OAK RIDGE NATIONAL LABORATORY

operated by
UNION CARBIDE CORPORATION
for the
U.S. ATOMIC ENERGY COMMISSION

MASTER

ORNL-TM-1950
(Rev. 1)

RECEIVED BY DTIC DEC 2 1970

PROGRAM WEIGHT: A FORTRAN IV PROGRAM FOR EVALUATION OF WEIGHTING
FUNCTIONS USED IN SMALL-ANGLE X-RAY SCATTERING

Mary G. Buchanan and Robert W. Hendricks

NOTICE This document contains information of a preliminary nature and was prepared primarily for internal use at the Oak Ridge National Laboratory. It is subject to revision or correction and therefore does not represent a final report.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

- A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or
- B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

Contract No. W-7405-eng-26

METALS AND CERAMICS DIVISION

PROGRAM WEIGHT: A FORTRAN IV PROGRAM FOR EVALUATION OF WEIGHTING
FUNCTIONS USED IN SMALL-ANGLE X-RAY SCATTERING

Mary G. Buchanan and Robert W. Hendricks

LEGAL NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

NOVEMBER 1970

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee
operated by
UNION CARBIDE CORPORATION
for the
U.S. ATOMIC ENERGY COMMISSION

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

PREFACE

Program WEIGHT: A FORTRAN IV Program For Evaluation of Weighting Functions Used in Small-Angle X-Ray Scattering was presented in its original form in September 1967 as ORNL-TM-1950. Since then, several errors have been discovered, both in the program and in the report. Experience over the past three years has also suggested several modifications which increase the usefulness of program WEIGHT. Among these additions are (1) an improved format for the output of the computed results, (2) a more extensive error checking procedure which prevents program failure in the event of certain unusual collimation conditions, (3) a plotting package for the CALCOMP plotters, (4) a timing routine which computes the CPU seconds for each weighting function, and (5) computation of the theoretical limit of resolution of the collimation system under consideration.

This present report supercedes the original report, which should be discarded. Source decks of the new program will be provided, on request, by R. W. Hendricks.

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

CONTENTS

	<u>Page</u>
Abstract	1
Introduction	1
Program Description	4
Detailed Operating Instructions	7
Acknowledgments	8
Appendix A. Program WEIGHT FORTRAN IV Listing	11
Appendix B. Flow Chart of Program WEIGHT	33
Appendix C. Definition of Selected Program Variables	45
Appendix D. Input Variables	51
Appendix E. Sample Calculations	57

PROGRAM WEIGHT: A FORTRAN IV PROGRAM FOR EVALUATION OF WEIGHTING
FUNCTIONS USED IN SMALL-ANGLE X-RAY SCATTERING

Mary G. Buchanan¹ and Robert W. Hendricks

ABSTRACT

A FORTRAN IV program is presented which evaluates the weighting functions needed to correct observed small-angle x-ray scattering data for the smearing effects of the collimation system. Moments of the weighting functions about both the origin and the mean may also be computed.

INTRODUCTION

In a previous paper, Hendricks and Schmidt² have derived a general equation for evaluating the slit-width and slit-length weighting functions which are necessary for correcting small-angle x-ray scattering data for the smearing effects of the collimation system. Their result, for slit widths, is

$$W_w(u) = L_U L(A_w(L_U + L))^{-1} \int_{x_a(u)}^{x_b(u)} dx e_w(x) \int_{z_a(x,u)}^{z_b(x,u)} dz f_w(z) \quad (1)$$

The limits of integration are given by

$$\begin{aligned} z_b(x,u) &= \text{Min}(z_{2i-1}; z_{2j-1}) \\ z_a(x,u) &= \text{Max}(z_{2i}; z_{2j}) \end{aligned} \quad (2a)$$

¹Formerly at Computing Technology Center, Oak Ridge Gaseous Diffusion Plant, now at Data Processing Center, E. I. DuPont de Nemours, Spruance Fibers Division, Richmond, Virginia.

²R. W. Hendricks and P. W. Schmidt, "Calculation of Weighting Functions for Collimation Corrections in Small-Angle X-Ray Scattering," Acta Phys. Austriaca 26, 97-122 (1967).

in which

$$z_i = \frac{(L_o - d_i)(Lu - x) + (-1)^{i+1}(L + L_o)h_i}{L + d_i} \quad (2b)$$

$$z_j = L_o u - \frac{(L_o + d_j)x + (-1)^j(L + L_o)h_j}{L - d_j},$$

and

$$x_b(u) = \text{Min}(x_{2i,2i'-1}; x_{2i,2j-1}; w_c) \quad (3)$$

$$x_a(u) = \text{Max}(x_{2i-1,2i'}; x_{2i-1,2j}; -w_c)$$

with

$$x_{ii'} = \frac{L(d_i - d_{i'})u + (-1)^i h_i(L + d_{i'}) + (-1)^{i'+1} h_{i'}(L + d_i)}{d_i - d_{i'}}$$

$$x_{ij} = \frac{(L - d_j)d_i u + (-1)^i h_i(L - d_j) + (-1)^{j+1} h_j(L + d_i)}{d_i + d_j}.$$

The weighting function is nonzero only in the range

$$u_{aaa} \leq u \leq u_{bbb} \quad (4)$$

where

$$u_{bbb} = \text{Min}[u_{2i-1,2i'}(w_c); u_{2i-1,2j}(w_c)] \quad (5)$$

$$u_{aaa} = \text{Max}[u_{2i,2i'-1}(-w_c); u_{2i,2j-1}(-w_c)]$$

$$u_{ii'} = \frac{x}{L} + \frac{(-1)^{i+1} h_i(L + d_{i'}) + (-1)^{i'+1} h_{i'}(L + d_i)}{L(d_i - d_{i'})}$$

and

$$u_{ij} = \frac{x(d_i + d_j) + (-1)^{i+1} h_i(L - d_j) + (-1)^j h_j(L + d_i)}{d_i(L - d_j)}.$$

In this derivation, the variables have the following meanings: L is the distance from the sample plane to the detector plane, L_0 is the distance from the sample plane to the tube plane, d_i is the distance from the sample plane to the i^{th} collimation edge between the sample plane and the tube plane, h_i is the distance from the i^{th} collimation edge to the camera axis, and d_j and h_j are similarly defined for edges between the sample plane and the detector plane (not including the slit at the detector). The width of the detector slit is $2w_c$, $e_w(x)$ describes the sensitivity of the detector in a direction parallel to the slit width, $f_w(z)$ describes the intensity of the focal spot in the direction of the slit-width, and A_w is a normalization constant which ensures that the area under the weighting function is unity. The weighting function for slit-lengths has the same form and can be evaluated from Eqs. (1) through (5) by making the appropriate changes in notation.

In the original derivation, the subscripts i and j took on all values in the ranges $1 \leq i \leq m$ and $1 \leq j \leq n$, respectively, subject to the condition $i' \leq i$. Here m is the number of pairs of edges between the sample plane and the tube plane, while n is the number of pairs of edges between the sample plane and the detector plane. In that derivation it is not necessary that each member of a pair of edges be equidistance from the sample plane. This leads to the possibility that the denominator in the equations for $x_{ii'}$ and $u_{ii'}$ could be zero for some values of i and i' in their allowable range. For hand calculations, division by zero produces an infinity which is easily handled. However, for computer evaluation, such division must be avoided. Careful examination of the original derivation reveals that if the edges occur in pairs in which each member of the pair is equidistant from the sample plane³ the restriction on i and i' becomes $i' < i$. It is easily verified that under these conditions, the denominators in $x_{ii'}$ and $u_{ii'}$ can never be zero.

³This condition in no way implies that the computation applies only to camera geometries which have real edges which occur in pairs equidistant from the sample. If, as in the Kratky camera, a real edge exists only on one side of the reference line, a dummy edge which is a large distance from the reference line is inserted on the other side of the reference line at the same distance from the sample plane. Such a dummy edge in no way alters the physical situation, but it does meet the mathematical requirements of the equations, and simplifies the computational procedure.

The slit-width collimation correction procedures outlined by Kratky, Porod, and Skala,⁴ and Sauder,⁵ require only knowledge of the k^{th} moment of the weighting function about either the origin or the mean. These moments are readily computed from the relationships

$$M_k = \int_{u_{aaa}}^{u_{bbb}} u^k w_w(u) du$$

and

(6)

$$M_k(\bar{u}) = \int_{u_{aaa}}^{u_{bbb}} (u - \bar{u})^k w_w(u) du .$$

The practical evaluation of Eqs. (1) and (6) is tedious except for geometries having a high degree of symmetry. However, they are ideally suited to computer evaluation. This report describes a FORTRAN IV program which may be used to calculate the weighting function and its moments for small-angle x-ray cameras having a variety of geometries. The reader is referred to the original derivation for a complete discussion of the approximations which limits the validity of the application of Eq. (1) and hence the usefulness of this program. A punched deck for the program may be obtained from R. W. Hendricks.

PROGRAM DESCRIPTION

The object of the program, for which a listing is given in Appendix A, is to evaluate the double integral described by Eq. (1). The integrations are performed by using Simpson's rule for evenly spaced intervals. However, the major portion of the program deals with the calculation of the limits of the integration from Eqs. (2) and (3). Appendix B is a flow chart which explains the operation of the program. Selected program variables are defined in Appendix C, while a complete listing of all variable

⁴O. Kratky, G. Porod, and Z. Skala, Acta Phys. Austriaca 13, 96 (1960).

⁵W. C. Sauder, J. Appl. Phys. 37, 1495 (1966).

names used in each subroutine as prepared by FORTDOC⁶ is given in Appendix A. In addition, Appendix A contains a complete input-output table for each subroutine which performs an input-output operation.

Before starting the main portion of the program, a short data format check is performed in loops 1 and 2 to ensure that the input data corresponds to pairs of edges, each member of which is equidistant from the sample plane. If this criterion is not met, an error message is presented in the output, and the computations are terminated.

Using Eqs. (4) and (5), the limits of integration u_{aaa} and u_{bbb} are computed by evaluating the appropriate values of $u_{ii}(\pm w_c)$ and $u_{ij}(\pm w_c)$ in loops 3 and 4 or loops 5 and 6, respectively, and comparing the results in SUBMIN and SUBMAX. Then the u increment is calculated as $(u_{bbb} - u_{aaa})/(NU - 1)$. The initial value of u is set equal to the lower limit, and each successive u up to the upper limit is calculated by adding the u increment to the preceding u value.

In essence, the remainder of the program consists of a series of nested loops. The outside loop (loop 7) calculates the u value. For each value of u , a pair of x limits is calculated in two pairs of inner loops. Loops 8 and 9 find $x_{ii}(u_\ell)$ while loops 10 and 11 find $x_{ij}(u_\ell)$. Values of $x_a(u_\ell)$ and $x_b(u_\ell)$ are found in SUBMAX and SUBMIN and the x values are incremented between these limits in the same manner as for u . In order to provide the same degree of accuracy in all of the integrations, the number of x increments (and z increments, to be described below) is the same as the number of u increments (NU). In turn, for each value of u and x a pair of z limits is calculated in loops 13 and 14. The value of z is incremented as before in loop 12, and the function $f_w(z)$ is integrated between these limits. If $f_w(z)$ is not a constant, the appropriate value is found from tabulated input data. It is readily seen that if a value of $f_w(z)$ is to be found for each value of z used in the program, $(NU)^3$ interpolations must be made, while $(NU)^2$ interpolations would be made to find the necessary values of $e_w(x)$. Since NU is of the order of 10^2 , this leads to an unreasonable number of interpolations and excessive machine time. In order to reduce this problem, an alternative procedure has been followed. Immediately upon reading in the $f_w(z)$ data,

⁶R. C. Durfee, private communication, July 1968.

a four point Lagrangian interpolation procedure^{7,8} generates a table consisting of $10*(NU - 1)$ equispaced values of $f_w(z)$ and z . Then a table called ZINT(I) is generated which contains a Simpson's Rule integration of $f_w(z)$ from $I = 1$ to $I = I$. With reference to Fig. 1, we find the integral

$$ZANS(MM) = \int_{z_a(x,u)}^{z_b(x,u)} f_w(z) dz \quad (7)$$

may be approximated by

$$ZANS(MM) = ZINT(INTT - 1) - ZINT(INTB) + AREAB + AREAT. \quad (8)$$

The areas AREAB and AREAT are found assuming $f_w(z)$ may be approximated by a straight line from $FWZ(I)$ to $FWZ(I + 1)$.

