Natural Science Report, Ochanomizu University, Vol. 40, No. 1, 1989
お茶の水女子大学 自然科学報告 第40巻 第1号

An Extension of Hesse-Rübartsch Method for the Determination of Two-parameter Distribution

Natsuki Hashitsume

Department of Physics, Faculty of Science,
Ochanomizu University

and

Hiroyuki Komatsu

Doctoral Course, Faculty of Engineering,
Tohoku University
(Received April 5, 1989)

To determine the hyperfine field distribution in a sample from measured overlapped Mössbauer spectra, Hesse and Rübartsch proposed a calculation scheme. This scheme is extended to determine a distribution of arbitrary two parameters and to be applicable to any other parameter distribution analysis in addition to the Mössbauer spectrum analysis. An example FORTRAN program is given for the Mössbauer spectrum analysis.

§1. Introducnion

In many problems in Mössbauer and other spectroscopies, experimentally determined spectra sometimes appear as broadened because of the overlap of lines due to distributions in physical parameters. B. Window proposed in 1971[1] a method to determine the hyperfine field distribution from observed Mössbauer spectra. In his method the distribution $P(H)$ is assumed to be an even function of the field H, and is expanded in terms of trigonometric functions in an interval $(-H_{\max}\leqq H\leqq H_{\max})$

$$\cos(n\pi H/H_{\max})-(-1)^n, \qquad (n=1, 2, \cdots, N) \tag{1.1}$$

and N coefficients of expansion are determined by the method of least square. The second term $(-1)^n$ in (1.1) is added to satisfy the boundary conditions $P(H_{\max})=0$. The basic functions (1.1) with even n's are negative, although $P(H)$ should be positive, but those are necessary to expand $P(H)$ such as symmetric about the middle point in the positive half range $(0, H_{\max})$ which he considered. If $P(0)$ should not vanish, the functions (1.1) with odd n's are necessary.

In 1974 J. Hesse and A. Rübartsch[2] proposed another method to calculate the parameter distribution $P(H)$. They discretized directly the function $P(H)$ considering only its special values $P_i=P(H_i)$, $i=1, 2, \cdots, N$, and tried to determine P_i's instead of Window's expansion coefficients. Furthermore, in order to avoid non-smooth or jagged field-dependence, which could be obtained by the direct calculation of P_i's, they introduced a modification of the least square method. Their basic idea came from the recommendations given for the numerical solution of integral equations, and is the addition of the sum of squares of curvatures $P_{i+1}-2P_i+P_{i-1}$, multiplied by a suitably chosen penalty γ, to the usual sum of squared error terms to be minimized. In Window's method the smoothness of the distribution was automatically guarantied by using a finite number N of cosines.

In this article we shall try to extend the Hesse-Rübartsch method determining a single parameter distribution to a method for a distribution depending on two parameters. The basic idea is the use of 2-dimensional "curvature" $P_{i,j+1}+P_{i,j-1}+P_{i+1,j}+P_{i-1,j}-4P_{i,j}$ in place of 1-dimensional $P_{i+1}+P_{i-1}-2P_i$, i.e. that a central value must not much differ from the average of its nearest neighbours. Although in the 1-dimensional case there are only two boundary points ($i=0$ and N), in the 2-dimensional case there are two kinds of boundary points, i.e. four corner points and points on four sides. This fact makes the added penalty term very complicated, and the further extension of the present method to 3 or more dimensional cases hopelessly tedious, if we do not use some softwares for algebraic computation.

In § 2 the formulation in the 2-dimensional case is made, and in § 3 an example FORTRAN program for the Mössbauer spectrum analysis is given. The graphic part of the program is machine dependent; here we assume MicroVAX II minicomputer.

§ 2. Method for the determination of parameter distribution

Two parameters A and B are assumed to have a 2-dimensional distribution function $P(A, B)$, and for each pair of values of A and B a line-shape function $Y_{\mathrm{cal}}(A, B, C)$ is assumed to be known, where C distinguishes the frequencies or the wave-numbers or other parameters, against which the spectral intensities are measured. Thus the theoretical spectral form is given in the form

$$Y_{\mathrm{theor}}(C)=\iint P(A, B)Y_{\mathrm{cal}}(A, B, C)dAdB. \qquad (2.1)$$

According to the method of least square, the distribution function $P(A, B)$ is determined by the minimum principle

$$\int\{Y_{\mathrm{theor}}(C)-Y_{\mathrm{obs}}(C)\}^2dC=\mathrm{minimum}. \qquad (2.2)$$

where $Y_{obs}(C)$ denotes the observed value of the intensity at C.

After Hesse and Rübartsch, we discretize A, B and C and number their possible values by integers IPA, IPB and IC, respectively: $IPA=1, 2, \cdots, NPA$; $IPB=1, 2, \cdots, NPB$; $IC=1, 2, \cdots, NC$, IC being the channel number used in measurements. Moreover, we arrange the 2-dimensional array $P(IPA, IPB)$ into a 1-dimensional one

$$P(IP)=P(IPA, IPB), \qquad IP=1, 2, \cdots, NP, \tag{2.3}$$

where we have put $NP=NPA \cdot NPB$ and

$$IP=IPB+NPB \cdot (IPA-1). \tag{2.4}$$

This is for the sake of unified treatment, by which the original single parameter Hesse-Rübartsch method can be included into our double parameter extension. When we make the fitting of an observed spectrum, we often need to estimate the background value of spectrum, so that we introduce one more parameter

$$P(NP+1)=\text{background}, \tag{2.5}$$

and replace (2.1) with the following

$$Y_{theor}(IC)=P(NP+1)-\sum_{IP=1}^{NP} P(IP)Y_{cal}(IP, IC). \tag{2.6}$$

Here we have changed the sign of Y_{cal} for the later convenience in the example program, i.e. for treating the absorption spectrum.

The minimum principle (2.2) is replaced by

$$\sum_{IC=1}^{NC} \{Y_{obs}(IC)-Y_{theor}(IC)\}^2+\gamma \Delta(P)=\text{minimum}, \tag{2.7}$$

where $\Delta(P)$ denotes the sum of squares of "curvatures" and is composed of three terms

$$\Delta(P)=\Delta_1(P)+\Delta_2(P)+\Delta_3(P). \tag{2.8}$$

The first term $\Delta_1(P)$ is the contributions coming from the points on the first row $IPA=1$; $IPB=1, 2, \cdots, NPB$ in the 2-dimensional array shown in Fig. 1:

$$\begin{aligned}\Delta_1(P)=&\{-2P(1)+P(2)+P(NPB+1)\}^2\\ &+\sum_{IPB=2}^{NPB-1}\{P(IPB-1)-3P(IPB)+P(IPB+1)+P(NPB+IPB)\}^2\\ &+\{P(NPB-1)-2P(NPB)+P(NPB \cdot 2)\}^2\end{aligned} \tag{2.9}$$

The second contribution $\Delta_2(P)$ comes from intermediate rows $IPA=2, 3, \cdots, NPA-1$:

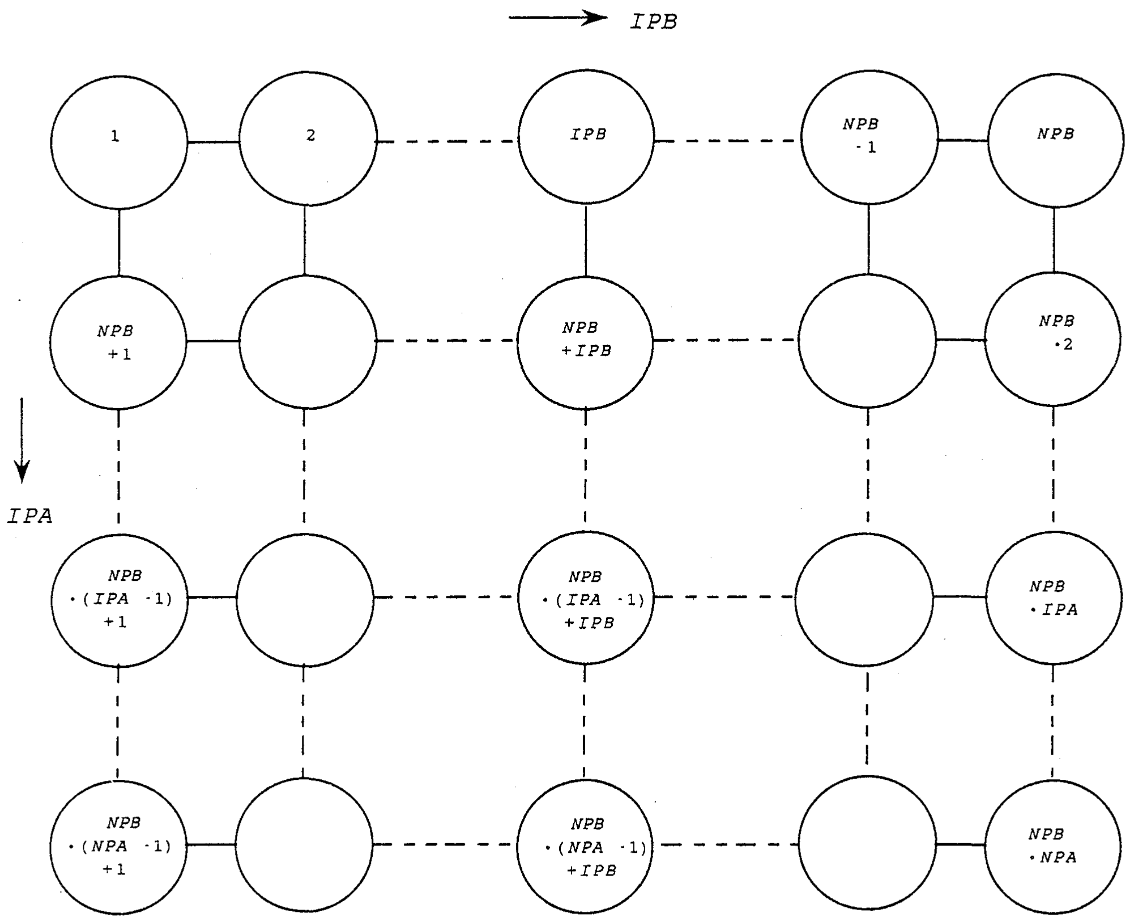

Fig. 1. 2-dimensional array of parameters.

