTH = PATH2 (TOPX, TOPT, T (K), X (K))
K. NE. NPLANE
PTHLNG = PATH1 (TOPW, TOPY, W (K), Y (K))
QPTH = PATH2 (TOPY, TOPW, W (K), Y (K))
PTHLNG . LT . 0
QPTH . LT . PTHLNG
0 . LE. QPTH
PTHLNG = QPTH
20
QPTH1 = PATH1 (TOPT, TOPX, T (K), X (K))
QPTH11 = PATH2 (TOPX, TOPT, T (K), X (K))
QPTH2 = PATH1 (TOPW, TOPY, W (K), Y (K))
QPTH22 = PATH2 (TOPY, TOPW, W (K), Y (K))
CALL ERRSET (210, 256, – 1, 1)
51
57
61
63
62
60
58
52

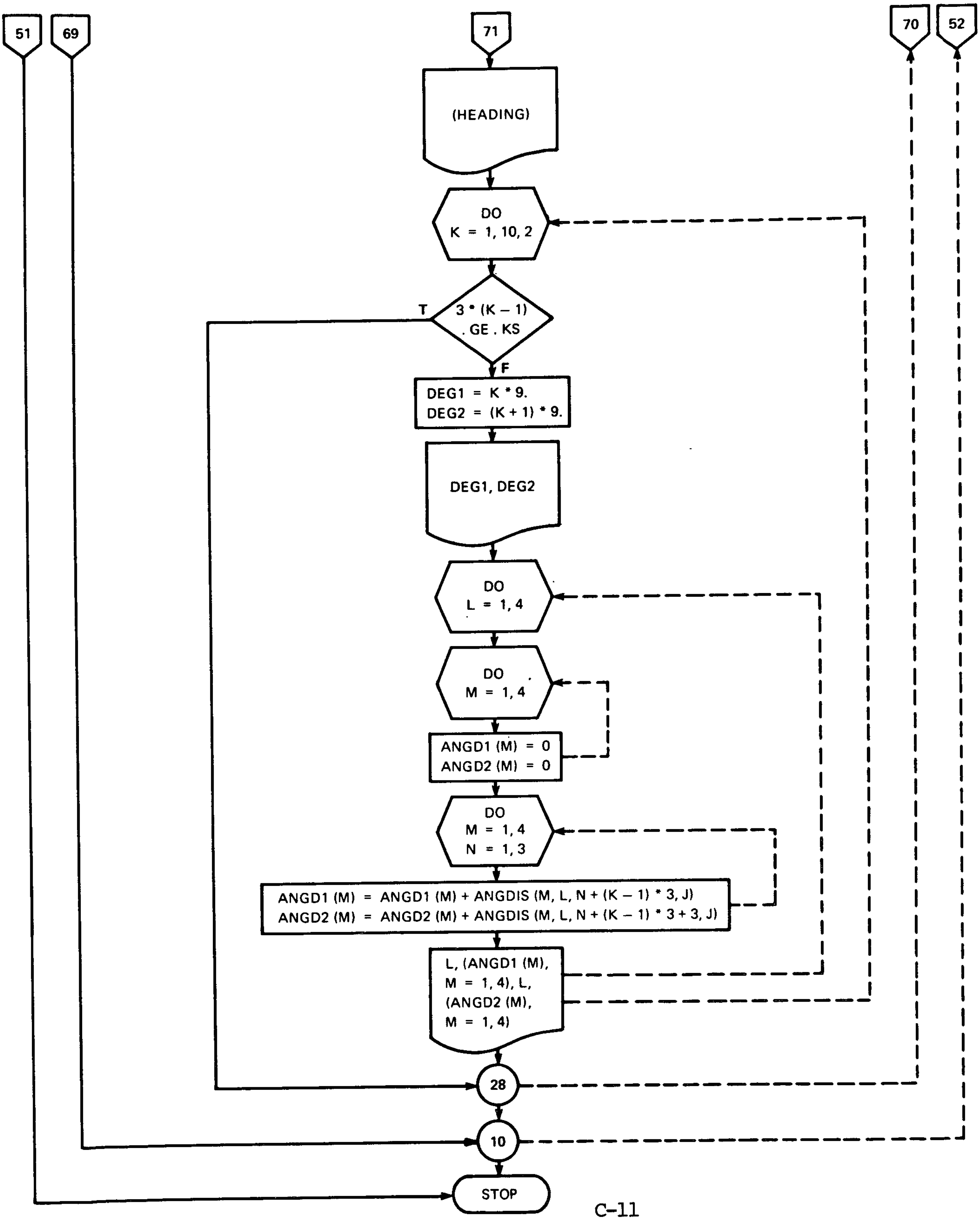

51
61
63
62
60
52
57
58
QPTH1 .LE. 0
QPTH11 .LT. QPTH1
0..LT. QPTH11
QPTH1 = QPTH11
QPTH2 .LE. 0
QPTH22 .LT. QPTH2
0. LT. QPTH22
QPTH2 = QPTH22
QPTH1 .LT. QPTH2
K. EQ. N PLANE
K. EQ. N PLANE
PTHLNG = QPTH1
PTHLNG = QPTH2
POS = 3
POS = 2
PTHLNG .EQ. 0
PTHLNG = PTHLNG * H (KS) / CTHETA
QI = LPPTH * PTHLNG + 1
TOTC (POS, QI) = TOTC (POS, QI) + 1
PTHLNG .GT. LAPTH
LAPTH = PTHLNG
NPAR = J-1
TIME = 1
NPAR
EXITPT (KS, POS) = EXITPT (KS, POS) + 1
51
65
52