⁷F. B. Hildebrande, Introduction to Numerical Analysis, p. 60, McGraw-Hill, New York, 1956.

⁸W. E. Milne, Numerical Calculus, , Princeton University Press, Princeton, New Jersey, 1949.

Fig. 1. Schematic Drawing Showing Method for Performing the z -Integrations.

Experience with the calculation of weighting functions which have a nonfully illuminated entrance slit has shown that if the number of points in the nonzero range of $f_w(z)$ falls below about 100, errors of a few per cent will occur in the z -integrations.

After the integrations have been performed, the corresponding x integral is evaluated. Here, it is not practical to attempt to reduce the number of interpolations in the $e_w(x)$ table in the same manner as for $f_w(z)$. Hence, for each value of $ZANS(MM)$, a value of $EWX(MM)$ is interpolated, and their product $XINT(MM) = EWX(MM)*ZANS(MM)$ becomes the integrand. The values from the x integrations are the unnormalized values of $W_w(u)$. An integration of these values in loop 15 gives the normalization constant. The final values of $W_w(u)$ are obtained by dividing the unnormalized values by this constant. The computation of the various moments is a straightforward integration of the weighting function multiplied by the appropriate factor as indicated by Eq. (6).

DETAILED OPERATING INSTRUCTIONS

The program requires up to 45 input constants and, if $e_w(x)$ and $f_w(z)$ are not constant, two input tables. Each input variable is defined in Appendix D and its format is described in Table D.1. Although the appendix is self-explanatory, a few general remarks are in order.

Experience has shown that for greatest accuracy in the integration, the number of points for which the integration is performed (NU) should exceed 100. Comparison of computed results in which $NU = 101$ with an analytic evaluation of Eq. (1) for a very simple geometry shows that integration errors are less than 0.001%.

The program computation time is critically dependent on (i) the value of NU , and (ii) whether $e_w(x)$ and $f_w(z)$ are constant or are interpolated. It is readily seen that, including the computations necessary for normalization, the running time is roughly proportional to $(NU)^3$. Hence, increasing NU by a factor of 2, increases the running time eightfold. Typical running times on an IBM 360/91 computer for programs having

NU = 101 are approximately 4 sec if $e_w(x)$ and $f_w(z)$ are constant, 30 sec if $e_w(x)$ is constant but $f_w(z)$ is read in in tabulated form, and 50 sec if both $e_w(x)$ and $f_w(z)$ are read in. Running times are increased by about a factor of 5 on the IBM 360/75, as can be seen in the examples. The improvement in computing times of about a factor of 50 over our previous results on the IBM 360/50 is to be noted.

The printed output consists of the input data, including the values of z , $f_w(z)$, x , and $e_w(x)$ if they were given: the normalization factor; the u values and corresponding $W_w(u)$ values; and the moments of the function, if they were calculated. Punched output, if requested, contains only u and $W_w(u)$ values. Two examples are listed in Appendix E, along with the CALCOMP plotted results.

ACKNOWLEDGMENTS

The authors wish to express their gratitude to J. F. Jamison and R. W. Henderson of the Computing Technology Center, Oak Ridge Gaseous Diffusion Plant for their help in preparing this program, to Dr. L. B. Shaffer of Anderson College for valuable discussion, and to W. S. Smith of the University of Wisconsin for bringing to attention two significant errors in the original program.

APPENDIX A
PROGRAM WEIGHT FORTRAN IV LISTING

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

[illegible]

```

L-----      EWX(LB)=YLAG(SIMPX(LB),XVAL,XWX,ILP,3,NUMX,IEC)      870
L-----      85 CONTINUE      880
      90 ID=1      890
      TEMPWC(1)=-WC(1)      900
C****      910
C      LOOP ONE      920
C****      930
      WRITE(6,8025)      940
      8025 FORMAT(1H1,3X,8HSD ARRAY///)      950
      LIN=5      960
      IFLAG=0      970
      DO 105 I=1,M      980
      NDX=2*I      990
      MDX=NDX-1      1000
C      1010
C... CHECK TO SEE THAT THE SD ARRAY CONTAINS PAIRS OF EDGES EQUIDISTANT      1020
C... FROM THE SAMPLE PLANE TO THE I-TH EDGE, I=1,2,...,2*M      1030
C      1040
      IF(SC(NDX).EQ.SD(NDX-1))GO TO 95      1050
      IFLAG=1      1060
      WRITE(6,8030)MDX,SD(MCX),NDX,SC(NDX)      1070
      8030 FORMAT(1H ,3HSD(,13,3H)= ,F10.5,10H*****ERROR/      1080
      13HSD(,13,3H)= ,F10.5,10H*****ERROR)      1090
      GO TO 100      1100
      95 WRITE(6,8035)MCX,SD(MDX),NDX,SD(NDX)      1110
      8035 FORMAT(1H ,3HSD(,13,3H)= ,F10.5/1H ,3HSD(,13,3H)= ,F10.5)      1120
      100 LIN=LIN+2      1130
      IF(LIN.LE.40)GO TO 105      1140
      LIN=5      1150
      WRITE(6,8025)      1160
L-----      105 CONTINUE      1170
C****      1180
C      END LOOP ONE      1190
C****      1200
C****      1210
C      BEGIN LOOP TWO      1220
C****      1230
      WRITE(6,8040)      1240
      8040 FORMAT(1H1,3X,8HTD ARRAY///)      1250
      LIN=5      1260
      DO 120 I=1,N      1270
      NDX=2*I      1280
      MDX=NDX-1      1290
C      1300
C... CHECK TO SEE THAT THE TD ARRAY CONTAINS PAIRS OF EDGES EQUIDISTANT      1310
C... THE SAMPLE PLANE TO THE J-TH EDGE, J=1,2,...,2*N      1320
C      1330
      IF(TD(NDX).EQ.TD(MDX))GO TO 110      1340
      IFLAG=1      1350
      WRITE(6,8045)NDX,TD(MCX),NDX,TD(NDX)      1360
      8045 FORMAT(1H ,3HTD(,13,3H)= ,F10.5,10H*****ERROR/1H ,      1370
      13HTD(,13,3H)= ,F10.5,10H*****ERROR)      1380
      GO TO 115      1390
      110 WRITE(6,8050)NDX,TD(MDX),NDX,TD(NDX)      1400
      8050 FORMAT(1H ,3HTD(,13,3H)= ,F10.5/1H ,3HTD(,13,3H)= ,F10.5)      1410
      115 LIN=LIN+2      1420
      IF(LIN.LE.40)GO TO 120      1430
      LIN=5      1440
      WRITE(6,8040)      1450
L-----      120 CONTINUE      1460
C****      1470
C      END LOOP TWO      1480
C****      1490
C****      1500
C****      1510
C****      1520
C****      1530
C****      1540
C****      1550
C****      1560
C****      1570
C****      1580
C****      1590
C****      1600
C****      1610
C****      1620
C****      1630
C****      1640
C****      1650
C****      1660
C****      1670
C****      1680
C****      1690
C****      1700
C****      1710
C****      1720
C****      1730
C****      1740
C****      1750
C****      1760
C****      1770

```

```

I J      U(I,J)=WC(1)*(SD(I)+TD(J))/(SD(I)*(EL-TD(J))+POWER(LLI)*H(I)/SD(I) 1780
I J      1)+POWER(J)*(EL+SD(I))*G(J)/(SD(I)*(EL-TD(J))) 1781
I J      UIO(NUMB)=U(I,J) 1790
I J      NUMB=NUMB 1800
I J      GO TO 155 1810
I J      150 NUMA=NUMA+1 1820
I J      U(I,J)=-WC(1)*(SD(I)+TD(J))/(SD(I)*(EL-TD(J))+POWER(LLI)*H(I)/SD(I) 1830
I J      1)+POWER(J)*(EL+SD(I))*G(J)/(SD(I)*(EL-TD(J))) 1831
I J      UIE(NUMA)=U(I,J) 1840
I J      NUMA=NUMA 1850
I J      CONTINUE 1860
I J      C 1870
I J      C... CALCULATE UAAA AND UBBB 1880
I J      C 1890
I J      CALL SUBMIN(UIO,UDD,UBBB,NUMB,NUMIO) 1900
I J      CALL SUBMAX(UIE,UDE,UAAA,NUMA,NUMIE) 1910
I J      UINC=(UBBB-UAAA)/FLOAT(NU-1) 1920
I J      C 1930
I J      C....START U LOOP 1940
I J      C 1950
I J      DO 390 L=1,NU 1960
I J      IF(L-1) 160,160,165 1970
I J      160 UVAL(L)=UAAA 1980
I J      GO TO 170 1990
I J      165 UVAL(L)=UVAL(L-1) + UINC 2000
I J      170 IO=0 2010
I J      IE=0 2020
I J      C 2030
I J      C... CALCULATE X(I,IPRIME) 2040
I J      C 2050
I J      IEE=0 2060
I J      IOO=0 2070
I J      DO 175 I=1,M 2080
I J      I 2090
I J      I 2100
I J      I 2110
I J      I 2120
I J      I 2130
I J      I 2140
I J      I 2141
I J      I 2142
I J      I 2150
I J      I 2160
I J      I 2170
I J      I 2180
I J      I 2181
I J      I 2182
I J      I 2190
I J      I 2200
I J      I 2210
I J      I 2220
I J      I 2230
I J      I 2240
I J      I 2250
I J      I 2260
I J      I 2261
I J      I 2270
I J      I 2280
I J      I 2290
I J      I 2300
I J      I 2310
I J      I 2320
I J      I 2330
I J      I 2340
I J      I 2350
I J      I 2360
I J      I 2370
I J      I 2380
I J      I 2390
I J      I 2400
I J      I 2410
I J      I 2420
I J      I 2430
I J      I 2440
I J      I 2450
I J      I 2460
I J      I 2470
I J      I 2480
I J      I 2490
I J      I 2500
I J      I 2510
I J      I 2520
I J      I 2530
I J      I 2540
I J      I 2550
I J      I 2560
I J      I 2570
I J      I 2580
I J      I 2590
I J      I 2600
I J      I 2610
I J      I 2620
I J      I 2630
I J      I 2640
I J      I 2650
I J      I 2660
I J      I 2670
I J      I 2671
I J      I 2680