$$\begin{aligned}\Delta_2(P)=\sum_{IPA=1}^{NPA-1}[&\{P(NPB[IPA-2]+1)-3P(NPB[IPA-1]+1)\\&+P(NPB[IPA-1]+2)+P(NPB\cdot IPA+1)\}^2\\&+\sum_{IPB=2}^{NPB-1}\{P(NPB[IPA-2]+IPB)+P(NPB[IPA-1]+IPB-1)\\&-4P(NPB[IPA-1]+IPB)\\&+P(NPB[IPA-1]+IPB+1)+P(NPB\cdot IPA+IPB)\}^2\\&+\{P(NPB[IPA-1])+P(NPB\cdot IPA-1)\\&-3P(NPB\cdot IPA)+P(NPB[IPA+1])\}^2].\end{aligned} \tag{2.10}$$

The last contribution $\Delta_3(P)$ comes from the lowest row $IPA=NPA$:

$$\begin{aligned}\Delta_3(P)=&\{P(NPB[NPA-2]+1)-2P(NPB[NPA-1]+1)\\&+P(NPB[NPA-1]+2)\}^2\\&+\sum_{IPB=2}^{NPB-1}\{P(NPB[NPA-2]+IPB)+P(NPB[NPA-1]+IPB-1)\\&-3P(NPB[NPA-1]+IPB)+P(NPB[NPA-1]+IPB+1)\}^2\end{aligned}$$

$$+\{P(NPB[NPA-1])+P(NPB\cdot NPA-1)-2P(NPB\cdot NPA)\}^2. \quad (2.11)$$

The normal equations for determining the distribution P are derived by differentiating the left-hand side of (2.7) with respect to the parameters $P(IP)$ as usual, and written in the form

$$\sum_{IP=1}^{NP}\left\{\sum_{IC=1}^{NC} Y_{\mathrm{cal}}(K, IC)Y_{\mathrm{cal}}(IP, IC)+\gamma D(K, IP)\right\}P(IP)$$

$$-\left\{\sum_{IC=1}^{NC} Y_{\mathrm{cal}}(K, IC)\right\}P(IP)=-\sum_{IC=1}^{NC} Y_{\mathrm{cal}}(K, IC)Y_{\mathrm{obs}}(IC),$$

$$(K=1, 2, \cdots, NP), \quad (2.12a)$$

$$-\sum_{IP=1}^{NP}\left\{\sum_{IC=1}^{NC} Y_{\mathrm{cal}}(IP, IC)\right\}P(IP)+NC\cdot P(NP+1)=\sum_{IC=1}^{NC} Y_{\mathrm{obs}}(IC). \quad (2.12b)$$

The mataix elements $D(K, L)$, $K, L=1, 2, \cdots, NP$, are determined after elementary but long tedious calculations. NPA and NPB must not be less than 5 for our construction of $\Delta(P)$. The structure of the matrix D can be seen in the smallest case of $NPA=NPB=5$, as shown in Fig. 2. The upper left and the lower right corner blocks of the matrix are special, and the other blocks are repeated on the diagonal direction.

The normal equations (2.12a) and (2.12b) can be solved easily by using the subroutines provided with computers.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	6	-5	1	0	0	-5	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	-5	12	-6	1	0	2	-7	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	-6	12	-6	1	0	2	-7	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
4	0	1	-6	12	-5	0	0	2	-7	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	1	-5	6	0	0	0	2	-5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
6	-5	2	0	0	0	12	-7	1	0	0	-6	2	0	0	0	1	0	0	0	0	0	0	0	0	0
7	2	-7	2	0	0	-7	20	-8	1	0	2	-8	2	0	0	0	1	0	0	0	0	0	0	0	0
8	0	2	-7	2	0	1	-8	20	-8	1	0	2	-8	2	0	0	0	1	0	0	0	0	0	0	0
9	0	0	2	-7	2	0	1	-8	20	-7	0	0	2	-8	2	0	0	0	1	0	0	0	0	0	0
10	0	0	0	2	-5	0	0	1	-7	12	0	0	0	2	-6	0	0	0	0	1	0	0	0	0	0
11	1	0	0	0	0	-6	2	0	0	0	12	-7	1	0	0	-6	2	0	0	0	1	0	0	0	0
12	0	1	0	0	0	2	-8	2	0	0	-7	20	-8	1	0	2	-8	2	0	0	0	1	0	0	0
13	0	0	1	0	0	0	2	-8	2	0	1	-8	20	-8	1	0	2	-8	2	0	0	0	1	0	0
14	0	0	0	1	0	0	0	2	-8	2	0	1	-8	20	-7	0	0	2	-8	2	0	0	0	1	0
15	0	0	0	0	1	0	0	0	2	-6	0	0	1	-7	12	0	0	0	2	-6	0	0	0	0	1
16	0	0	0	0	0	1	0	0	0	0	-6	2	0	0	0	12	-7	1	0	0	-5	2	0	0	0
17	0	0	0	0	0	0	1	0	0	0	2	-8	2	0	0	-7	20	-8	1	0	2	-7	2	0	0
18	0	0	0	0	0	0	0	1	0	0	0	2	-8	2	0	1	-8	20	-8	1	0	2	-7	2	0
19	0	0	0	0	0	0	0	0	1	0	0	0	2	-8	2	0	1	-8	20	-7	0	0	2	-7	2
20	0	0	0	0	0	0	0	0	0	1	0	0	0	2	-6	0	0	1	-7	12	0	0	0	2	-5
21	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	-5	2	0	0	0	6	-5	1	0	0
22	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	2	-7	2	0	0	-5	12	-6	1	0
23	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	2	-7	2	0	1	-6	12	-6	1
24	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	2	-7	2	0	1	-6	12	-5
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	2	-5	0	0	1	-5	6

Fig. 2. Matrix elements of D (K, L) for $NPA=NPB=5$ $(NP=25)$.

§ 3. Example FORTRAN program for Mössbauer spectroscopy

We shall present an example FORTRAN program, which is useful in the evaluation of overlapped Mössbauer spectra. In this program $Y_{cal}(IP, IC)$'s are calculated after diagonalizing the Hamiltonian for a set of given physical parameters by a program originally coded by Professor Toshizo Fujita of Hiroshima University (1965 in ALGOL for a computer set in Institute for Solid Stote Physics, The University of Tokyo).

In this program we have used graphic subroutines provided for MicroVAX II minicomputer, which have to be adapted if other computers are used. We have used the Gauss-Jordan method to solve the normal equations, which will be better to be replaced by other method, e.g. the LU-decomposition method[3]. In order to show the estimated distribution P on the console screen display, we have tentatively used the contour map representation, for which a subroutine coded by Professor Masatake Mori of Tsukuba University[4] is used. This may be reslaced by a 3-dimensional perspective representation.

§ 4. Acknowledgement

One of the authors (N. H.) has no experience on data analysis, especially in the Mössbauer spectroscopy, and owes much to Professor Atsuko Ito for her suggestions based on her long time experience on the latter field.