51
65
52
WLPPTH = 100. / LPPTH
AVEPTH = TOTPTH / EXITPT (NPLANE, 1)
WLPPTH,
LAPTH,
AVEPTH
DO 22
J = 1, NPLANE
T
T (J)
. NE . T0
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 * SQY * 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
51
52


51
66
52
DO
J = 1, 100
TOTC (4, J) = TOTC (1, J) + TOTC (2, J) + TOTC (3, J)
SUMC (100) = TOTC (4, 100)
DO
J = 1, 99
SUMC (100 – J) = TOTC (4, 100 – J) + SUMC (101 – J)
DO
J = 1, 100
EFFGF = SUMC (J) * 3.141593 *T0 * W0 / NPAR
T
EFFGF
. EQ . 0.
F
PBIN = J / LPPTH
GPBI (J) = PBIN
GTOT (J) = TOTC (4, J)
GEFF (J) = EFFGF
PBIN, TOTC (2, J)
TOTC (3, J), TOTC
(1, J), TOTC (4, J),
EFFGF
CALL G
(5, SUBH1,
GPBI, GEFF, J)
CALL G
(5, SUBH2,
GPBI, GTOT, J)
(HEADING)
ZERO = NPAR
DO 27
J = 1, 30
QS = QS + AND (J)
POLAR = 3 * J
GAND (J) = AND (J)
GPOL (J) = POLAR
CPSD = AND (J) / (20626.48 * COS ((J – 1) * 5.235988 E – 2) – COS (J * 5.235988 E – 2))
GCPS (J) = CPSD
51
67
68
52


51
67
68
52
POLAR,
AND (J),
CPSD
JS = J
QS
. EQ . EXITPT
(N PLANE, 1)
T
F
27
CALL G
(5, SUBH3,
GPOL, GAND,
JS)
CALL G
(5, SUBH4,
GPOL, GCPS ,
JS)
DO 28
J = 1, 11
J . EQ . 1
GRAPH (2)
. EQ . 0
J . NE . 1
GRAPH (J)
. EQ . 0
J . EQ . 1
I3 = GRAPH (J)
I2 = GRAPH (J – 1) + 1
DO
I1 = I2, I3
ZERO = ZERO – EXITPT (I1, 2) – EXITPT (I1, 3)
GRAPH (J),
ZERO
68
51
69
70
52


51
69
68
70
52
DO
K = 1, 50
XB = K * TGR (J) / 100.
YB = K * WGR (J) / 100.
GXB (K) = XB
GYB (K) = YB
GXDI (K) = XDIS (K, J)
GYDI (K) = YDIS (K, J)
K, XB, XDIS (K, J),
YB, YDIS (K, J)
CALL G
(5, SUBH 5,
GXB, GXDI,
50)
CALL G
(5, SUBH 6,
GYB, GYDI,
50)
GRAPH (J)
TTANG = 0
DO 31
K = 1, 30
TANG = 0
ANGLE = 3. * K
GANG (K) = ANGLE
DO
L = 1, 4
M = 1, 4
TANG = TANG + ANGDIS (M, L, K, J)
ANGLE, ((ANGDIS
(M, L, K, J), L = 1, 4),
M = 1, 4), TANG
TTANG = TTANG + TANG
GTAN (K) = TANG
KS = K
T
TTANG
EQ . ZERO
F
31
SIZE = ANGLE / 3.
CALL G
(5, SUBH7,
GANG, GTAN,
SIZE)
51
69
71
70
52


51
69
71
70
52
(HEADING)
DO
K = 1, 10, 2
3 * (K – 1)
. GE . KS
T
F
DEG1 = K * 9.
DEG2 = (K + 1) * 9.
DEG1, DEG2
DO
L = 1, 4
DO
M = 1, 4
ANGD1 (M) = 0
ANGD2 (M) = 0
DO
M = 1, 4
N = 1, 3
ANGD1 (M) = ANGD1 (M) + ANGDIS (M, L, N + (K – 1) * 3, J)
ANGD2 (M) = ANGD2 (M) + ANGDIS (M, L, N + (K – 1) * 3 + 3, J)
L, (ANGD1 (M),
M = 1, 4), L,
(ANGD2 (M),
M = 1, 4)
28
10
STOP