```

```

L M I      220 IO=IO+1                                2690
L M I      ZIO(IO)=ZIO(I)                            2700
L M I      IO=IO                                      2710
L M I      GO TO 230                                  2720
L M I      225 IE=IE+1                                2730
L M I      ZIE(IE)=ZIO(I)                            2740
L M I      IE=IE                                      2750
L M I----- 230 CONTINUE                             2760
L M I      C                                          2770
L M I      C... CALCULATE Z(J)                       2780
L M I      C                                          2790
L M J-----  DO 245 J=1,N2                           2800
L M J      LLJ=J+1                                    2810
L M J      ZJ(J)=(ELO*(EL-TD(J))*UVAL(L)-(ELO+TD(J))*XRAY(MM)+POWER(LLJ) * 2820
L M J      1(EL+ELO)*G(J))/(EL-TD(J))                2821
L M J      IF(MOD(J,2)) 235,240,235                  2830
L M J      235 JO=JO+1                                2840
L M J      ZJO(JO)=ZJ(J)                             2850
L M J      JO=JO                                       2860
L M J      GO TO 245                                  2870
L M J      240 JE=JE+1                                2880
L M J      ZJE(JE)=ZJ(J)                             2890
L M J      JE=JE                                       2900
L M J----- 245 CONTINUE                             2910
L M I      C                                          2920
L M I      CALCULATE ZB(X,U) AND ZA(X,U)              2930
L M I      C                                          2940
L M I      CALL SUBMIN(ZIO,ZJO,ZBU,IO,JO)            2950
L M I      CALL SUBMAX(7IF,7IF,7IF,7IF,IE,JE)        2960
L M I      ZBU(MM)=ZBU(I)                             2970
L M I      ZAU(MM)=ZAU(I)                             2980
L M I      C                                          2990
L M I      C.....DO Z INTEGRATION                   3000
L M I      C                                          3010
L M I      IF(IFZ) 250,325,250                        3020
L M I      250 IF(ZAU(MM)).GE. SIMPZ(1).AND.          3030
L M I      1 ZAU(MM).LE. SIMPZ(1TONU)) GO TO 255      3031
L M I      WRITE(6,8055)                               3040
L M I      8055 FORMAT(1H0,20X,'ZAU OUT OF RANGE OF ZVAL INPUT'/ 3050
L M I      1 20X,'TERMINATE PROCESSING'//)            3051
L M I      GO TO 5                                      3060
L M I      255 IF(ZBU(MM)).GE. SIMPZ(1).AND.          3070
L M I      1 ZBU(MM).LE. SIMPZ(1TONU)) GO TO 260      3071
L M I      WRITE(6,8060)                               3080
L M I      8060 FORMAT(1H0,20X,'ZBU OUT OF RANGE OF ZVAL INPUT'/ 3090
L M I      1 20X,'TERMINATE PROCESSING'//)            3091
L M I      GO TO 5                                      3100
L M I      260 AREAB=0.                                3110
L M I      AREAT=0.                                    3120
L M I      IF(ZAU(MM)-SIMPZ(1))265,265,270            3130
L M I      IF(ZBU(MM)-SIMPZ(1TONU))290,320,320        3140
L M I----- 270 DO 280 INTB=2,1TONU                  3150
L M I      IF(ZAU(MM)-SIMPZ(INTB))275,285,280         3160
L M I      275 FWZINB=FWZ(INTB-1)+(ZAU(MM)-SIMPZ(INTB-1))*(FWZ(INTB)-FWZ(INTB-1)) 3170
L M I      1/ZINCR                                     3171
L M I      AREAB=0.5*(SIMPZ(INTB)-ZAU(MM))*(FWZ(INTB)+FWZINB) 3180
L M I      GO TO 285                                    3190
L M I----- 280 CONTINUE                             3200
L M I      285 INTB=INTB+1                             3210
L M I      GO TO 295                                  3220
L M I      290 INTB=1                                   3230
L M I----- 295 DO 305 INTT=1,1TONU                   3240
L M I      IF(ZBU(MM)-SIMPZ(INTT))300,310,305         3250
L M I      300 FWZINT=FWZ(INTT-1)+(ZBU(MM)-SIMPZ(INTT-1))*(FWZ(INTT)-FWZ(INTT-1)) 3260
L M I      1/ZINCR                                     3261
L M I      AREAT=0.5*(ZBU(MM)-SIMPZ(INTT))*(FWZ(INTT)+FWZINT) 3270
L M I      GO TO 315                                    3280
L M I----- 305 CONTINUE                             3290
L M I      310 AREAT=ZINT(INTT)-ZINT(INTT-1)          3300
L M I      315 ZANS(MM)=ZINT(INTT-1)-ZINT(INTB)*AREAB+AREAT 3310
L M I      GO TO 330                                    3320
L M I      320 ZANS(MM)=ZINT(1TONU)                   3330
L M I      GO TO 330                                    3340
L M I      325 ZANS(MM)=ZBU(MM)-ZAU(MM)               3350
L M I----- 330 CONTINUE                             3360
L M I      C                                          3370
L M I      C.....END X LOOP                          3380
L M I      C                                          3390
L M I      C                                          3400
L M I      C.....DU X INTEGRATION                   3410
L M I      C                                          3420
L M I      IF(IEK)335,365,335                         3430
L M I----- 335 DO 360 MM=1,NU                        3440
L M I      IF(XRAY(MM)).GE. SIMPX(1).AND.             3450
L M I      1 XRAY(MM).LE. SIMPX(1TONU)) GO TO 340     3451
L M I      WRITE(6,8065)                               3460
L M I      8065 FORMAT(1H0,20X,'XRAY OUT OF RANGE OF XVAL INPUT'/ 3470
L M I      1 20X,'TERMINATE PROCESSING'//)            3471
L M I      340 IF(XRAY(MM)-SIMPX(1))345,350,350       3480
L M I      345 XINT(MM)=0.                             3490
L M I      GO TO 360                                    3500
L M I      350 IF(XRAY(MM)-SIMPX(1TONU))355,355,345   3510
L M I      355 XINT(MM)=ZANS(MM)*YLAG(XRAY(MM),SIMPX,EWX,0,2,1TONU,IEO) 3520
L M I----- 360 CONTINUE                             3530
L M I      GO TO 375                                    3540
L M I----- 365 DO 370 I=1,NU                        3550
L M I----- 370 XINT(I)=ZANS(I)                      3560

```

```

L      C
L      C.....WRITE INTERMEDIATE COMPUTATIONS
L      C
L      375 IF(KOUT)280,385,380
L      380 WRITE(6,8070)UVAL(L),XAU(L),XBU(L),(ZAU(ML),ZBU(ML),ZANS(ML),XINT(
L      1ML),ML=1,NU)
L      6070 FORMAT(1H1,3HIU=,F9.5,5X,4HXAU=,E14.6,5X,4HXB=,E14.6/3X,2(3HZAU,
L      111X,3HZBU,11X,4HZANS,10X,4HXINT,10X)/(1X,8E14.6))
L      C
L      C... CALCULATE UNNORMALIZED WW(U)
L      C
L      385 CALL SIMP(XRAY,XINT,XANS,NU)
L      WWUP(L)=XANS(NU)*EL*ELO/(EL+ELC)
L      C
L      390 CONTINUE
L      C
L      C.....END U LOOP
L      C
L      C... CALCULATE AW (NORMALIZATION FACTOR)
L      C
L      CALL SIMP(UVAL,WWUP,AANS,NU)
L      IF(AANS(NU).NE.0.) GO TO 395
L      WRITE(6,8075)
L      8075 FCMT(1H0,20X,'AANS(NU) = 0.'//)
L      GO TO 5
L      C
L      C...CALCULATE NORMALIZED WW(U) AND CONVERT TO MILLIRADIANS
L      C
L      395 DO 400 L=1,NU
L      WWU(L)=WWUP(L)*0.001/AANS(NU)
L      UVAL(L)=1000.*UVAL(L)
L      400 CONTINUE
L      C
L      C... CALCULATE THE MOMENTS ABOUT THE ORIGIN
L      C
L      C... DO WE WANT TO CALCULATE MOMENTS ABOUT THE ORIGIN.
L      C
L      IF(KK) 435,435,405
L      DO 430 MAP=1,KK
M      405
M      DO 410 LAP=1,NU
M      410 USTORE(LAP)=(UVAL(LAP)**MAP)*WWU(LAP)
M      CALL SIMP(UVAL,USTORE,AMOMNT,NU)
M      IF(MAP=2) 415,420,425
M      415 FIRSTM=AMOMNT(NU)
M      GO TO 430
M      420 SECDMM=AMOMNT(NU)
M      GO TO 430
M      425 THIRDM=AMOMNT(NU)
M      430 CONTINUE
M      C
M      C... DO WE WANT TO CALCULATE MOMENTS ABOUT THE MEAN.
M      C
M      435 IF(JJ) 470,470,440
M      C
M      C... CALCULATE THE MOMENTS ABOUT THE MEAN
M      C
M      440 DO 465 MAP=1,JJ
M      445 DO 445 LAP=1,NU
M      445 UDIF(LAP)=(UVAL(LAP)-FIRSTM)**MAP*WWU(LAP)
M      CALL SIMP(UVAL,UDIF,AMOMNT,NU)
M      IF(MAP=2) 450,455,460
M      450 FRSTM=AMOMNT(NU)
M      GO TO 465
M      455 SCNDMM=AMOMNT(NU)
M      GO TO 465
M      460 THRDMM=AMOMNT(NU)
M      465 CONTINUE
M      470 CONTINUE
I      DO 475 I=1,NU
I      475 UVAL(I)=UVAL(I)-FIRSTM
I      WRITE(6,8080)
I      8080 FORMAT(1H1,20X,'INPUT DATA'//)
I      WRITE(6,8085)M,N,NU,KK,JJ,IOUT,IFZ,IEX,NUMZ,NUMX,KOUT,EL,ELO,WC
I      8085 FORMAT(1H0,4X,2HM=,I1,3X,2HN=,I1,3X,3HNU=,I3,3X,3HKK=,I1,3X,3HJJ=,
I      111,3X,5HIOUT=,I1,3X,4HIFZ=,I1//
I      21X,4HIEX=,I1,3X,5HNUMZ=,I5,3X,5HNUMX=,I5,3X,5HKOUT=,I1//
I      3X,3HEL=,F8.4,3X,4HEL0=,F8.4,3X,3HWC=,F8.4//)
I      WRITE(6,8090)
I      8090 FORMAT(1H0,2X,12H SUBSCRIPT I,10X,12H SUBSCRIPT J)
I      WRITE(6,8095)(SD(I),TD(I),I=1,8),(H(I),G(I),I=1,8)
I      8095 FORMAT(1H0, 2(6H D(I)=,F7.3,10X)/2(6H D(2)=,F7.3,10X)/2(6H D(3)=
I      1,F7.3,10X)/2(6H D(4)=,F7.3,10X)/ 2(6H D(5)=,F7.3,10X)/2(6H D(6)=,F
I      27.3,10X)/2(6H D(7)=,F7.3,10X)/2(6H D(8)=,F7.3,10X)/2(6H H(1)=,F7.3
I      3,10X)/2(6H H(2)=,F7.3,10X)/2(6H H(3)=,F7.3,10X)/2(6H H(4)=,F7.3,10
I      4X)/2(6H H(5)=,F7.3,10X)/2(6H H(6)=,F7.3,10X)/2(6H H(7)=,F7.3,10X)/
I      52(6H H(8)=,F7.3,10X))
I      IF(IEX)480,485,480
I      480 CALL WOUT1(NUMX,XVAL,XWX)
I      485 IF(IFZ)490,495,490
I      490 CALL WOUT2(NUMZ,ZVAL,ZWZ)
I      495 IF(IOUT) 500,505,500
I      500 WRITE(7,8100) NU
I      8100 FORMAT(15)
I      WRITE(7,8105) (UVAL(I),WWU(I),I=1,NU)
I      8105 FORMAT(2E20.8,40X)
I      505 WRITE(6,8110) (TITLE(I),I=1,80)
I      8110 FORMAT(1H1,20X,80A1,///)
I      WRITE(6,8115)