References

[1] B. Window: J. Phys. E: Sci. Instrum. **4** (1971) 401-402.
[2] J. Hesse and A. Rübartsch: J. Phys. E: Sci. Instrum. **7** (1974) 526-532.
[3] M. Mori: "Programming in Numerical Calculations" (in Japanese) 2nd ed., Iwanami-shoten, Tokyo 1987, Chap. 5, pp. 52-78.
[4] M. Mori: "Curves and Surfaces" (in Japanese), Kyoiku-shuppan, Tokyo 1974, pp. 136-137.

```
      PROGRAM HESSE3V
*************************************************************************
* This program MOSSP3V must be run on the console of MicroVAX II.         *
*    This program can be used to compare MOeSsbauer SPectral line shapes  *
*      for various values of physical parameters in case of POWDER or     *
*        SINGLE CRYSTAL samples with an experiment.                       *
*    In "Moessbauer Spectrum" window only the central portion from 30th   *
*      to 226th (or from 60th to 452th) channel are shown compared to     *
*        the total channel number 256 (or 512).                           *
*    "Moessbauer Spectrum" window can be pushed or popped by selecting    *
*      "Push behined" or "Pop in front" in the icon Menu of the window.   *
*    Experimental data file must be made in the format: 4 lines of memo   *
*      in "A80" format and 64 (or 128 or 32) lines of 8 data in "8F9.0"   *
*        format. And this file must be placed in the same directory as    *
*          this program is.                                               *
*    Experimental values are read from this file and plotted with small   *
*      circles in "Moessbauer Spectrum" window, and this whole set of     *
*        circles can be shifted or changed in its dip-scale by adjusting  *
*          the point of maximum dip with using MOUSE.                     *
*    Calculated line shape curves are shown with using 6 colors in the    *
*      same window, which can be printed out on Laser Printer.            *
*    Calculated data can be filed into the hard disk, which can be sent   *
*      to NEC PC9801 terminal in MS-DOS format.                           *
* Before running the program, if hard copies are needed,"Workstation Setup"*
*    should be done.                                                      *
*   Select "Printer setup" and give "Aspect ratio" as 1 to 1 and          *
*     "Enter new print destination" in the following format:              *
*            DUB1:[user's default directory name]                         *
*       in which a SIXEL file SCREEN_OUTPUT.SIX is written when           *
*     "Print (portion of) screen" is selected in "Workstation Option"     *
*       window.                                                           *
*     Don't forget to "Save current settings" before "Exit this menu"     *
*       in "Workstation Setup" window.                                    *
*   After the program ends, Laser Printer output can be obtained          *
*     by issuing KPRINT command as follows, if necessary, by specifing    *
*       the version number for each SIXEL file:                           *
*        $ KPRINT/LN80/SIX DUB1:[user's default directory name]           *
*                         SCREEN_OUTPUT.SIX;version number                *
*       ($ is the DCL prompt).                                            *
*************************************************************************
      CHARACTER*1 YN
    1 WRITE(*,'($,'' New Datum?(Y/N) '')')
      READ(*,'(A)') YN
       IF(.NOT.((YN.EQ.'Y').OR.(YN.EQ.'y'))) GO TO 4
          CALL DATFRD
    2     CALL SCRINI
          CALL PLTINI
          CALL DATPLT
    3 WRITE(*,'($,'' New parameters?(Y/N)'')')
       READ(*,'(A)') YN
        IF(.NOT.((YN.EQ.'Y').OR.(YN.EQ.'y'))  )GO TO 1
          CALL PARAMT
          CALL HESSE
          CALL PLOTNG
      WRITE(*,'($,'' Disk-filing?(Y/N)'')')
       READ(*,'(A)') YN
        IF((YN.EQ.'Y').OR.(YN.EQ.'y')) THEN
          CALL DISKFL
        ENDIF
      WRITE(*,'($,'' Continue Analysis?'')')
      READ(*,'(A)') YN
       IF((YN.EQ.'Y').OR.(YN.EQ.'y')) THEN
        WRITE(*,'($,'' Clear screen?'')')
        READ(*,'(A)') YN
```

```
         IF((YN.EQ.'Y').OR.(YN.EQ.'y')) THEN
             CALL SCRFIN
             GO TO 2
         ELSE
             GO TO 3
         ENDIF
       ELSE
           CALL SCRFIN
       ENDIF
    4 CONTINUE
      STOP
      END
**********************************************************************
* Screen plot initiarizing subroutine
**********************************************************************
      SUBROUTINE SCRINI
*
* In the color map Color Map Index runs from 0 to 7, and
*   the corresponding colors are as follows:
*     0:white(used for background) 1:black 2:blue 3:cyan
*     4:green 5:yellow 6:magenta 7:red.
*   Index of ATRiBute Blocks runs from 2 to 9 in correspondence to
*     Color Map Indices 1 to 7 and 0,respectively.
*
      INTEGER VD_ID,WD_ID,VCM_ID
      INTEGER*2 KEYBUF(2)
      COMMON /SCREN/ VD_ID,WD_ID,IATRBB,INDEXW,KEYBUF
        INCLUDE 'SYS$LIBRARY:UISENTRY'
        INCLUDE 'SYS$LIBRARY:UISUSRDEF'
      VCM_ID=UIS$CREATE_COLOR_MAP(8)
      VD_ID=UIS$CREATE_DISPLAY(0.0,0.0,25.6,18.0,37.3,26.7,VCM_ID)
      WD_ID=UIS$CREATE_WINDOW(VD_ID,'SYS$WORKSTATION',
     *        'MOESSBAUER SPECTRUM',3.0,0.0,22.6,18.0,28.6,26.7)
        CALL UIS$SET_COLOR(VD_ID,0,1.0,1.0,1.0)
        CALL UIS$SET_COLOR(VD_ID,1,0.0,0.0,0.0)
        CALL UIS$SET_COLOR(VD_ID,2,0.0,0.0,0.7)
        CALL UIS$SET_COLOR(VD_ID,3,0.0,0.5,0.5)
        CALL UIS$SET_COLOR(VD_ID,4,0.0,0.5,0.0)
        CALL UIS$SET_COLOR(VD_ID,5,0.7,0.7,0.0)
        CALL UIS$SET_COLOR(VD_ID,6,0.7,0.0,0.7)
        CALL UIS$SET_COLOR(VD_ID,7,1.0,0.0,0.0)
        CALL UIS$SET_WRITING_INDEX(VD_ID,0,2,1)
        CALL UIS$SET_WRITING_INDEX(VD_ID,0,3,2)
        CALL UIS$SET_WRITING_INDEX(VD_ID,0,4,3)
        CALL UIS$SET_WRITING_INDEX(VD_ID,0,5,4)
        CALL UIS$SET_WRITING_INDEX(VD_ID,0,6,5)
        CALL UIS$SET_WRITING_INDEX(VD_ID,0,7,6)
        CALL UIS$SET_WRITING_INDEX(VD_ID,0,8,7)
        CALL UIS$SET_WRITING_INDEX(VD_ID,0,9,0)
        CALL UIS$SET_POINTER_AST(VD_ID,WD_ID,CROSS,0)
       CALL UIS$PUSH_VIEWPORT(WD_ID)
      RETURN
      END
**************************************************************************
* Data file reading subroutine
*************************************************************************
      SUBROUTINE DATFRD
*
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER VD_ID,WD_ID
      INTEGER*2 KEYBUF(2)
      COMMON /SCREN/VD_ID,WD_ID,IATRBB,INDEXW,KEYBUF
      COMMON /OBSDT/YOBS(1024),NCHANL,CP
      COMMON /LINES/CV,WD(8),SF(512),YFIT(512),SQKAI
```

```
      CHARACTER OBDFNM*12,MEMO(4)*80,YN*1
      COMMON /OFMEMO/MEMO
      DIMENSION YWORK(1024)
      REAL HALFPT
    1 WRITE(*,'($,'' Observed data file name? (########.@@@)'')')
        READ(*,'(A12)',ERR=1) OBDFNM
    2 WRITE(*,'($,'' Number of data (512 or 1024 or 256) ?'')')
        READ(*,'(I4)',ERR=2) NDATA
          IF(.NOT.((NDATA.EQ.512).OR.(NDATA.EQ.1024).OR.(NDATA.EQ.
     *             256))) GO TO 2
      OPEN(5,FILE=OBDFNM,STATUS='OLD')