```



```

8115 FORMAT(1H0,4X,2H U,7X,6H WW(U),37X,2H U,7X,6H WW(U)//) 4460
      NNU=NU/2 4470
      WRITE(6,8120)(UVAL(I),WWU(I),UVAL(I+NNU),WWU(I+NNU),I=1,NNU) 4480
8120 FORMAT(1H ,F11.6,E13.5,28X,F11.6,E13.5) 4490
      IF(MOD(NU,2))510,515,510 4500
510  WRITE(6,8125) UVAL(NU),WWU(NU) 4510
8125 FORMAT(1H ,52X,F11.6,E13.5) 4520
      WRITE(6,8130)AANS(NU) 4530
8130 FORMAT(1H0,24HNORMALIZATION FACTOR AW=,E13.5) 4540
515  IF(KK) 520,540,520 4550
520  WRITE(6,8135)FIRSTM,SECONDM,THIRDM 4560
8135 FORMAT(1H0,40HMOMENTS OF THE FUNCTION ABOUT THE ORIGIN//1X,5H1ST 4570
      1=,E15.5,10X,5H2ND =,E15.5,10X,5H3RD =,E15.5) 4571
I----- DO 525 I=1,NU 4580
I      II=NU+1-I 4590
I      IF(WWU(II) .NE. 0.) GO TO 530 4600
I----- 525 CONTINUE 4610
530  RESOLV=1.5417/(UVAL(II)*0.001) 4620
      WRITE(6,8140) RESOLV, UVAL(II) 4630
8140 FORMAT(1H0,'THEORETICAL LIMIT OF RESOLUTION = ',F10.1,' ANG.'// 4640
      1 OR = ',F10.2,' MR.'//) 4641
      IF(JJ) 535,540,535 4650
535  WRITE(6,8145)FIRSTMM,SCNDMM,THRCMM 4660
8145 FORMAT(1H0,38HMOMENTS OF THE FUNCTION ABOUT THE MEAN//1X,5H1ST =, 4670
      1E15.5,10X,5H2ND =,E15.5,10X,5H3RD =,E15.5) 4671
540  CALL TIMER(1) 4680
      IF(IPLT .EQ. 0) GO TO 5 4690
      IF(IPLT .LT. 0) GO TO 545 4700
      CALL PLOTG(APLTZ,2000) 4710
      IPLT=-IPLT 4720
545  CALL LNPLOT(-NU,1,UVAL,1,.07,WWU) 4730
      CALL SYMBOL(1.0,10.6,0.14,TITLE(1),0.0,80) 4740
      CALL SYMBOL(5.0,0.1,0.14,16HU (MILLIRADIANS),0.0,16) 4750
      CALL SYMBOL(0.3,4.0,0.14,5HWW(L),90.0,5) 4760
      CALL PLOT(13.0,0.0,-3) 4770
      GC TO 5 4780
550  STOP 4790
      END 4800

```

TABLE OF VARIABLE NAMES

VARIABLE NAME	ARRAY DIMENSIONS	TOTAL NO ELEMENTS	ELEMENT TYPE	DECLARATION	CROSS REFERENCE TABLE (BY SEQUENCE NUMBERS)
AANS	(202)	202	REAL*4	LOCAL	140, 3750, 3760, 3840, 4530
AMCMNT	(202)	202	REAL*4	LOCAL	140, 3970, 3980, 4010, 4030, 4150, 4170, 4190, 4210
APLTZ	(2000)	2000	REAL*4	LOCAL	150, 4710
AREAB			REAL*4	LOCAL	3110* 3180* 3310
AREAT			REAL*4	LOCAL	3120* 3270* 3300* 3310
BLANK			REAL*4	LOCAL (DATA)	170, 360, 370, 380, 390
BLNK			REAL*4	LOCAL (DATA)	180, 270
EL			REAL*4	LOCAL	430, 1630, 1670, 1780, 1820, 2140, 2180, 2260, 2670, 2820, 3670, 4280
ELD			REAL*4	LOCAL	430, 2670, 2820, 3670, 4280
EWX	(1001)	1001	REAL*4	LOCAL	140, 830* 870* 3520
FIRSTM			REAL*4	LOCAL	150* 3590* 4140, 4250, 4560
FLOAT			REAL*4	FUNCTION	1920, 2490
FRSTMM			REAL*4	LOCAL	220* 4170* 4660
FWZ	(1001)	1001	REAL*4	LOCAL	140, 630* 670* 690, 3170, 3180, 3260, 3270
FWZINB			REAL*4	LOCAL	3170* 3180
FWZINT			REAL*4	LOCAL	3260* 3270
G	(8)	8	REAL*4	LOCAL	140, 390* 490, 1780, 1830, 2260, 2820, 4320
H	(8)	8	REAL*4	LOCAL	140, 380* 460, 1630, 1670, 1780, 1820, 2140, 2180, 2260, 2670, 4320
I			INTEGER*4	LOCAL	260* 270, 280* 450* 460* 540* 740* 980* 990, 1260* 1270, 1570* 1590, 1610, 1630, 1650, 1670, 1710* 1730, 1740, 1780, 1790, 1830, 1840, 2080* 2160, 2120, 2140, 2160, 2180, 2230* 2260, 2270, 2310, 2350, 2650* 2660, 2670, 2680, 2760, 2740, 3550* 3560, 4240* 4250, 4320* 4410* 4430* 4460* 4580* 4590
ID			INTEGER*4	LOCAL	890* 2450, 2460
IE			INTEGER*4	LOCAL	2020* 2300* 2310, 2320* 2410, 2590* 2730* 2740, 2750* 2960
ICC			INTEGER*4	LOCAL	870
IEE			INTEGER*4	LOCAL	2060* 2150* 2180, 2410
IEU			INTEGER*4	LOCAL	670, 3520
IEX			INTEGER*4	LOCAL	300, 730, 3430, 4280, 4340
IFLAG			INTEGER*4	LOCAL	970* 1060* 1340* 1510
IFZ			INTEGER*4	LOCAL	300, 530, 3020, 4280, 4360
II			INTEGER*4	LOCAL	4590* 4600, 4620, 4630
IINTB			INTEGER*4	LOCAL	3210* 3230* 3240
ILP			INTEGER*4	LOCAL	860* 870
IND			INTEGER*4	LOCAL	660* 670
IINTB			INTEGER*4	LOCAL	3150* 3160, 3170, 3180, 3210, 3310
INTT			INTEGER*4	LOCAL	3240* 3250, 3260, 3270, 3300, 3310
IO			INTEGER*4	LOCAL	2010* 2340* 2350, 2360* 2420, 2580* 2690* 2700, 2710* 2950
IOO			INTEGER*4	LOCAL	2070* 2110* 2140, 2420
IOUT			INTEGER*4	LOCAL	300, 4280, 4380
IP			INTEGER*4	LOCAL	1580* 1590, 1620, 1630, 1660, 1670, 2090* 2100, 2130, 2140, 2170, 2180
IPLOT			INTEGER*4	LOCAL	300, 4690, 4700, 4720*

ITONU	INTEGER*4	LOCAL	340* 3070, 3520	500, 3140,	600, 3150,	690, 3240,	780, 3330,	800, 3450,	3030, 3510,
ITWONU	INTEGER*4	LOCAL	320*	330,	340				
J	INTEGER*4	LOCAL	480* 1830, 2310, 2890	490* 1840, 2350,	1720* 2240* 2800*	1750, 2250, 2810,	1760, 2260, 2820,	1780, 2280, 2830,	1790, 2290, 2850,
JE	INTEGER*4	LOCAL	2610*	2880*	2890,	2900*	2960		
JJ	INTEGER*4	LOCAL	300,	4080,	4120,	4280,	4650		
JO	INTEGER*4	LOCAL	2600*	2840*	2850,	2860*	2950		
K	INTEGER*4	LOCAL	350*	360,	370,	380,	390		
KK	INTEGER*4	LOCAL	300,	3930,	3940,	4280,	4550		
KOUT	INTEGER*4	LOCAL	300,	3600,	4280				
L	INTEGER*4	LOCAL	1960* 2480, 3830*	1970, 2490, 3840,	2000, 2550, 3850	2140, 2670,	2180, 2820,	2260, 3010,	2470, 3670,
LA	INTEGER*4	LOCAL	580*	590,	600*	620,	630,	670	
LAP	INTEGER*4	LOCAL	3950*	3960,	4130*	4140			
LB	INTEGER*4	LOCAL	780*	790,	800*	820,	830,	870	
LIN	INTEGER*4	LOCAL	960* 1420*	1120*	1130,	1140*	1250*	1400*	1410,
LLI	INTEGER*4	LOCAL	1610* 2120*	1630, 2140,	1650* 2160*	1670, 2180,	1730* 2660*	1780, 2670	1830,
LLJ	INTEGER*4	LOCAL	2250*	2260,	2810*	2820			
LLP	INTEGER*4	LOCAL	1620* 2180	1630,	1660*	1670,	2130*	2140,	2170*
LNPL0T		SUBROUTINE	4730						
LO	INTEGER*4	LOCAL	610*	620,	630,	660			
LP	INTEGER*4	LOCAL	810*	820,	830,	860			
M	INTEGER*4	LOCAL	300, 4280	410,	980,	1570,	1580,	2080,	2090,
M&P	INTEGER*4	LOCAL	3940*	3960,	3980,	4120*	4140,	4160	
MDX	INTEGER*4	LOCAL	1000*	1070,	1100,	1200*	1320,	1350,	1380
ML	INTEGER*4	LOCAL	3610*						
MM	INTEGER*4	LOCAL	2530* 2980, 3180, 3440*	2540, 3030, 3250, 3450,	2550, 3070, 3260, 3480,	2570, 3130, 3270, 3490,	2670, 3140, 3310, 3510,	2820, 3160, 3330, 3520	2970, 3170, 3350,
MOD	INTEGER*4	FUNCTION	1740, 2830,	1750, 4500	1760,	2270,	2280,	2290,	2680,
MZ	INTEGER*4	LOCAL	410*	450,	460,	1710,	2230,	2650	
N	INTEGER*4	LOCAL	300,	420,	1260,	4280			
NDX	INTEGER*4	LOCAL	990* 1330,	1000, 1350,	1050, 1380	1070,	1100,	1270*	1280,
NNU	INTEGER*4	LOCAL	4470*	4480					
NU	INTEGER*4	LOCAL	300, 3550, 3840, 4150, 4410, 4730	320, 3610, 3950, 4170, 4470,	1920, 3660, 3970, 4190, 4500,	1960, 3670, 3990, 4210, 4510,	2490, 3750, 4010, 4240, 4530,	2530, 3760, 4030, 4280, 4580,	3440, 3830, 4130, 4390, 4590,
NUMA	INTEGER*4	LOCAL	1690*	1820*	1840,	1850*	1910		
NUMB	INTEGER*4	LOCAL	1700*	1770*	1790,	1800*	1900		
NUMIE	INTEGER*4	LOCAL	1560*	1600*	1630,	1910			
NUMIO	INTEGER*4	LOCAL	1550*	1640*	1670,	1900			
NUMX	INTEGER*4	LOCAL	300, 4350	740,	750,	760,	810,	870,	4280,
NUMZ	INTEGER*4	LOCAL	300,	540,	560,	610,	670,	4280,	4370