        READ(5,'(A80)') (MEMO(I),I=1,4)
      WRITE(*,'(1X,A80)') (MEMO(I),I=1,4)
        READ(5,'(8F9.0)') (YOBS(I),I=1,NDATA)
      WRITE(*,'($,'' Do you want to delete the data file? (Y/N)'')')
        READ(*,'(A)') YN
          IF((YN.EQ.'Y').OR.(YN.EQ.'y')) THEN
            CLOSE(5,STATUS='DELETE')
          ELSE
            CLOSE(5)
          ENDIF
    3 WRITE(*,'($,'' Is the folding of data necessary ? (Y/N)'')')
        READ(*,'(A)',ERR=3) YN
      IF(((YN.EQ.'Y').OR.(YN.EQ.'y')).AND.(NDATA.NE.256)) THEN
    4    WRITE(*,'('' Input the half point (e.g. 256.5 or 512.5),'',
     *         /$,'' where the data set is folded !'')')
            READ(*,'(F10.1)',ERR=4) HALFPT
            IF(HALFPT.EQ.0.0) GO TO 4
            NHALF=INT(HALFPT)
              NCHANL=256
               IF(NDATA.EQ.1024) NCHANL=512
              NSHIFT=NCHANL-NHALF
          DO 30 I=1,NDATA
              J=I+NSHIFT
            IF((J.GE.1).AND.(J.LE.NDATA)) YWORK(J)=YOBS(I)
   30     CONTINUE
              NINI=IABS(NSHIFT)+1
              NDATA1=NDATA+1
          DO 40 I=NINI,NCHANL
              YOBS(I)=(YWORK(I)+YWORK(NDATA1-I))*0.5D+00
   40    CONTINUE
      ELSE
          NINI=1
          NCHANL=256
         IF((NDATA.EQ.512).OR.(NDATA.EQ.1024)) NCHANL=512
      ENDIF
         YOBMIN=1.0D+10
         NINIMD=NINI
          IF(NINI.EQ.1) NINIMD=3
       DO 50 I=3,NCHANL-2
          YOBSI=YOBS(I)
         IF(YOBSI.LT.YOBMIN) THEN
            YOBMIN=YOBSI
            MINCHL=I
         ENDIF
   50  CONTINUE
         WRITE(*,*) ' MINCHL=',MINCHL,' NINI=',NINI
    5  WRITE(*,'('' Give 2 channel intervals (IBGINI,IBGFIN) and'',
     *        /'' (JBGINI,JBGFIN)''/'' from which back ground '',
     *         '' level is determined (IBGINI has to be greater'',
     *         '' than '',I5,'' and JBGFIN smaller than '',I5)')
     *        NINI,NCHANL
         READ(*,'(4I5)',ERR=5) IBGINI,IBGFIN,JBGINI,JBGFIN
        IF((IBGINI.LE.NINI).OR.(JBGFIN.GE.NCHANL)) GO TO 5
```

```
        IF(IBGINI.EQ.1) IBGINI=2
          BACKGD=0.0D+00
        DO 60 I=IBGINI,IBGFIN
          BACKGD=BACKGD+YOBS(I)
   60 CONTINUE
        DO 70 I=JBGINI,JBGFIN
          BACKGD=BACKGD+YOBS(I)
   70 CONTINUE
          BACKGD=BACKGD/DBLE(IBGFIN-IBGINI+JBGFIN-JBGINI+2)
          YOBDIP=BACKGD-YOBMIN
          FACTOR=8.0D+00/YOBDIP
          NINI1=NINI+1
        DO 80 I=1,NINI
         YOBS(I)=0.0D0
   80 CONTINUE
        DO 90 I=NINI1,NCHANL
         YOBS(I)=(BACKGD-YOBS(I))*FACTOR
   90 CONTINUE
          WRITE(*,'(6F12.5)') (YOBS(I),I=1,10)
          YOBDIP=8.0D+00
          INDEXW=2
    6 WRITE(*,'($,'' Position of the center for Fe metal(channel)?'')')
        READ(*,'(F10.5)',ERR=6) CP
        IF(CP.EQ.0.0) GO TO 6
    7 WRITE(*,'($,'' Channel per velocity (channel/mm/sec)?'')')
        READ(*,'(F10.5)',ERR=7) CV
        IF(CV.EQ.0.0) GO TO 7
      RETURN
      END
*******************************************************************************
* Experimental data plotting subroutines
*******************************************************************************
      SUBROUTINE DATPLT
*
* Channel number of the maximum dip point is shown with a short
*   vertical line segment.
*
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /OBSDT/YOBS(1024),NCHANL,CP
      INTEGER VD_ID,WD_ID
      INTEGER*2 KEYBUF(2)
      COMMON /SCREN/VD_ID,WD_ID,IATRBB,INDEXW,KEYBUF
      REAL*4 XC,YC,XL,XR,BG,DX
        INCLUDE 'SYS$LIBRARY:UISENTRY'
        INCLUDE 'SYS$LIBRARY:UISUSRDEF'
         BG=17.0
         XL=0.0
         XR=25.6
         DX=0.1
       IF(NCHANL.EQ.512) DX=0.05
      DO 10 I=1,NCHANL
         XC=REAL(I)*DX+XL
          IF(XC.GT.XR) XC=XR
         YC=BG-SNGL(YOBS(I))
        CALL UIS$CIRCLE(VD_ID,INDEXW,XC,YC,0.05)
   10 CONTINUE
      RETURN
      END
******************************************************************************
* Parameters input subroutine
******************************************************************************
      SUBROUTINE PARAMT
*
      IMPLICIT REAL*8(A-H,O-Z)
```

```
      COMMON /PARAM1/CENTER,GG,QQ,E,XDEG,YDEG,XXG,YYG
      COMMON /PARAM2/GAMMA,IPS
      COMMON /PARAM3/AMIN,AMAX,BMIN,BMAX,NPA,NPB,NP,KSWTCH,LSWTCH(2)
      COMMON /REGION/IFI,IFF,IWI,IWF,WW
      COMMON /OBSDT/YOBS(1024),NCHANL,CP
      CHARACTER YN*1
      COMMON /LINES/CV,WD(8),SF(512),YFIT(512),SQKAI
      COMMON /PROB/P(226),AAV
          ISWTCH=0
      WRITE(*,'($,'' Do you want to input or change all parameters?'',
     *            '' (Y/N). First time reply "Y"!'')')
        READ(*,'(A)') YN
         IF((YN.EQ.'Y').OR.(YN.EQ.'y')) THEN
          ISWTCH=1
          GO TO 200
         ENDIF
100   WRITE(*,'(''****** MENU ************************************** '',
     *         /''                                                    '',
     *         /''    1. Hyperfine field                              '',
     *         /''    2. Quadrupole splitting                         '',
     *         /''    3. Asymmetry                                    '',
     *         /''    4. Position of the center of sample spectrum   '',
     *         /''    5. Hyperfine field direction                    '',
     *         /''    6. Gamma ray direction                          '',
     *         /''    7. Half width at half maximum(mm/sec) of lines'',
     *         /''    8. Penalty for jaggy distribution               '',
     *         /''    9. Fitting region                               '',
     *         /''   10. Important region                             '',
     *         /''   11. Exit this menu                               '',
     *        /$,''  Input an item number: '')')
        READ(*,'(I2)') ITEM
          GO TO (1,2,3,4,5,6,7,8,9,10,20) ITEM
            GO TO 100
200   WRITE(*,'($,'' Single or double parameter ? '',
     *           /'' Input (1 or 2)'')')
      READ(*,'(I1)',ERR=200) KSWTCH
        IF((KSWTCH.NE.1).AND.(KSWTCH.NE.2))GO TO 200
300     WRITE(*,'($,''******* Distribution *******      '',
     *           /''                                  '',
     *           /''    1. Hyperfine field            '',
     *           /''    2. Quadrupole splitting       '',
     *           /''    3. Asymmetry                  '',
     *           /''    4. Center of sample spectrum  '',
     *           /''    5. Theta of Hyperfine fild    '',
     *           /''    6. Phi of Hyperfine field     '',
     *           /''    7. Theta of Gamma ray         '',
     *           /''    8. Phi of Gamma ray           '',
     *           /''                                  '')')
      IF(KSWTCH.EQ.1)THEN
        WRITE(*,'($,''   Input an item number: '')')
        READ(*,'(I1)',ERR=300) LSWTCH(1)
        IF(LSWTCH(1).GT.8)GO TO 300
        LSWTCH(2)=0
       ENDIF
      IF(KSWTCH.EQ.2)THEN
        WRITE(*,'('' Input two item numbers. '')')
        WRITE(*,'($,'' First parameter: '')')
        READ(*,'(I1)',ERR=300) LSWTCH(1)
        WRITE(*,'($,'' Second parameter: '')')
        READ(*,'(I1)',ERR=300) LSWTCH(2)
        IF((LSWTCH(1).GT.8).OR.(LSWTCH(2).GT.8))GO TO 300
        IF(LSWTCH(1).EQ.LSWTCH(2))GO TO 300
      ENDIF
        WRITE(*,*) LSWTCH(1)
```

```
      WRITE(*,*) LSWTCH(2)
1 IF(LSWTCH(1).EQ.1)THEN
     WRITE(*,'($,'' Minimum and maximum'',
*              '' of Hyperfine field(kOe)? '')')
      READ(*,'(2F10.5)',ERR=1) AMIN,AMAX
      WRITE(*,'($,'' Number of field values'',
*              '' in this range ? '')')
      READ(*,'(I3)',ERR=1) NPA
      IF(NPA.LT.4) GO TO 1
    ELSE
      IF(LSWTCH(2).EQ.1)THEN
         WRITE(*,'($,'' Minimum and maximum'',
*                '' of Hyperfine field(kOe) ? '')')
         READ(*,'(2F10.5)',ERR=1) BMIN,BMAX
         WRITE(*,'($,'' Number of field values'',
*                '' in this range ? '')')
         READ(*,'(I3)',ERR=1) NPB
         IF(NPB.LT.4) GO TO 1
      ELSE
         WRITE(*,'($,'' Hyperfine field(kOe) ? '')')
         READ(*,'(F10.5)',ERR=1) GG
      ENDIF
    ENDIF
      IF(ISWTCH.NE.1) GO TO 100
2 IF(LSWTCH(1).EQ.2)THEN
     WRITE(*,'($,'' Minimum and maximum'',
*             '' of Quadrupole splitting(mm/sec)? '')')
      READ(*,'(2F10.5)',ERR=2) AMIN,AMAX
      WRITE(*,'($,'' Number of Quadrupole spliting values'',
*              '' in this range ? '')')
      READ(*,'(I3)',ERR=2) NPA
      IF(NPA.LT.4) GO TO 2
    ELSE
      IF(LSWTCH(2).EQ.2)THEN
     WRITE(*,'($,'' Minimum and maximum'',
*             '' of Quadrupole splitting(mm/sec)?'')')
         READ(*,'(2F10.5)',ERR=2) BMIN,BMAX
         WRITE(*,'($,'' Number of Quadrupole splitting values'',
*                '' in this range ? '')')
         READ(*,'(I3)',ERR=2) NPB
         IF(NPB.LT.4) GO TO 2
      ELSE
         WRITE(*,'($,'' Quadrupole splitting(mm/sec) ? '')')
         READ(*,'(F10.5)',ERR=2) QQ
      ENDIF
    ENDIF
      IF(ISWTCH.NE.1) GO TO 100
3 IF(LSWTCH(1).EQ.3)THEN
     WRITE(*,'($,'' Minimum and maximum of Asymmetry?'')')
      READ(*,'(2F10.5)',ERR=3) AMIN,AMAX
      WRITE(*,'($,'' Number of Asymmetry values'',
*              '' in this range ? '')')
      READ(*,'(I3)',ERR=3) NPA
      IF(NPA.LT.4) GO TO 3
    ELSE
      IF(LSWTCH(2).EQ.3)THEN
     WRITE(*,'($,'' Minimum and maximum of Asymmetry?'')')
         READ(*,'(2F10.5)',ERR=3) BMIN,BMAX
         WRITE(*,'($,'' Number of Asymmetry values'',
*                '' in this range ? '')')
         READ(*,'(I3)',ERR=3) NPB
         IF(NPB.LT.4) GO TO 3
      ELSE
         WRITE(*,'($,'' Asymmetry ? '')')
```

```
          READ(*,'(F10.5)',ERR=3) E
        ENDIF
      ENDIF
        IF(ISWTCH.NE.1) GO TO 100
4   IF(LSWTCH(1).EQ.4)THEN
       WRITE(*,'($,'' Minimum and maximum'',
 *               ''of the center of sample(channel)?'')')
        READ(*,'(2F10.5)',ERR=4) AMIN,AMAX
        WRITE(*,'($,'' Number of center values'',
 *                '' in this range ? '')')
        READ(*,'(I3)',ERR=4) NPA
        IF(NPA.LT.4) GO TO 4
      ELSE
        IF(LSWTCH(2).EQ.4)THEN
       WRITE(*,'($,'' Minimum and maximum'',
 *               '' of the center of sample(channel)?'')')
          READ(*,'(2F10.5)',ERR=4) BMIN,BMAX
          WRITE(*,'($,'' Number of center values'',
 *                   '' in this range ? '')')
          READ(*,'(I3)',ERR=4) NPB
          IF(NPB.LT.4) GO TO 4
        ELSE
          WRITE(*,'($,'' Center of sample(channel) ? '')')
          READ(*,'(F10.5)',ERR=4) CENTER
        ENDIF
      ENDIF
        IF(ISWTCH.NE.1) GO TO 100
5   IF(LSWTCH(1).EQ.5)THEN
       WRITE(*,'($,'' Minimum and maximum'',
 *               '' of theta(XDEG) of Hyperfine field(deg) ? '')')
        READ(*,'(2F10.5)',ERR=5) AMIN,AMAX
        WRITE(*,'($,'' Number of angle values'',
 *                '' in this range ? '')')
        READ(*,'(I3)',ERR=5) NPA
        IF(NPA.LT.4) GO TO 5
      ELSE
        IF(LSWTCH(2).EQ.5)THEN
       WRITE(*,'($,'' Minimum and maximum'',
 *               '' of theta(XDEG) of Hyperfine field(deg) ? '')')
          READ(*,'(2F10.5)',ERR=5) BMIN,BMAX
          WRITE(*,'($,'' Number of angle?'')')
          READ(*,'(I3)',ERR=5) NPB
          IF(NPB.LT.4) GO TO 5
        ELSE
          WRITE(*,'($,'' Theta(XDEG) of Hyperfine field(deg)?'')')
          READ(*,'(F10.5)',ERR=5) XDEG
        ENDIF
      ENDIF
        IF(ISWTCH.NE.1) GO TO 100
    IF(LSWTCH(1).EQ.6)THEN
       WRITE(*,'($,'' Minimum and maximum'',
 *         '' of phi(YDEG) of Hyperfine field(deg)?'')')
        READ(*,'(2F10.5)',ERR=5) AMIN,AMAX
       WRITE(*,'($,'' Number of angle values in this range ? '')')
        READ(*,'(I3)',ERR=5) NPA
        IF(NPA.LT.4) GO TO 5
      ELSE
        IF(LSWTCH(2).EQ.6)THEN
       WRITE(*,'($,'' Minimum and maximum'',
 *          '' of phi(YDEG) of Hyperfine field(deg)?'')')
          READ(*,'(2F10.5)',ERR=5) BMIN,BMAX
       WRITE(*,'($,'' Number of angle values in this range ? '')')
          READ(*,'(I3)',ERR=5) NPB
          IF(NPB.LT.4) GO TO 5
```

```
      ELSE
         WRITE(*,'($,'' Phi(YDEG)'',
     *                  ''of Hyperfine field(deg) ? '')')
         READ(*,'(F10.5)',ERR=5) YDEG
      ENDIF
    ENDIF
      IF(ISWTCH.NE.1) GO TO 100
6 IF(LSWTCH(1).EQ.7)THEN
     WRITE(*,'($,'' Minimum and maximum'',
     *             '' of theta(XXG) of Gamma ray(deg) ? '')')
      READ(*,'(2F10.5)',ERR=6) AMIN,AMAX
      WRITE(*,'($,'' Number of angle values'',
     *              '' in this range ? '')')
      READ(*,'(I3)',ERR=6) NPA
      IF(NPA.LT.4) GO TO 6
    ELSE
      IF((LSWTCH(2).EQ.7))THEN
     WRITE(*,'($,'' Minimum and maximum'',
     *             '' of theta(XXG) of Gamma ray(deg) ? '')')
         READ(*,'(2F10.5)',ERR=6) BMIN,BMAX
         WRITE(*,'($,'' Number of angle values'',
     *                  '' in this range ? '')')
         READ(*,'(I3)',ERR=6) NPB
         IF(NPB.LT.4) GO TO 6
      ELSE
         WRITE(*,'($,'' Theta(XXG) of'',
     *                  '' Gamma ray direction(deg) ? '')')
         READ(*,'(F10.5)',ERR=6) XXG
      ENDIF
    ENDIF
      IF(ISWTCH.NE.1) GO TO 100
  IF((LSWTCH(1).EQ.8))THEN
     WRITE(*,'($,'' Minimum and maximum'',
     *             '' of phi(YYG) of Gamma ray(deg) ? '')')
      READ(*,'(2F10.5)',ERR=6) AMIN,AMAX
      WRITE(*,'($,'' Number of angle values'',
     *              '' in this range ? '')')
      READ(*,'(I3)',ERR=6) NPA
      IF(NPA.LT.4) GO TO 6
    ELSE
      IF((LSWTCH(2).EQ.8))THEN
     WRITE(*,'($,'' Minimum and maximum'',
     *        '' of phi(YYG) of Hyperfine field(deg) ? '')')
         READ(*,'(2F10.5)',ERR=6) BMIN,BMAX
         WRITE(*,'($,'' Number of angle values'',
     *                  '' in this range ? '')')
         READ(*,'(I3)',ERR=6) NPB
         IF(NPB.LT.4) GO TO 6
      ELSE
         WRITE(*,'($,'' Phi(YYG)'',
     *                  '' of Gamma ray direction(deg) ? '')')
         READ(*,'(F10.5)',ERR=6) YYG
      ENDIF
    ENDIF
      IF(ISWTCH.NE.1) GO TO 100
7 WRITE(*,'('' 8 half width at half maximum(mm/sec) of lines?'')')
   READ(*,'(8F10.5)',ERR=7) (WD(I),I=1,8)
    WWD=WD(1)*WD(2)*WD(3)*WD(4)*WD(5)*WD(6)*WD(7)*WD(8)
    IF(WWD.EQ.0.0) GO TO 7
    IF(ISWTCH.NE.1) GO TO 100
8 WRITE(*,'($,'' Penalty for jaggy distribution ? '')')
   READ(*,'(F10.5)',ERR=8) GAMMA
    IF(ISWTCH.NE.1) GO TO 100