N2			INTEGER*4	LOCAL	420*	480,	490,	1720,	2240,	2800	
PLOT				SUBROUTINE	4770						
PLOTS				SUBROUTINE	4710						
POWER			REAL*4	FUNCTION	1630, 2670,	1670, 2820	1760,	1830,	2140,	2180,	2260,
RESOLV			REAL*4	LOCAL	4620*	4630					
SCNDMM			REAL*4	LOCAL	230*	4190*	4660				
SD	(8)	8	REAL*4	LOCAL	140, 1670, 4320	360* 1780,	450, 1830,	1050, 2140,	1070, 2180,	1100, 2260,	1630, 2670,
SECNDM			REAL*4	LOCAL	200*	4010*	4560				
SIMP				SUBROUTINE	690,	3660,	3750,	3970,	4150		
SIMPX	(1001)	1001	REAL*4	LOCAL	140, 3510,	770* 3520	790*	820,	870,	3450,	3480,
SIMPZ	(1001)	1001	REAL*4	LOCAL	140, 3070, 3260,	570* 3130, 3270	590* 3140,	620, 3160,	670, 3170,	690, 3180,	3030, 3250,
SUBMAX				SUBROUTINE	1910,	2420,	2460,	2960			
SUBMIN				SUBROUTINE	1900,	2410,	2450,	2950			
SYMBOL				SUBROUTINE	4740,	4750,	4760				
TD	(8)	8	REAL*4	LOCAL	140, 1820,	370* 2260,	480, 2820,	1330, 4320	1350,	1380,	1780,
TEMPWC	(1)	1	REAL*4	LOCAL	140,	900*	2460				
THIRDM			REAL*4	LOCAL	210*	4030*	4560				
THRDMM			REAL*4	LOCAL	240*	4210*	4660				
TIMER				SUBROUTINE	250,	4680					
TITLE	(80)	80	LOGICAL*1	LOCAL	160,	270*	280,	4430,	4740		
TWONU			REAL*4	LOCAL	330*	560,	750,	760			
U	(8,8)	64	REAL*4	LOCAL	140,	1780*	1790,	1830*	1840		
UAAA			REAL*4	LOCAL	1910,	1920,	1980				
UBBB			REAL*4	LOCAL	1900,	1920					
UDIF	(202)	202	REAL*4	LOCAL	140,	4140*	4150				
UEE	(8)	8	REAL*4	LOCAL	140,	1630*	1910				
UIE	(6)	8	REAL*4	LOCAL	140,	1840*	1910				
UINC			REAL*4	LOCAL	1920*	2000					
UIC	(8)	8	REAL*4	LOCAL	140,	1790*	1900				
UUC	(8)	8	REAL*4	LOCAL	140,	1670*	1900				
USTORE	(202)	202	REAL*4	LOCAL	140,	3960*	3970				
UVAL	(202)	202	REAL*4	LOCAL	140, 2820, 4150, 4730	1980* 3610, 4250*	2000* 3750, 3850*	2140, 3960, 4480,	2180, 3970, 4510,	2260, 4140, 4620,	2670, 4140, 4630,
WC	(1)	1	REAL*4	LOCAL	140, 2450,	430, 4280	900,	1630,	1670,	1780,	1830,
WOUT1				SUBROUTINE	4350						
WOUT2				SUBROUTINE	4370						
WWU	(202)	202	REAL*4	LOCAL	140, 4600,	3840* 4730	3960,	4140,	4410,	4480,	4510,
WWUP	(202)	202	REAL*4	LOCAL	140,	3670*	3750,	3840			
X	(8,8)	64	REAL*4	LOCAL	140,	2260*	2310,	2350			
XANS	(202)	202	REAL*4	LOCAL	140,	3660,	3670				
XAU	(202)	202	REAL*4	LOCAL	140,	2480*	2490,	2550,	3610		
XAU1			REAL*4	LOCAL	2460,	2480					
XBU	(202)	202	REAL*4	LOCAL	140,	2470*	2490,	3610			

NAME	UNIT	REAL	LOCAL	2450	2470					
XBU1	(8)	8	REAL*4	LOCAL	140,	2310*	2410			
XINC			REAL*4	LOCAL	2490*	2570				
XINCR			REAL*4	LOCAL	750*	760*	790			
XINT	(1001)	1001	REAL*4	LOCAL	140,	3490*	3520*	3560*	3610,	3660
XIO	(8)	8	REAL*4	LOCAL	140,	2350*	2420			
XMAX			REAL*4	LOCAL	2420,	2440				
XMAXC	(1)	1	REAL*4	LOCAL	140,	2440*	2460			
XMIN			REAL*4	LOCAL	2410,	2430				
XMINO	(1)	1	REAL*4	LOCAL	140,	2430*	2450			
XRAY	(1001)	1001	REAL*4	LOCAL	140,	2550*	2570*	2670,	2820,	3450,
					3510,	3520,	3660			
XVAL	(50)	50	REAL*4	LOCAL	140,	740,	750,	760,	770,	820,
					4350					870,
XWX	(50)	50	REAL*4	LOCAL	140,	740,	830,	870,	4350	
XXE	(8)	8	REAL*4	LOCAL	140,	2100*	2410			
XXU	(8)	8	REAL*4	LOCAL	140,	2140*	2420			
YLAG			REAL*4	FUNCTION	670,	870,	3520			
ZANS	(202)	202	REAL*4	LOCAL	140,	3310*	3330*	3350*	3370,	3380,
ZAU	(202)	202	REAL*4	LOCAL	140,	2980*	3030,	3130,	3160,	3170,
					3350,	3610				
ZAU1			REAL*4	LOCAL	2960,	2980				
ZBU	(202)	202	REAL*4	LOCAL	140,	2970*	3070,	3140,	3250,	3260,
					3350,	3610				
ZBU1			REAL*4	LOCAL	2950,	2970				
ZI	(8)	8	REAL*4	LOCAL	140,	2670*	2700,	2740		
ZIE	(8)	8	REAL*4	LOCAL	140,	2740*	2960			
ZINCR			REAL*4	LOCAL	840*	800,	3170,	3260		
ZINT	(1001)	1001	REAL*4	LOCAL	140,	690,	3300,	3310,	3330	
ZIN	(8)	8	REAL*4	LOCAL	140,	2700*	2950			
ZJ	(8)	8	REAL*4	LOCAL	140,	2820*	2850,	2890		
ZJE	(8)	8	REAL*4	LOCAL	140,	2890*	2960			
ZJO	(8)	8	REAL*4	LOCAL	140,	2850*	2950			
ZVAL	(50)	50	REAL*4	LOCAL	140,	540,	560,	570,	620,	670,
										4370
ZWZ	(50)	50	REAL*4	LOCAL	140,	540,	630,	670,	4370	

* = VARIABLES USED ON LEFT HAND SIDE OF EQUAL SIGN

INPUT / OUTPUT TABLE

DEFINING SEQ. NO.	I/O OPERATION	LOGICAL UNIT	I/O TYPE	FORMAT USED	I/O LIST
280	READ	5	BCD	8000	(TITLE(I), I=1, 80)
300	READ	5	BCD	8005	M, N, NU, KK, JJ, IOUT, IFZ, IEX, NUMZ, NUMX, KOUT, IPLOT
430	READ	5	BCD	8010	EL, ELC, WC
450	READ	5	BCD	8015	(SD(I), I=1, M2)
460	READ	5	BCD	8015	(H(I), I=1, M2)
480	READ	5	BCD	8015	(TD(J), J=1, N2)
490	READ	5	BCD	8015	(G(J), J=1, N2)
540	READ	5	BCD	8020	(ZVAL(I), ZHZ(I), I=1, NUMZ)
740	READ	5	BCD	8020	(XVAL(I), XWX(I), I=1, NUMX)
940	WRITE	6	BCD	8025	
1070	WRITE	6	BCD	8030	MDX, SD(MDX), NDX, SD(NDX)
1100	WRITE	6	BCD	8035	MDX, SD(MDX), NDX, SD(NDX)
1150	WRITE	6	BCD	8025	
1230	WRITE	6	BCD	8040	
1350	WRITE	6	BCD	8045	MDX, TD(MDX), NDX, TD(NDX)
1380	WRITE	6	BCD	8050	MDX, TD(MDX), NDX, TD(NDX)
1430	WRITE	6	BCD	8040	
3040	WRITE	6	BCD	8055	
3080	WRITE	6	BCD	8060	
3460	WRITE	6	BCD	8065	
3610	WRITE	6	BCD	8070	UVAL(L), XAU(L), XBU(L), (ZAU(ML), ZBU(ML), ZANS(ML), XINT(ML), ...)
3770	WRITE	6	BCD	8075	
4260	WRITE	6	BCD	8080	
4280	WRITE	6	BCD	8085	M, N, NU, KK, JJ, IOUT, IFZ, IEX, NUMZ, NUMX, KOUT, EL, ELC, WC
4300	WRITE	6	BCD	8090	
4320	WRITE	6	BCD	8095	(SD(I), TD(I), I=1, 8), (H(I), G(I), I=1, 8)
4430	WRITE	6	BCD	8110	(TITLE(I), I=1, 80)
4450	WRITE	6	BCD	8115	
4480	WRITE	6	BCD	8120	(UVAL(I), WWU(I), UVAL(I+NNU), WWU(I+NNU), I=1, NNU)
4510	WRITE	6	BCD	9125	UVAL(NU), WWU(NU)
4530	WRITE	6	BCD	8130	AANS(NU)
4560	WRITE	6	BCD	9135	FIRSTM, SECONDM, THIRDM
4630	WRITE	6	BCD	8140	RESULV, UVAL(II)
4660	WRITE	6	BCD	8145	FRSTMM, SCNDMM, THRDMM
4390	WRITE	7	BCD	8100	NU
4410	WRITE	7	BCD	8105	(UVAL(I), WWU(I), I=1, NNU)

```

FUNCTION POWER(LL)
POWER = FLOAT(ISIGN(1,(1-2*MOD(LL,2))))
RETURN
END

```

```

POWER 10
POWER 20
POWER 30
POWER 40

```

TABLE OF VARIABLE NAMES

VARIABLE NAME	ARRAY DIMENSIONS	TOTAL NO ELEMENTS	ELEMENT TYPE	DECLARATION	CONSECUTIVE (BY SEQUENCE NUMBERS)
FLOAT			REAL*4	FUNCTION	20
ISIGN			INTEGER*4	FUNCTION	20
LL			INTEGER*4	PARAMETER	10, 20
MOD			INTEGER*4	FUNCTION	20
POWER			REAL*4	ENTRY POINT	10, 20*

* = VARIABLES USED ON LEFT HAND SIDE OF EQUAL SIGN

```

SUBROUTINE SUBMIN(A,B,C,M,N)
DIMENSION A(1),B(1)
AMIN=A(1)
BMIN=B(1)
DO 10 I=1,M
IF(AMIN-A(I)) 10,5,5
I----- 5
AMIN=A(I)
I----- 10
CONTINUE

J-----
DO 20 J=1,N
IF(BMIN-B(J)) 20,15,15
J----- 15
BMIN=B(J)
J----- 20
CONTINUE
IF(AMIN-BMIN)25,25,30
25 C=AMIN
GO TO 35
30 C=BMIN
25 RETURN
END

```

```

SUBMI 10
SUBMI 20
SUBMI 30
SUBMI 40
SUBMI 50
SUBMI 60
SUBMI 70
SUBMI 80

SUBMI 90
SUBM 100
SUBM 110
SUBM 120
SUBM 130
SUBM 140
SUBM 150
SUBM 160
SUBM 170
SUBM 180

```

TABLE OF VARIABLE NAMES

VARIABLE NAME	ARRAY DIMENSIONS	TOTAL NO ELEMENTS	ELEMENT TYPE	DECLARATION	CROSS REFERENCE TABLE (BY SEQUENCE NUMBERS)				
A	(1)	1	REAL*4	PARAMETER	10,	20,	30,	60,	70
AMIN			REAL*4	LOCAL	30*	60,	70*	130,	140
B	(1)	1	REAL*4	PARAMETER	10,	20,	40,	100,	110
BMIN			REAL*4	LOCAL	40*	100,	110*	130,	160
C			REAL*4	PARAMETER	10,	140*	160*		
I			INTEGER*4	LOCAL	50*	60,	70		
J			INTEGER*4	LOCAL	90*	100,	110		
M			INTEGER*4	PARAMETER	10,	50			
N			INTEGER*4	PARAMETER	10,	90			
SUBMIN			REAL*4	ENTRY POINT	10				

* * VARIABLES USED ON LEFT HAND SIDE OF EQUAL SIGN


```

SUBROUTINE SIMP(XX,FX,AX,NX)
C PROGRAM AUTHOR ANON., CENTRAL DATA PROCESSING, 1/1/65.
DIMENSION XX(2),FX(2),AX(2)
DELX=XX(2)-XX(1)
AX(1)=0.0
I----- DO 10 IX=2,NX,2
I          IF(NX-IX)5,15,5
I          5 AX(IX)=DELX*(5.0*FX(IX-1)+8.0*FX(IX)-FX(IX+1))/12.0+AX(IX-1)
I----- 10 AX(IX+1)=DELX*(FX(IX-1)+4.0*FX(IX)+FX(IX+1))/3.0+AX(IX-1)
          GO TO20
          15 AX(NX)=DELX*(-FX(NX-2)+8.0*FX(NX-1)+5.0*FX(NX))/12.0+AX(NX-1)
          20 RETURN
          END
SIMP 10
SIMP 20
SIMP 30
SIMP 40
SIMP 50
SIMP 60
SIMP 70
SIMP 80
SIMP 90
SIMP 100
SIMP 110
SIMP 120
SIMP 130