9 WRITE(*,'($,'' Fitting region ? [IFI,IFF]'')')
```

```
      READ(*,'(2I5)',ERR=9) IFI,IFF
        IF((IFI.LE.2).OR.(IFI.GE.NCHANL)) GO TO 9
        IF((IFF.LE.2).OR.(IFF.GE.NCHANL)) GO TO 9
        IF(ISWTCH.NE.1) GO TO 100
   10 WRITE(*,'($,'' Important region (IWI,IWF) and weight WW ? '')')
      READ(*,'(2I5,F10.5)',ERR=10)IWI,IWF,WW
        IF(WW.EQ.0.0) GO TO 10
        IF(ISWTCH.NE.1) GO TO 100
   20 CONTINUE
         NP=(NPA+1)*(NPB+1)
         IF(NP.GT.225) GO TO 200
      RETURN
      END
*****************************************************************************
* Parameters plot subroutine
*****************************************************************************
      SUBROUTINE PLTTEXT
*
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /PARAM1/CENTER,GG,QQ,E,XDEG,YDEG,XXG,YYG
      COMMON /PARAM2/GAMMA,IPS
      COMMON /PARAM3/AMIN,AMAX,BMIN,BMAX,NPA,NPB,NP,KSWTCH,LSWTCH(2)
      COMMON /REGION/IFI,IFF,IWI,IWF,WW
      CHARACTER YN*1,STRING*23,C(9)*6,CC(8)*5
      COMMON /LINES/CV,WD(8),SF(512),YFIT(512),SQKAI
      COMMON /PROB/P(226),AAV
      COMMON /SCREN/VD_ID,WD_ID,IATRBB,INDEXW,KEYBUF
*-----------------------------------------------------------------------------
        STRING=' Hyperfine field  :'
          CALL UIS$TEXT(VD_ID,2,STRING,15.0,7.0)
        STRING=' Quadrupole sp.   :'
          CALL UIS$TEXT(VD_ID,2,STRING,15.0,6.5)
        STRING=' Asymmetry        :'
          CALL UIS$TEXT(VD_ID,2,STRING,15.0,6.0)
        STRING=' Center of sample :'
          CALL UIS$TEXT(VD_ID,2,STRING,15.0,5.5)
        STRING=' Hhf        theta :'
          CALL UIS$TEXT(VD_ID,2,STRING,15.0,5.0)
        STRING='            phi   :'
          CALL UIS$TEXT(VD_ID,2,STRING,15.0,4.5)
        STRING=' Gamma ray  theta :'
          CALL UIS$TEXT(VD_ID,2,STRING,15.0,4.0)
        STRING='            phi   :'
          CALL UIS$TEXT(VD_ID,2,STRING,15.0,3.5)
        STRING=' Half width     :'
          CALL UIS$TEXT(VD_ID,2,STRING,15.0,3.0)
        STRING=' Penalty          :'
          CALL UIS$TEXT(VD_ID,2,STRING,15.0,2.0)
        STRING=' Fit. region & wt :'
          CALL UIS$TEXT(VD_ID,2,STRING,15.0,1.5)
        STRING=' Important region :'
          CALL UIS$TEXT(VD_ID,2,STRING,15.0,1.0)
        STRING=' Sq. Kai          :'
          CALL UIS$TEXT(VD_ID,2,STRING,15.0,0.5)
*-----------------------------------------------------------
          ENCODE(8,100,C(1)) AMIN
          ENCODE(8,100,C(2)) AMAX
          ENCODE(8,200,C(3)) NPA
          ENCODE(8,100,C(4)) BMIN
          ENCODE(8,100,C(5)) BMAX
          ENCODE(8,200,C(6)) NPB
  100 FORMAT(F6.2)
  200 FORMAT(I3)
*1---------------------------------------------
```

```
      IF(LSWTCH(1).EQ.1)THEN
         STRING='('//C(1)//','//C(2)//')'//C(3)
         CALL UIS$TEXT(VD_ID,2,STRING,18.5,7.0)
       ELSE
         IF(LSWTCH(2).EQ.1)THEN
            STRING='('//C(4)//','//C(5)//')'//C(6)
            CALL UIS$TEXT(VD_ID,2,STRING,18.5,7.0)
         ELSE
            ENCODE(8,100,C(7)) GG
            CALL UIS$TEXT(VD_ID,2,C(7),18.5,7.0)
         ENDIF
       ENDIF
*2--------------------------------------------------
      IF(LSWTCH(1).EQ.2)THEN
         STRING='('//C(1)//','//C(2)//')'//C(3)
         CALL UIS$TEXT(VD_ID,2,STRING,18.5,6.5)
       ELSE
         IF(LSWTCH(2).EQ.2)THEN
            STRING='('//C(4)//','//C(5)//')'//C(6)
            CALL UIS$TEXT(VD_ID,2,STRING,18.5,6.5)
         ELSE
            ENCODE(8,100,C(7)) QQ
            CALL UIS$TEXT(VD_ID,2,C(7),18.5,6.5)
         ENDIF
       ENDIF
*3--------------------------------------------------
      IF(LSWTCH(1).EQ.3)THEN
         STRING='('//C(1)//','//C(2)//')'//C(3)
         CALL UIS$TEXT(VD_ID,2,STRING,18.5,6.0)
       ELSE
         IF(LSWTCH(2).EQ.3)THEN
            STRING='('//C(4)//','//C(5)//')'//C(6)
            CALL UIS$TEXT(VD_ID,2,STRING,18.5,6.0)
         ELSE
            ENCODE(8,100,C(7)) E
            CALL UIS$TEXT(VD_ID,2,C(7),18.5,6.0)
         ENDIF
       ENDIF
*4--------------------------------------------------
      IF(LSWTCH(1).EQ.4)THEN
         STRING='('//C(1)//','//C(2)//')'//C(3)
         CALL UIS$TEXT(VD_ID,2,STRING,18.5,5.5)
       ELSE
         IF(LSWTCH(2).EQ.4)THEN
            STRING='('//C(4)//','//C(5)//')'//C(6)
            CALL UIS$TEXT(VD_ID,2,STRING,18.5,5.5)
         ELSE
            ENCODE(8,100,C(7)) CENTER
            CALL UIS$TEXT(VD_ID,2,C(7),18.5,5.5)
         ENDIF
       ENDIF
*5--------------------------------------------------
      IF(LSWTCH(1).EQ.5)THEN
         STRING='('//C(1)//','//C(2)//')'//C(3)
         CALL UIS$TEXT(VD_ID,2,STRING,18.5,5.0)
       ELSE
         IF(LSWTCH(2).EQ.5)THEN
            STRING='('//C(4)//','//C(5)//')'//C(6)
            CALL UIS$TEXT(VD_ID,2,STRING,18.5,5.0)
         ELSE
            ENCODE(8,100,C(7)) XDEG
            CALL UIS$TEXT(VD_ID,2,C(7),18.5,5.0)
         ENDIF
       ENDIF
```

```
*6-----------------------------------------------
      IF(LSWTCH(1).EQ.6)THEN
        STRING='('//C(1)//','//C(2)//')'//C(3)
        CALL UIS$TEXT(VD_ID,2,STRING,18.5,4.5)
      ELSE
        IF(LSWTCH(2).EQ.6)THEN
          STRING='('//C(4)//','//C(5)//')'//C(6)
          CALL UIS$TEXT(VD_ID,2,STRING,18.5,4.5)
        ELSE
          ENCODE(8,100,C(7)) YDEG
          CALL UIS$TEXT(VD_ID,2,C(7),18.5,4.5)
        ENDIF
      ENDIF
*7-----------------------------------------------
      IF(LSWTCH(1).EQ.7)THEN
        STRING='('//C(1)//','//C(2)//')'//C(3)
        CALL UIS$TEXT(VD_ID,2,STRING,18.5,4.0)
      ELSE
        IF(LSWTCH(2).EQ.7)THEN
          STRING='('//C(4)//','//C(5)//')'//C(6)
          CALL UIS$TEXT(VD_ID,2,STRING,18.5,4.0)
        ELSE
          ENCODE(8,100,C(7)) XXG
          CALL UIS$TEXT(VD_ID,2,C(7),18.5,4.0)
        ENDIF
      ENDIF
*8-----------------------------------------------
      IF(LSWTCH(1).EQ.8)THEN
        STRING='('//C(1)//','//C(2)//')'//C(3)
        CALL UIS$TEXT(VD_ID,2,STRING,18.5,3.5)
      ELSE
        IF(LSWTCH(2).EQ.8)THEN
          STRING='('//C(4)//','//C(5)//')'//C(6)
          CALL UIS$TEXT(VD_ID,2,STRING,18.5,3.5)
        ELSE
          ENCODE(8,100,C(7)) YYG
          CALL UIS$TEXT(VD_ID,2,C(7),18.5,3.5)
        ENDIF
      ENDIF
*------------------------------------------------
      DO 10 I=1,8
        ENCODE(8,300,CC(I)) WD(I)
   10 CONTINUE
  300 FORMAT(F5.2)
        STRING=CC(1)//','//CC(2)//','//CC(3)//','//CC(4)
      CALL UIS$TEXT(VD_ID,2,STRING,18.0,3.0)
        STRING=CC(5)//','//CC(6)//','//CC(7)//','//CC(8)
      CALL UIS$TEXT(VD_ID,2,STRING,18.0,2.5)
        ENCODE(8,100,C(7)) GAMMA
      CALL UIS$TEXT(VD_ID,2,C(7),18.5,2.0)
        ENCODE(8,200,C(7)) IFI
        ENCODE(8,200,C(8)) IFF
        STRING='('//C(7)//','//C(8)//')'
      CALL UIS$TEXT(VD_ID,2,STRING,18.5,1.5)
        ENCODE(8,200,C(7)) IWI
        ENCODE(8,200,C(8)) IWF
        ENCODE(8,100,C(9)) WW
        STRING='('//C(7)//','//C(8)//')'//C(9)
      CALL UIS$TEXT(VD_ID,2,STRING,18.5,1.0)
        ENCODE(8,100,C(7)) SQKAI
      CALL UIS$TEXT(VD_ID,2,C(7),18.5,0.5)
      RETURN
      END
*****************************************************************************
```

```
* Hesse Method subroutine
***********************************************************************
      SUBROUTINE HESSE
*
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /PARAM1/CENTER,GG,QQ,E,XDEG,YDEG,XXG,YYG
      COMMON /PARAM2/GAMMA,IPS
      COMMON /PARAM3/AMIN,AMAX,BMIN,BMAX,NPA,NPB,NP,KSWTCH,LSWTCH(2)
      COMMON /REGION/IFI,IFF,IWI,IWF,WW
      CHARACTER YN*1
      COMMON /LINES/CV,WD(8),SF(512),YFIT(512),SQKAI
      COMMON /PROB/P(226),AAV
      DIMENSION D(226,226),A(226,226),B(226),FHX(226,512),WEIGHT(512)
      DIMENSION YERROR(512)
      COMMON /OBSDT/YOBS(1024),NCHANL,CP
*------------------------------------------------------------------------
*     IMPORTANT REGION AND ITS WEIGHT WW
*------------------------------------------------------------------------
    1 WRITE(*,'($,'' Powder or Single?''
     *       /$,'' Input 0 or 1.'')')
       READ(*,'(I1)',ERR=1) IPS
       WRITE(*,'($,'' --- Now computing! --- '')')
       IF((IPS.NE.0).AND.(IPS.NE.1))GO TO 1
      WSUM=0.0D0
      IW1=IWI-1
      DO 10 I=IFI,IW1
        WEIGHT(I)=1.0D0
        WSUM=WSUM+1.0D0
   10 CONTINUE
      DO 20 I=IWI,IWF
        WEIGHT(I)=WW
        WSUM=WSUM+WW
   20 CONTINUE
      IW1=IWF+1
      DO 30 I=IW1,IFF
        WEIGHT(I)=1.0D0
        WSUM=WSUM+1.0D0
   30 CONTINUE
      FW=(IFF-IFI+1)/WSUM
      DO 40 I=IFI,IFF
        WEIGHT(I)=WEIGHT(I)*FW
   40 CONTINUE
*------------------------------------------------------------------------
*     FORMATION OF NORMAL EQUATIONS
*------------------------------------------------------------------------
      IF (KSWTCH.EQ.1)THEN
          CALL DMATRIX1(NPA,D)
          NP=NPA+1
           DELTAA=(AMAX-AMIN)/DBLE(NPA)
          DO 51 IP=1,NPA+1
            AA=DBLE(IP-1)*DELTAA+AMIN
            IF(LSWTCH(1).EQ.1) GG=AA
            IF(LSWTCH(1).EQ.2) QQ=AA
            IF(LSWTCH(1).EQ.3) E=AA
            IF(LSWTCH(1).EQ.4) CENTER=AA
            IF(LSWTCH(1).EQ.5) XDEG=AA
            IF(LSWTCH(1).EQ.6) YDEG=AA
            IF(LSWTCH(1).EQ.7) XXG=AA
            IF(LSWTCH(1).EQ.8) YYG=AA
                CALL MATRIX
              IF(IPS.EQ.0) CALL POWDER
              IF(IPS.EQ.1) CALL SINGLE
               CALL POSITN
               CALL SPECTR
```

```
            DO 50 IX=IFI,IFF
              FHX(IP,IX)=SF(IX)
50        CONTINUE
51       CONTINUE
        ENDIF
*-----------------------------------------------------------------
      IF (KSWTCH.EQ.2)THEN
          CALL DMATRIX2(NPA+1,NPB+1,D)
           NP=(NPA+1)*(NPB+1)
           DELTAA=(AMAX-AMIN)/DBLE(NPA)
           DELTAB=(BMAX-BMIN)/DBLE(NPB)
        DO 56 IA=1,NPA+1
          AA=DBLE(IA-1)*DELTAA+AMIN
          IF(LSWTCH(1).EQ.1) GG=AA
          IF(LSWTCH(1).EQ.2) QQ=AA
          IF(LSWTCH(1).EQ.3) E=AA
          IF(LSWTCH(1).EQ.4) CENTER=AA
          IF(LSWTCH(1).EQ.5) XDEG=AA
          IF(LSWTCH(1).EQ.6) YDEG=AA
          IF(LSWTCH(1).EQ.7) XXG=AA
          IF(LSWTCH(1).EQ.8) YYG=AA
         DO 55 IB=1,NPB+1
           BB=DBLE(IB-1)*DELTAB+BMIN
           IF(LSWTCH(2).EQ.1) GG=BB
           IF(LSWTCH(2).EQ.2) QQ=BB
           IF(LSWTCH(2).EQ.3) E=BB
           IF(LSWTCH(2).EQ.4) CENTER=BB
           IF(LSWTCH(2).EQ.5) XDEG=BB
           IF(LSWTCH(2).EQ.6) YDEG=BB
           IF(LSWTCH(2).EQ.7) XXG=BB
           IF(LSWTCH(2).EQ.8) YYG=BB
            CALL MATRIX
           IF(IPS.EQ.0) CALL POWDER
           IF(IPS.EQ.1) CALL SINGLE
            CALL POSITN
            CALL SPECTR
            IP=(NPB+1)*(IA-1)+IB
            DO 54 IX=IFI,IFF
            FHX(IP,IX)=SF(IX)
54        CONTINUE
55       CONTINUE
56      CONTINUE
      ENDIF
*-----------------------------------------------------------------
      DO 60 IX=IFI,IFF
       FHX(NP+1,IX)=1.0D0
60    CONTINUE
      DO 72 IP=1,NP+1
        DO 71 JP=IP,NP+1
          AIJ=0.0D0
           DO 70 IX=IFI,IFF
            AIJ=AIJ+FHX(IP,IX)*FHX(JP,IX)*WEIGHT(IX)
70         CONTINUE
          AIJ=AIJ+GAMMA*D(IP,JP)
          A(IP,JP)=AIJ
          A(JP,IP)=AIJ
71      CONTINUE
72    CONTINUE
      DO 81 JP=1,NP+1
        BJ=0.0D0
         DO 80 IX=IFI,IFF
           BJ=BJ+FHX(JP,IX)*WEIGHT(IX)*YOBS(IX)
80       CONTINUE
        P(JP)=BJ
```

```
   81 CONTINUE
*-------------------------------------------------------------------------------
*     SOLVING NORMAL EQUATIONS
*-------------------------------------------------------------------------------
      EPS=1.0D-14
      CALL GAUSS(A,NP+1,P,EPS)
      SQKAI=0.0D0
      DO 100 IX=1,512
         YFIT(IX)=0.0D0
  100 CONTINUE
      DO 111 IX=IFI,IFF
        YF=0.0D0
          DO 110 IP=1,NP
            YF=YF+FHX(IP,IX)*P(IP)
  110     CONTINUE
        YFIT(IX)=YF+P(NP+1)
        YE=YFIT(IX)-YOBS(IX)
        YERROR(IX)=YE
        SQKAI=SQKAI+YE*YE*WEIGHT(IX)
  111 CONTINUE
      SQKAI=SQKAI/(IFF-IFI+1-NP)
      CBG=P(NP+1)
*------------------------------------------------------------
      PSUM=0.0D0
      DO 82 IP=1,NP
         PSUM=PSUM+P(IP)
   82 CONTINUE
      DO 83 IP=1,NP
         P(IP)=P(IP)/PSUM
   83 CONTINUE
       WRITE(*,'(6F12.5)') (P(IP),IP=1,NP+1)
      IF (KSWTCH.EQ.1) THEN
          AAV=0.0D0
          DO 90 IA=1,NP
             H=DBLE(IA-1)*DELTAA+AMIN
             AAV=AAV+H*P(IA)
   90     CONTINUE
        CALL PRBPLT1
       ELSE
        CALL PRBPLT2
       ENDIF
        CALL PLTTEXT
      RETURN
      END
****************************************************************************
*    FORMATION D MATRIX
****************************************************************************
      SUBROUTINE DMATRIX1(N,D)
*
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION D(226,N+2)
*
      DO 11 I=1,N+1
         DO 10 J=I,N+1
            DIJ=0.0D0
             K=IABS(I-J)
              IF(K.EQ.0) DIJ=6.0D0
              IF(K.EQ.1) DIJ=-4.0D0
              IF(K.EQ.2) DIJ=1.0D0
            D(I,J)=DIJ
            D(J,I)=DIJ
   10    CONTINUE
          D(I,N+2)=0.0D0
          D(N+2,I)=0.0D0
```

```
   11 CONTINUE
         D(N+2,N+2)=0.0D0
         D(1,1)=1.0D0
         D(1,2)=-2.0D0
         D(2,1)=-2.0D0
         D(2,2)=5.0D0
           D(N+1,N+1)=1.0D0
           D(N+1,N)=-2.0D0
           D(N,N+1)=-2.0D0
           D(N,N)=5.0D0
        RETURN
        END
*****************************************************************************
      SUBROUTINE DMATRIX2(NPA,NPB,D)
*
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION D(226,NPA*NPB+1)
      NP=NPA*NPB
      DO 11 K=1,NP
         DO 10 L=K,NP
            D(K,L)=0.0D+00
   10    CONTINUE
   11 CONTINUE
      DO 21 K=1,NP
           I=MOD(K,NPB)
            IF(I.EQ.0) I=NPB
            KNPB=K+NPB
         DO 20 L=K,NP
           IF(L.EQ.K)THEN
               D(K,K)=20.0D+00
                  IF ((I.EQ.1).OR.(I.EQ.NPB))D(K,K)=12.0D+00
               ENDIF
           IF(L.EQ.K+1)THEN
               D(K,K+1)=-8.0D+00
                  IF((I.EQ.1).OR.(I.EQ.NPB-1)) D(K,K+1)=-7.0D+00
                  IF(I.EQ.NPB) D(K,K+1)=0.0D+00
               ENDIF
           IF(L.EQ.K+2)THEN
               D(K,K+2)=1.0D+00