```

TABLE OF VARIABLE NAMES

VARIABLE NAME	ARRAY DIMENSIONS	TOTAL NO ELEMENTS	ELEMENT TYPE	DECLARATION	CROSS REFERENCE TABLE (BY SEQUENCE NUMBERS)					
AX	(2)	2	REAL*4	PARAMETER	10,	30,	50*	80*	90*	110*
DELX			REAL*4	LOCAL	40*	80,	90,	110		
PA	(2)	2	REAL*4	PARAMETER	10,	20,	80,	90,	110	
IX			INTEGER*4	LOCAL	60*	70,	80,	90		
NX			INTEGER*4	PARAMETER	10,	60,	70,	110		
SIMP			REAL*4	ENTRY POINT	10					
XX	(2)	2	REAL*4	PARAMETER	10,	30,	40			

* = VARIABLES USED ON LEFT HAND SIDE OF EQUAL SIGN

```

SUBROUTINE SUBMAX(A,B,C,M,N)
DIMENSION A(1),B(1)
AMAX=A(1)
BMAX=B(1)
DO 10 I=1,M
  IF(AMAX-A(I)) 5,10,10
  I----- 5
  I----- 10
  AMAX=A(I)
  CONTINUE
DO 20 J=1,N
  IF(BMAX-B(J)) 15,20,20
  J----- 15
  J----- 20
  BMAX=B(J)
  CONTINUE
  IF(AMAX-BMAX)25,30,30
  25 C=BMAX
  GO TO 35
  30 C=AMAX
  35 RETURN
END

```

SUBMA 10
 SUBMA 20
 SUBMA 30
 SUBMA 40
 SUBMA 50
 SUBMA 60
 SUBMA 70
 SUBMA 80
 SUBMA 90
 SUBM 100
 SUBM 110
 SUBM 120
 SUBM 130
 SUBM 140
 SUBM 150
 SUBM 160
 SUBM 170
 SUBM 180

TABLE OF VARIABLE NAMES

VARIABLE NAME	ARRAY DIMENSIONS	TOTAL NO ELEMENTS	ELEMENT TYPE	DECLARATION	CROSS REFERENCE TABLE (BY SEQUENCE NUMBERS)				
A	(1)	1	REAL*4	PARAMETER	10,	20,	30,	60,	70
AMAX			REAL*4	LOCAL	30*	60,	70*	130,	160
B	(1)	1	REAL*4	PARAMETER	10,	20,	40,	100,	110
BMAX			REAL*4	LOCAL	40*	100,	110*	130,	140
C			REAL*4	PARAMETER	10,	140*	160*		
I			INTEGER*4	LOCAL	50*	60,	70		
J			INTEGER*4	LOCAL	90*	100,	110		
M			INTEGER*4	PARAMETER	10,	50			
N			INTEGER*4	PARAMETER	10,	90			
SUBMAX			REAL*4	ENTRY POINT	10				

* = VARIABLES USED ON LEFT HAND SIDE OF EQUAL SIGN

```

      FUNCTION YLAG(XI,X,Y,IND1,N1,IMAX,IEX), YLAG 10
C  PROG.AUTHORS A.A.BROOKS AND E.C.LONG,CENTRAL DATA PROCESSING,1/1/65. YLAG 20
C  LAGRANGIAN INTERPOLATION YLAG 30
C  X1 IS INTERPOLATED ENTRY INTO X-ARRAY YLAG 40
C  N IS THE ORDER OF LAGRANGIAN INTERPOLATION YLAG 50
C  Y IS ARRAY FROM WHICH YLAG IS OBTAINED BY INTERPOLATION YLAG 60
C  IND IS THE MIN-I FOR X(I).GT.X1 YLAG 70
C  IF IND=0,X-ARRAY WILL BE SEARCHED YLAG 80
C  IMAX IS MAX INDEX OF X-AND Y-ARRAYS YLAG 90
C  EXTRAPOLATION CAN OCCUR,IEX=-1 OR +1 YLAG 100
C  DIMENSION X(1),Y(1) YLAG 110
      IND=IND1 YLAG 120
      N=N1 YLAG 130
      IEX=0 YLAG 140
      IF(N-IMAX) 10,10,5 YLAG 150
5      N=IMAX YLAG 160
      IEX=N YLAG 170
10      IF(IND)15,15,30 YLAG 180
J-----15      DO 20 J=1,IMAX YLAG 190
J              IF (X1-X(J)) 25,90,20 YLAG 200
J-----20      CONTINUE YLAG 210
              ICX=1 YLAG 220
              GO TO 55 YLAG 230
25      IND=J YLAG 240
30      IF (IND-1) 35,35,40 YLAG 250
25      ICX=-1 YLAG 260
40      INL=IND-(N+1)/2 YLAG 270
      IF (INL)45,45,50 YLAG 280
45      INL=1 YLAG 290
50      INU=INL+N-1 YLAG 300
      IF (INU-IMAX) 60, 60, 55 YLAG 310
55      INL=IMAX-N+1 YLAG 320
      INU=IMAX YLAG 330
60      S=0. YLAG 340
      P=1. YLAG 350
J-----      DO 80 J=INL,INU YLAG 360
J              P=P*(X1-X(J)) YLAG 370
J              D=1. YLAG 380
J I-----      DO 75 I=INL,INU YLAG 390
J I              IF (I-J)70,65,70 YLAG 400
J I              XD=XI YLAG 410
J I              GO TO 75 YLAG 420
J I              XD=X(J) YLAG 430
J I-----75      D=D*(XD-X(I)) YLAG 440
J-----80      S=S+Y(J)/D YLAG 450
              YLAG=S*P YLAG 460
85      RETURN YLAG 470
40      YLAG=Y(I) YLAG 480
              GO TO 85 YLAG 490
      END YLAG 500

```

TABLE OF VARIABLE NAMES

VARIABLE NAME	ARRAY DIMENSIONS	TOTAL NO ELEMENTS	ELEMENT TYPE	DECLARATION	CROSS REFERENCE TABLE (BY SOURCE NUMBERS)				
D			REAL*4	LOCAL	380*	440*	450		
I			INTEGER*4	LOCAL	390*	400,	440		
IEX			INTEGER*4	PARAMETER	10,	140*	170*	220*	260*
IMAX			INTEGER*4	PARAMETER	10,	150,	160,	190,	310,
INU			INTEGER*4	LOCAL	140*	180,	240*	250,	270
IND1			INTEGER*4	PARAMETER	10,	120			
INL			INTEGER*4	LOCAL	270*	280,	290*	300,	320*
INU			INTEGER*4	LOCAL	300*	310,	330*	360,	390
J			INTEGER*4	LOCAL	190*	200,	240,	360*	370,
N			INTEGER*4	LOCAL	450,	480		400,	430,
N1			INTEGER*4	PARAMETER	130*	150,	160*	170,	270,
P			REAL*4	LOCAL	300,	370*	460	300,	320
S			REAL*4	LOCAL	340*	450*	460		
X	(1)	1	REAL*4	PARAMETER	10,	110,	200,	370,	430,
XD			REAL*4	LOCAL	410*	430*	440		

XI		REAL*4	PARAMETER	10,	200,	370,	410
Y	(1)	1 REAL*4	PARAMETER	10,	110,	450,	480
YLAG		REAL*4	ENTRY POINT	10,	460*	480*	

* = VARIABLES USED ON LEFT HAND SIDE OF EQUAL SIGN

```

SUBROUTINE WOUT1(NUMX,XVAL,XWX)
DIMENSION XVAL(50),XWX(50)
5  I1=NUMX/4
  J1=I1
  J2=I1*2
  J3=I1*3
  WRITE(6,8000)
8000 FORMAT(1H0,20X,'DETECTOR SENSITIVITY*///')
  WRITE(6,8005)(XVAL(ML),XWX(ML),XVAL(ML+J1),XWX(ML+J1),XVAL(ML+J2),
1XWX(ML+J2),XVAL(ML+J3),XWX(ML+J3),ML=1,I1)
8005 FORMAT(1H0,8X,4HXVAL,11X,3HXWX,12X,4HXVAL,11X,4HXWX,11X,4HXVAL,
111X,3HXWX,12X,4HXVAL,11X,3HXWX,/(4(1X,F14.6,1X,F14.6)))
  NN=MOD(NUMX,4)
  IF(NN)10,15,10
10  NN=NUMX-NN+1
  WRITE(6,8010)(XVAL(J),XWX(J),J=NN,NUMX)
8010 FORMAT(4(1X,F14.6,1X,F14.6))
15  CONTINUE
  RETURN
END

```

```

WOUT1 10
WOUT1 20
WOUT1 30
WOUT1 40
WOUT1 50
WOUT1 60
WOUT1 70
WOUT1 80
WOUT1 90
WOUT1 91
WOUT1 100
WOUT1 101
WOUT1 110
WOUT1 120
WOUT1 130
WOUT1 140
WOUT1 150
WOUT1 160
WOUT1 170
WOUT1 180

```

TABLE OF VARIABLE NAMES

VARIABLE NAME	ARRAY DIMENSIONS	TOTAL NO ELEMENTS	ELEMENT TYPE	DECLARATION	CROSS REFERENCE TABLE (BY SEQUENCE NUMBERS)				
I1			INTEGER*4	LOCAL	30*	40,	50,	60,	90
J			INTEGER*4	LOCAL	140*				
J1			INTEGER*4	LOCAL	40*	90			
J2			INTEGER*4	LOCAL	50*	90			
J3			INTEGER*4	LOCAL	60*	90			
ML			INTEGER*4	LOCAL	90*				
MOD			INTEGER*4	FUNCTION	110				
NN			INTEGER*4	LOCAL	110*	120,	130*	140	
NUMX			INTEGER*4	PARAMETER	10,	30,	110,	130,	140
WOUT1			REAL*4	ENTRY POINT	10				
XVAL	(50)	50	REAL*4	PARAMETER	10,	20,	90,	140	
XWX	(50)	50	REAL*4	PARAMETER	10,	20,	90,	140	

* = VARIABLES USED ON LEFT HAND SIDE OF EQUAL SIGN

INPUT / OUTPUT TABLE

DEFINING SEQ. NO.	I/O OPERATION	LOGICAL UNIT	I/O TYPE	FORMAT USED	I/O LIST
70	WRITE	6	BLU	8000	
90	WRITE	6	BCD	8005	(XVAL(ML),XWX(ML),XVAL(ML+J1),XWX(ML+J1),XVAL(ML+J2),....
140	WRITE	6	BCD	8010	(XVAL(J),XWX(J),J=NN,NUMX)

```

SUBROUTINE WOUT2(NUMZ,ZVAL,ZWZ)
DIMENSION ZVAL(50),ZWZ(50)
5  I1=NUMZ/4
   J1=I1
   J2=I1*2
   J3=I1*3
   WRITE(6,8000)
8000 FORMAT(1H0,20X,'FOCAL SPOT INTENSITY'///)
   WRITE(6,8005)((ZVAL(ML),ZWZ(ML),ZVAL(ML+J1),ZWZ(ML+J1),ZVAL(ML+J2),ZWZ(ML+J2),
   12WZ(ML+J2),ZVAL(ML+J3),ZWZ(ML+J3),ML=1,I1)
8005 FORMAT(1H0,8X,4HZVAL,11X,3HZWZ,12X,4HZVAL,11X,3HZWZ,12X,4HZVAL,
   111X,3HZWZ,12X,4HZVAL,11X,3HZWZ/(4(1X,F14.6,1X,F14.6)))
   NN=MOD(NUMZ,4)
   IF(NN)10,15,10
10  NN=NUMZ-NN+1
   WRITE(6,8010)((ZVAL(J),ZWZ(J),J=NN,NUMZ)
8010 FORMAT(4(1X,F14.6,1X,F14.6))
15  CONTINUE
   RETURN
END

```

```

WOUT2 10
WOUT2 20
WOUT2 30
WOUT2 40
WOUT2 50
WOUT2 60
WOUT2 70
WOUT2 80
WOUT2 90
WOUT2 91
WOUT 100
WOUT 101
WOUT 110
WOUT 120
WOUT 130
WOUT 140
WOUT 150
WOUT 160
WOUT 170
WOUT 180

```

TABLE OF VARIABLE NAMES

VARIABLE NAME	ARRAY DIMENSIONS	TOTAL NO ELEMENTS	ELEMENT TYPE	DECLARATION	CROSS REFERENCE TABLE (BY SEQUENCE NUMBERS)				
I1			INTEGER*4	LOCAL	30*	40,	50,	60,	90
J			INTEGER*4	LOCAL	140*				
J1			INTEGER*4	LOCAL	40*	90			
J2			INTEGER*4	LOCAL	50*	90			
J3			INTEGER*4	LOCAL	60*	90			
ML			INTEGER*4	LOCAL	90*				
MOD			INTEGER*4	FUNCTION	110				
NN			INTEGER*4	LOCAL	110*	120,	130*	140	
NUMZ			INTEGER*4	PARAMETER	10,	30,	110,	130,	140
WOUT2			REAL*4	ENTRY POINT	10				
ZVAL	(50)	50	REAL*4	PARAMETER	10,	20,	90,	140	
ZWZ	(50)	50	REAL*4	PARAMETER	10,	20,	90,	140	

* = VARIABLES USED ON LEFT HAND SIDE OF EQUAL SIGN

INPUT / OUTPUT TABLE

DEFINING SEQ. NO.	I/O OPERATION	LOGICAL UNIT	I/O TYPE	FORMAT USED	I/O LIST
70	WRITE	6	BCD	8000	
90	WRITE	6	BCD	8005	(ZVAL(ML),ZWZ(ML),ZVAL(ML+J1),ZWZ(ML+J1),ZVAL(ML+J2),...
140	WRITE	6	BCD	8010	(ZVAL(J),ZWZ(J),J=NN,NUMZ)

```

SUBROUTINE TIMER(INIT)
IF(INIT .EQ. 0) GO TO 5
TIME=(FLOAT(ICLOCK(0))-TMP)/100.
WRITE(6,8000) TIME
8000 FORMAT(1H0//10X,'TIME = ',F6.2,' CPU SECONDS'///)
IF(INIT .LT. 0) GO TO 10
5   TMP=FLOAT(ICLOCK(0))
10  RETURN
END

```

```

TIMER 10
TIMER 20
TIMER 30
TIMER 40
TIMER 50
TIMER 60
TIMER 70
TIMER 80
TIMER 90

```

TABLE OF VARIABLE NAMES

VARIABLE NAME	ARRAY DIMENSIONS	TOTAL NO ELEMENTS	ELEMENT TYPE	DECLARATION	CROSS REFERENCE TABLE (BY SEQUENCE NUMBERS)	
FLUAY			REAL*4	FUNCTION	20,	70
ICLOCK			INTEGER*4	FUNCTION	30,	70
INIT			INTEGER*4	PARAMETER	10,	20, 60
TIME			REAL*4	LOCAL	30*	40
TIMER			REAL*4	ENTRY POINT	10	
TMP			REAL*4	LOCAL	30,	70*

* = VARIABLES USED ON LEFT HAND SIDE OF EQUAL SIGN

INPUT / OUTPUT TABLE

DEFINING SEQ. NU.	I/O OPERATION	LOGICAL UNIT	I/O TYPE	FORMAT USED	I/O LIST
40	WRITE	6	BCD	8000	TIME

APPENDIX B
FLOW CHART OF PROGRAM WEIGHT

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

ORNL-DWG 67-7176

APPENDIX C
DEFINITION OF SELECTED PROGRAM VARIABLES

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

<u>Variable</u>	<u>Description</u>
AANS(202)	Array containing values of the normalization factor. The last element in the array is the most accurate value.
AMOMNT(202)	Array used in calculating moments of the function.
AREAB	Area under $f_w(z)$ between $z_a(u,x)$ and next highest value of SIMPZ.
AREAT	Area under $f_w(z)$ between $z_b(x,u)$ and next lowest value of ZIMPZ.
EWX(1001)	Array containing calculated values of $e_w(x)$.
FIRSTM	First moment about the origin.
FRSTMM	First moment about the mean.
FWZ(1001)	Array containing calculated values of $f_w(z)$.
FWZINB	Interpolated value of $f_w(z)$ corresponding to $z_a(x,u)$.
FWZINT	Interpolated value of $f_w(z)$ corresponding to $z_b(x,u)$.
G(8)	Array containing h_j .
H(8)	Array containing h_i .
IEX	Flag indicating whether or not x and $e_w(x)$ values are given as data.
IFLAG	Computed flag indicating whether the input data corresponds to pairs of edges with each member of the pair equidistant from the sample plane.
IFZ	Flag indicating whether or not z and $f_w(z)$ values are given as data.
IOUT	Flag indicating form of output.
JJ	Flag indicating whether or not moments about the mean are to be calculated.
KK	Flag indicating whether or not moments about the origin are to be calculated.
M	Number of pairs of slits between sample plane and tube plane.
N	Number of pairs of slits between sample plane and detector plane.
NU	Number values of u .
NUMX	Number of pairs of x and $e_w(x)$ values given as data.
NUMZ	Number of pairs of z and $f_w(z)$ values given as data.
SD(8)	Array containing d_i .
SECNDM	Second moment about the origin.

<u>Variable</u>	<u>Description</u>
SCNDMM	Second moment about the mean.
SIMPX(1001)	Array containing x values that corresponds to $e_w(x)$.
SIMPZ(1001)	Array containing z values that corresponds to $f_w(z)$.
TD(8)	Array containing d_j .
TEMPWC	Contains $-w_c$.
U(8,8)	Values of u_{ij} calculated at $x = w_c$ and $x = -w_c$ used in calculation of the u limits u_{aaa} and u_{bbb} .
UEE(8)	$u_{2i,2i'-1}$.
UDIF(202)	Array containing values of $(u_i - \bar{u})^k * W_w(u)$ for each value of u.
UIE(8)	$u_{2i;2j-1}$.
UINC	u increment.
UIO(8)	$u_{2i-1,2j}$.
UOO(8)	$u_{2i-1,2i'}$.
USTORE(202)	Array containing values of $u^k * W_w(u)$ for each value of u.
UVAL(202)	u array.
WC	w_c .
WWU(202)	Array containing normalized $W_w(u)$ for all u values.
WWUF(202)	Array containing unnormalized $W_w(u)$ values for all u values.
X(8,8)	x values used in calculating x limits.
XANS(202)	Array containing results of x integration; last element of the array contains most accurate answer.
XAU(202)	$x_u(u)$.
XBU(202)	$x_b(u)$.
XIE(8)	$x_{2i;2j-1}$.
XINC	x increment.
XINT(202)	Array containing the integrand for x integration; $XINT(I) = EWX(I) * ZANS(I)$.
XIO(8)	$x_{2i-1;2j}$.

<u>Variable</u>	<u>Description</u>
XMAXD	Variable containing maximum x.
XMIND	Variable containing minimum x.
XRAY(1001)	Array containing x values for the integration from XAU to XBU.
XVAL(50)	x values given as data.
XWX(50)	$e_w(x)$ values given as data.
XXE(8)	$x_{2i,2i'-1}$.
XXO(8)	$x_{2i-1,2i'}$.
ZANS(202)	Array containing results of z integration; last element contains most accurate answer.
ZAU(202)	$z_a(x,u)$.
ZBU(202)	$z_b(x,u)$.
ZI(8)	z_i : z values calculated using d_i and h_i .
ZIE(8)	z_{2i} .
ZINT(I)	Array containing the integral of FWZ from SIMPZ(1) to SIMPZ(I).
ZINCR	Increment of z integration.
ZIO(8)	z_{2i-1} .
ZJ(8)	z_j : z values calculated using d_j and h_j .
ZJO(8)	z_{2j-1} .
ZVAL(50)	Array containing z values given as data.
ZWZ(50)	Array containing $f_w(z)$ values given as data.

**THIS PAGE
WAS INTENTIONALLY
LEFT BLANK**

APPENDIX D
INPUT VARIABLES

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

<u>Variable</u>	<u>Description</u>
EL	The distance from the sample plane to the detector plane; equivalent to L .
ELO	The distance from the sample plane to the tube plane; equivalent to L_0 .
G(J)	The distance from the j^{th} slit to the camera axis; equivalent to h_j .
H(I)	The distance from the i^{th} slit to the camera axis; equivalent to h_i .
IEX	Flag indicating whether or not x and $e_w(x)$ supplied as data; for IEX = 0, set $e_w(x) = 1$ for all values of x . When IEX $\neq$ 0, values of x and $e_w(x)$ will be read in as data.
IFZ	Flag indicating whether or not z and $f_w(z)$ supplied as data; for IFZ = 0, set $f_w(z) = 1$ for all values of z . When IFZ $\neq$ 0, values of z and $f_w(z)$ will be read in as data.
IOUT	Flag indicating whether or not a punched deck of the final u and $W_w(u)$ values is to be punched.
JJ	Flag indicating whether or not moments about the mean are to be calculated. JJ = 0, do not calculate moments; JJ = 1, calculate 1st moment; JJ = 2, calculate 1st and 2nd moments; JJ = 3, calculate 1st, 2nd, and 3rd moments. 3 is the maximum value of JJ.
KK	Flag indicating whether or not moments about origin are to be calculated. If KK = 0, JJ must also be 0. KK = 0, do not calculate moments; KK = 1, calculate 1st moment; KK = 2, calculate 1st and 2nd moments; KK = 3, calculate 1st, 2nd, and 3rd moments. 3 is the maximum value of KK.
KOUT	Flag indicating whether or not the values of u , $x_a(u)$, $x_b(u)$ versus $z_a(x,u)$ and $z_b(x,u)$ are printed out. This information is helpful in error analysis. KOUT = 0, no printout; KOUT $\neq$ 0, printout.
M	Number of pairs of slits between sample plane and tube plane; $1 \leq M \leq 4$.
N	Number of pairs of slits between sample plane and detector plane; $1 \leq N \leq 4$.
NU	Number of u values to be calculated; NU - 1 is the number of intervals to be used in evaluating the integrals. NU $\leq$ 200.
NUMX	Number of pairs of x and $e_w(x)$ supplied as data.
NUMZ	Number of pairs of z and $f_w(z)$ supplied as data.

<u>Variables</u>	<u>Description</u>
SD(I)	Array of distances from sample plane to the i^{th} slit between the sample plane and the tube plane; $1 \leq I \leq 2*M$.
TD(J)	Array of distances from sample plane to the j^{th} slit between sample plane and detector plane; $1 \leq J \leq 2*N$.
X(NUMX)	Array of x values.
XWX(NUMX)	Array of $e_w(x)$ values.
WC	One-half width of receiving slit.
Z(NUMZ)	Array of z values.
ZWZ(NUMZ)	Array of $f_w(z)$ values.

Table D.1. Format of Information Shown on Data Cards

Card Number ^a	Format	Variables
1	80A1	TITLE
2	12I5	M, N, NU, KK, JJ, IOUT, IFZ, IEX, NUMZ, NUMX, KOUT, IPLOT
3	3F10.4	EL, ELO, WC
4	8F10.3	SD(I), where I = 1,2, ..., 2*M
5	8F10.3	H(I)
6	8F10.3	TD(J), where J = 1,2, ..., 2*N
7	8F10.3	G(J)
8-20	8F10.3	Z(K), ZWZ(K) alternately; K = 1,2, ..., NUMZ
21-32	8F10.3	X(K), XWX(K) alternately; K = 1,2, ..., NUMX

^aCards 8-32 are not necessary if IFZ and IEX are both 0. Provision has been made for 50 values each of X, XWX, Z, and ZWZ. If the user requires fewer than this number, the number of data cards required for the values can be reduced. For example, if NUMZ = 25, there would be 25 Z values and 25 ZWZ values. Therefore, the Z and ZWZ values would only occupy cards 8-14 and the X and XWX values would be started in data card number 14. If Z and ZWZ values are not read in as data, the X and XWX values start in data card number 8. Reading Z, ZWZ, X, and XWX values in as data is optional. The only restriction is that if Z values are fed in, ZWZ values must be furnished; the same applies for X and XWX.

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

APPENDIX E
SAMPLE CALCULATIONS

Example 1: Computations for a typical Kratky camera geometry. Note the use of dummy edges at $H(I)$ for $I = 2$ and $I = 3$, and at $H(J)$ for $J = 1$ and $J = 2$. There were no $e_w(x)$ or $f_w(z)$ input data.

Example 2: Same as example 1 except with $e_w(x)$ and $f_w(z)$ values read in as input data.

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

EXAMPLE 1

INPUT DATA

M=4 N=1 NU=101 KK=1 JJ=2 IQUT=0 IFZ=0
 IEX=0 NUMZ= 0 NUMX= 0 KOUT=0
 EL=262.0000 ELO=281.0000 WC= 0.3500

SUBSCRIPT I	SUBSCRIPT J
D(1)= 63.000	D(1)= 10.000
D(2)= 63.000	D(2)= 10.000
D(3)= 78.000	D(3)=*****
D(4)= 78.000	D(4)=*****
E(5)=138.000	D(5)=*****
D(6)=138.000	D(6)=*****
D(7)=198.000	D(7)=*****
D(8)=198.000	D(8)=*****
H(1)= 0.0	H(1)=999.000
H(2)=999.000	H(2)=999.000
H(3)= 0.0	H(3)=*****
H(4)=999.000	H(4)=*****
F(5)=999.000	H(5)=*****
H(6)= 0.0	H(6)=*****
H(7)= 0.200	H(7)=*****
F(8)= 0.0	H(8)=*****

KRATKY CAMERA FULLY ILLUMINATED ENT SLIT WC=0.350

U	WW(U)	U	WW(U)
-4.961871	0.0	0.190790	0.21251E 00
-4.858817	0.16050F-03	0.293842	0.21022E 00
-4.755764	0.64201E-03	0.296894	0.20737E 00
-4.652711	0.14445E-02	0.499948	0.20396E 00
-4.549657	0.25680E-02	0.603000	0.19998E 00
-4.446604	0.40126E-02	0.706053	0.19545E 00
-4.343551	0.57781E-02	0.809106	0.19037E 00
-4.240498	0.78646E-02	0.912158	0.18472E 00
-4.137444	0.10272E-01	0.912158	0.17862E 00
-4.034390	0.13001E-01	1.015211	0.17247E 00
-3.931337	0.16050E-01	1.118263	0.16631E 00
-3.828282	0.19421E-01	1.221316	0.16016E 00
-3.725220	0.23112E-01	1.324149	0.15401E 00
-3.622177	0.27125E-01	1.427422	0.14786E 00
-3.519124	0.31458E-01	1.530475	0.14171E 00
-3.416070	0.36113E-01	1.633527	0.13556E 00
-3.313017	0.41089E-01	1.736580	0.12941E 00
-3.209964	0.46288E-01	1.839633	0.12326E 00
-3.106911	0.52003E-01	1.942685	0.11710E 00
-3.003858	0.57941E-01	2.045728	0.11095E 00
-2.900805	0.64201E-01	2.148790	0.10480E 00
-2.797751	0.70782E-01	2.251842	0.98651E-01
-2.694698	0.77683E-01	2.354896	0.92500E-01
-2.591645	0.84906E-01	2.457949	0.86348E-01
-2.488592	0.92450E-01	2.561002	0.80197E-01
-2.385539	0.10031E 00	2.664054	0.74148E-01
-2.282486	0.10850E 00	2.767107	0.68334E-01
-2.179433	0.11682E 00	2.870159	0.62758E-01
-2.076380	0.12514E 00	2.973212	0.57420E-01
-1.973327	0.13347E 00	3.076265	0.52319E-01
-1.870274	0.14179E 00	3.179317	0.47455E-01
-1.767220	0.15011E 00	3.282371	0.42828E-01
-1.664167	0.15836E 00	3.385423	0.38438E-01
-1.561114	0.16611E 00	3.488476	0.34286E-01
-1.458061	0.17320E 00	3.591529	0.30371E-01
-1.355007	0.17994E 00	3.694581	0.26693E-01
-1.251954	0.18602E 00	3.797634	0.23253E-01
-1.148901	0.19154E 00	3.900686	0.20050E-01
-1.045848	0.19651E 00	4.003739	0.17084E-01
-0.942795	0.20147E 00	4.106792	0.14255E-01
-0.839742	0.20476E 00	4.209845	0.11800E-01
-0.736689	0.20804E 00	4.312898	0.96099E-02
-0.633636	0.21077E 00	4.415950	0.75931E-02
-0.530582	0.21294E 00	4.519003	0.58134E-02
-0.427529	0.21458E 00	4.622056	0.42711E-02
-0.324476	0.21561E 00	4.725108	0.29663E-02
-0.221423	0.21611E 00	4.828161	0.18986E-02
-0.118370	0.21604E 00	4.931213	0.10679E-02
-0.015316	0.21542E 00	5.034266	0.71680E-03
0.087737	0.21425E 00	5.137320	0.42711E-02
		5.240372	0.21870E-03
		5.343425	0.28951E-10

NORMALIZATION FACTOR AW= 0.73762E-01

MOMENTS OF THE FUNCTION ABOUT THE ORIGIN

1ST = 0.26260E 01 2ND = 0.0 3RD = 0.0

THEORETICAL LIMIT OF RESOLUTION = 288.5 ANG.
OR = 5.34 MR.

MOMENTS OF THE FUNCTION ABOUT THE MEAN

1ST = -0.29599E-04 2ND = 0.30411E 01 3RD = 0.0

TIME = 12.60 CPU SECONDS

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

EXAMPLE 2

INPUT DATA

M=4 N=1 NU=101 KK=1 JJ=2 IOUT=0 IFZ=1
 IEX=0 NUMZ= 10 NUMX= 0 KOUT=0
 EL=262.0000 ELO=281.0000 WC= 0.3500

SUBSCRIPT I

D(1)= 63.000
 D(2)= 63.000
 D(3)= 78.000
 D(4)= 78.000
 C(5)=138.000
 C(6)=138.000
 D(7)=198.000
 D(8)=198.000
 H(1)= 0.0
 H(2)=999.000
 H(3)= 0.0
 H(4)=999.000
 H(5)=999.000
 H(6)= 0.0
 H(7)= 0.300
 H(8)= 0.0

SUBSCRIPT J

D(1)= 10.000
 D(2)= 10.000
 D(3)=*****
 D(4)=*****
 D(5)=*****
 D(6)=*****
 D(7)=*****
 D(8)=*****
 H(1)=999.000
 H(2)=999.000
 H(3)=*****
 H(4)=*****
 H(5)=*****
 H(6)=*****
 H(7)=*****
 H(8)=*****

FOCAL SPOT INTENSITY

ZVAL	ZWZ	ZVAL	ZWZ	ZVAL	ZWZ	ZVAL	ZWZ
-0.100000	0.0	0.305000	1.000000	0.376000	0.400000	0.515000	1.000000
0.304000	0.0	0.375000	1.000000	0.514000	0.400000	0.585000	1.000000
0.586000	0.0	0.800000	0.0				

KRATKY CAMERA NONFULLY ILLUMINATED ENT SLIT WC=0.350 FS DEPRESSED

U	WW(U)	U	WW(U)
-5.301841	0.0	-0.149179	0.28933E 00
-5.198787	0.0	-0.046127	0.28967E 00
-5.095734	0.0	0.056925	0.28880E 00
-4.992681	0.0	0.159978	0.28674E 00
-4.889627	0.0	0.263030	0.28330E 00
-4.786573	0.0	0.366083	0.27803E 00
-4.683520	0.0	0.469136	0.27089E 00
-4.580467	0.0	0.572188	0.26188E 00
-4.477413	0.0	0.675241	0.25125E 00
-4.374360	0.0	0.778294	0.24020E 00
-4.271307	0.0	0.881347	0.22871E 00
-4.168253	0.0	0.984400	0.21671E 00
-4.065200	0.0	1.087452	0.20502E 00
-3.962147	0.0	1.190505	0.19388E 00
-3.859094	0.0	1.293557	0.18331E 00
-3.756040	0.0	1.396610	0.17328E 00
-3.652987	0.0	1.499663	0.16336E 00
-3.549933	0.0	1.602715	0.15331E 00
-3.446880	0.0	1.705760	0.14315E 00
-3.343827	0.12548E-04	1.808821	0.13287E 00
-3.240774	0.11487E-02	1.911874	0.12247E 00
-3.137721	0.41557E-02	2.014926	0.11195E 00
-3.034668	0.90371E-02	2.117979	0.10132E 00
-2.931615	0.15793E-01	2.221032	0.90565E-01
-2.828562	0.24134E-01	2.324084	0.79784E-01
-2.725509	0.33276E-01	2.427137	0.69404E-01
-2.622456	0.43148E-01	2.530190	0.59462E-01
-2.519403	0.53776E-01	2.633242	0.49971E-01
-2.416349	0.65152E-01	2.736296	0.40923E-01
-2.313296	0.77282E-01	2.839348	0.32341E-01
-2.210243	0.90169E-01	2.942401	0.24394E-01
-2.107190	0.10381E 00	3.045453	0.17560E-01
-2.004137	0.11801E 00	3.148506	0.11845E-01
-1.901084	0.13155E 00	3.251559	0.72480E-02
-1.798031	0.14440E 00	3.354611	0.37797E-02
-1.694977	0.15669E 00	3.457664	0.14282E-02
-1.591924	0.16843E 00	3.560717	0.19935E-03
-1.488871	0.17961E 00	3.663770	0.0
-1.385818	0.19022E 00	3.766823	0.0
-1.282764	0.20029E 00	3.869875	0.0
-1.179711	0.21030E 00	3.972928	0.0
-1.076658	0.22022E 00	4.075980	0.0
-0.973605	0.23037E 00	4.179033	0.0
-0.870552	0.24064E 00	4.282086	0.0
-0.767499	0.25103E 00	4.385139	0.0
-0.664446	0.26152E 00	4.488192	0.0
-0.561393	0.27108E 00	4.591244	0.0
-0.458340	0.27846E 00	4.694297	0.0
-0.355286	0.28424E 00	4.797350	0.0
-0.252233	0.28777E 00	4.900402	0.0
		5.003455	0.0

NORMALIZATION FACTOR $\Delta W = 0.31507E-01$

MOMENTS OF THE FUNCTION ABOUT THE ORIGIN

1ST = 0.39660E 01 2ND = 0.0 3RD = 0.0

THEORETICAL LIMIT OF RESOLUTION = 433.0 ANG.
OR = 3.56 MR.

MOMENTS OF THE FUNCTION ABOUT THE MEAN

1ST = -0.20634E-04 2ND = 0.16539E 01 3RD = 0.0

TIME = 54.96 CPU SECONDS

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

ORNL-TM-1950
(Rev. 1)

INTERNAL DISTRIBUTION

1-3.	Central Research Library	69.	J. H Frye, Jr.
4.	ORNL - Y-12 Technical Library	70.	Anne B. Gordon (K-25)
	Document Reference Section	71-73.	R. W. Henderson (K-25)
5-60.	Laboratory Records Department	74-123.	R. W. Hendricks
61.	Laboratory Records, ORNL RC	124-126.	M. R. Hill
62.	ORNL Patent Office	127.	C. K. Johnson
63.	G. M. Adamson, Jr.	128.	H. A. Levy
64.	B. S. Borie	129.	C. J. McHargue
65.	W. R. Busing	130.	D. A. Sundberg
66.	J. E. Cunningham	131.	D. B. Trauger
67.	F. L. Culler	132.	H. L. Yakel
68.	J. E. Epperson		

EXTERNAL DISTRIBUTION

- 133. D. F. Cope, RDT, SSR, AEC, Oak Ridge National Laboratory
- 134. D. K. Stevens, Division of Research, AEC, Washington
- 135. M. J. Whitman, Division of Reactor Development and Technology,
AEC, Washington
- 136. Laboratory and University Division, AEC, Oak Ridge Operations
- 137-151. Division of Technical Information Extension