                  IF((I.EQ.NPB).OR.(I.EQ.NPB-1)) D(K,K+2)=0.0D+00
               ENDIF
           IF(L.EQ.K+NPB)THEN
               D(K,K+NPB)=-8.0D+00
                  IF((I.EQ.1).OR.(I.EQ.NPB)) D(K,K+NPB)=-6.0D+00
               ENDIF
           IF(L.EQ.KNPB+1)THEN
               D(K,KNPB+1)=2.0D+00
                  IF(I.EQ.NPB) D(K,KNPB+1)=0.0D+00
               ENDIF
           IF(L.EQ.KNPB-1)THEN
               D(K,KNPB-1)=2.0D+00
                  IF(I.EQ.1) D(K,KNPB-1)=0.0D+00
               ENDIF
           IF(L.EQ.K+NPB*2)THEN
               D(K,K+NPB*2)=1.0D+00
              ENDIF
      IF ((K.LE.NPB).OR.(K.GT.NP-NPB))THEN
            IF(L.EQ.K)THEN
                D(K,K)=12.0D+00
                IF((I.EQ.1).OR.(I.EQ.NPB)) D(K,K)=6.0D+00
               ENDIF
            IF(L.EQ.K+1)THEN
                D(K,K+1)=-6.0D+00
                IF((I.EQ.1).OR.(I.EQ.NPB-1)) D(K,K+1)=-5.0D+00
```

```
                IF(I.EQ.NPB) D(K,K+1)=0.0D+00
               ENDIF
            IF(L.EQ.K+NPB)THEN
                D(K,K+NPB)=-7.0D+00
                IF((I.EQ.1).OR.(I.EQ.NPB)) D(K,K+NPB)=-5.0D+00
               ENDIF
         ENDIF
       IF((K.GT.NP-NPB*2).AND.(K.LE.NP-NPB)) THEN
         D(K,K+NPB)=-7.0D+00
         IF((I.EQ.1).OR.(I.EQ.NPB)) D(K,K+NPB)=-5.0D+00
       ENDIF
   20 CONTINUE
   21 CONTINUE
      DO 31 K=1,NP
       DO 30 L=K+1,NP
          D(L,K)=D(K,L)
   30  CONTINUE
   31 CONTINUE
      NP1=NP+1
         DO 40 K=1,NP
          D(K,NP1)=0.0D+00
          D(NP1,K)=0.0D+00
   40 CONTINUE
      RETURN
      END
******************************************************************************
*     Disk-filing of calculated data
******************************************************************************
      SUBROUTINE DISKFL
*
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /PARAM1/CENTER,GG,QQ,E,XDEG,YDEG,XXG,YYG
      COMMON /PARAM2/GAMMA,IPS
      COMMON /PARAM3/AMIN,AMAX,BMIN,BMAX,NPA,NPB,NP,KSWTCH,LSWTCH(2)
      COMMON /REGION/IFI,IFF,IWI,IWF,WW
      CHARACTER YN*1,FNAME*12,MEMO(4)*80
      COMMON /FLNAM/FNAME
      COMMON /LINES/CV,WD(8),SF(512),YFIT(512),SQKAI
      COMMON /PROB/P(226),AAV
      COMMON /OBSDT/YOBS(1024),NCHANL,CP
      COMMON /OFMEMO/MEMO
*-----------------------------------------------------------------------
    1 WRITE(*,'($,'' Calculated data file name? (########.@@@)'')')
      READ(*,'(A)',ERR=1) FNAME
      OPEN(8,FILE=FNAME,STATUS='NEW')
        WRITE(8,'(A80)') MEMO(1)
        WRITE(8,'(A80)') MEMO(2)
        WRITE(8,'(A80)') MEMO(3)
        WRITE(8,'(A80)') MEMO(4)
        WRITE(8,'(2F12.5)') CP,CV
        WRITE(8,'(5F12.5)') CENTER,GG,QQ,E,GAMMA
        WRITE(8,'(4F12.5)') XDEG,YDEG,XXG,YYG
        WRITE(8,'(4F12.5)') WD(1),WD(2),WD(3),WD(4)
        WRITE(8,'(4F12.5)') WD(5),WD(6),WD(7),WD(8)
        WRITE(8,'(5I3)') NCHANL,IPS,KSWTCH,LSWTCH(1),LSWTCH(2)
        WRITE(8,'(4F12.5)') AMIN,AMAX,BMIN,BMAX
        WRITE(8,'(2I5)') NPA,NPB
        WRITE(8,'(4I5)') IFI,IFF,IWI,IWF
        WRITE(8,'(F12.5)') WW
        WRITE(8,'(6F12.7)') (YOBS(K),K=1,6)
        WRITE(*,'(6F12.7)') (YOBS(K),K=1,6)
        WRITE(8,'(6F12.7)') (YFIT(L),L=1,6)
        WRITE(8,'(6F12.7)') (P(M),M=1,6)
        WRITE(8,'(2F12.7)') SQKAI,AAV
```

```
        CLOSE(8)
        WRITE(*,'('' Filing is finished'')')
      RETURN
      END
*****************************************************************************
* Calculated spectral curve plotting subroutines
*****************************************************************************
      SUBROUTINE PLTINI
*
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /LINES/CV,WD(8),SF(512),YFIT(512),SQKAI
      COMMON /OBSDT/YOBS(1024),NCHANL,CP
      DIMENSION ST(10)
      INTEGER VD_ID,WD_ID
      INTEGER*2 KEYBUF(2)
      COMMON /SCREN/ VD_ID,WD_ID,IATRBB,INDEXW,KEYBUF
      REAL*4 XR,XL,YS1,YS2,PX,PY,SHIFT,ST
      CHARACTER STRING*80,C(10)*4,CC*10
        INCLUDE 'SYS$LIBRARY:UISENTRY'
        INCLUDE 'SYS$LIBRARY:UISUSRDEF'
        CALL UIS$SET_LINE_STYLE(VD_ID,0,10,'FFF0FFF0'X)
        CALL UIS$PLOT(VD_ID,10,0.0,17.0,25.6,17.0)
*-----------------------------------------------------------
        XL=0.0
        XR=25.6
      IF(NCHANL.EQ.512)THEN
          IVMAX=ANINT(512.0/SNGL(CV))
          DX=0.05
        ELSE
          IVMAX=ANINT(256.0/SNGL(CV))
          DX=0.1
        ENDIF
         IV=IVMAX/2
      DO 20 I=1,2
          IF (I.EQ.1) THEN
               PY=8.0
                 YS1=0.4
            ELSE
               PY=17.5
                 YS1=-0.4
            ENDIF
        DO 10 J=-IV,IV
           PX=XL+SNGL(CP)*DX-REAL(J-1)*CV*DX
           CALL UIS$PLOT(VD_ID,0,PX,PY,PX,PY+YS1)
   10   CONTINUE
       CALL UIS$PLOT(VD_ID,0,XL,PY,XL+25.6,PY)
            PX=XL+SNGL(CP)*DX
       CALL UIS$CIRCLE(VD_ID,INDEXW,PX,PY,0.1)
   20 CONTINUE
*------------------------------------------------------------------
      PY=7.5
         PX=XL+20.0
         CALL UIS$TEXT(VD_ID,0,'(mm/sec)',PX,PY)
      PY=7.8
      DO 30 I=-IV,IV
         ST(I)=REAL(I)
         PX=XL+SNGL(CP)*DX+ST(I)*CV*DX-0.4
         ENCODE(9,1000,C(I))ST(I)
 1000 FORMAT(F4.1)
         CALL UIS$TEXT(VD_ID,0,C(I),PX,PY)
   30 CONTINUE
      RETURN
      END
****************************************************************************
```

```
      SUBROUTINE PLOTNG
*
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /PARAM1/CENTER,GG,QQ,E,XDEG,YDEG,XXG,YYG
      COMMON /PARAM2/GAMMA,IPS
      COMMON /PARAM3/AMIN,AMAX,BMIN,BMAX,NPA,NPB,NP,KSWTCH,LSWTCH(2)
      COMMON /REGION/IFI,IFF,IWI,IWF,WW
      CHARACTER YN*1
      COMMON /LINES/CV,WD(8),SF(512),YFIT(512),SQKAI
      COMMON /PROB/P(226),AAV
      INTEGER VD_ID,WD_ID
      INTEGER*2 KEYBUF(2)
      COMMON /SCREN/ VD_ID,WD_ID,IATRBB,INDEXW,KEYBUF
      COMMON /OBSDT/YOBS(1024),NCHANL,CP
      REAL*4 XR,XL,XS,YS,XS1,YS1,BG,SCL,DX
        INCLUDE 'SYS$LIBRARY:UISENTRY'
        INCLUDE 'SYS$LIBRARY:UISUSRDEF'
       CALL UIS$POP_VIEWPORT(WD_ID)
         BG=17.0
         XL=0.0
         XR=25.6
         DX=0.1
        IF(NCHANL.EQ.512)DX=0.05
      DO 20 L=IFI,IFF
         XS=REAL(L)*DX+XL
          IF(XS.GT.XR) XS=XR
         YS=BG-SNGL(YFIT(L))
         XS1=XS+DX
         YS1=BG-SNGL(YFIT(L+1))
          IF(XS1.GT.XR) XS1=XR
        CALL UIS$PLOT(VD_ID,IATRBB,XS,YS,XS1,YS1)
   20 CONTINUE
       YS=8.0
       YS1=17.5
       IF(LSWTCH(1).EQ.4)THEN
          DELTAA=(AMAX-AMIN)/NPA
          DO 30 I=1,NPA+1
            CENTER=DBLE(I-1)*DELTAA+AMIN
            XS=XL+SNGL(CENTER)*DX
            CALL UIS$PLOT(VD_ID,IATRBB,XS,YS+0.5,XS,YS-0.5)
            CALL UIS$PLOT(VD_ID,IATRBB,XS,YS1+0.5,XS,YS1-0.5)
   30    CONTINUE
         ELSE
          IF(LSWTCH(2).EQ.4)THEN
             DELTAB=(BMAX-BMIN)/NPB
             DO 40 I=1,NPB+1
                CENTER=DBLE(I-1)*DELTAB+BMIN
                XS=XL+SNGL(CENTER)*DX
                CALL UIS$PLOT(VD_ID,IATRBB,XS,YS+0.5,XS,YS-0.5)
                CALL UIS$PLOT(VD_ID,IATRBB,XS,YS1+0.5,XS,YS1-0.5)
   40        CONTINUE
          ELSE
               XS=XL+SNGL(CENTER)*DX
               CALL UIS$PLOT(VD_ID,IATRBB,XS,YS+0.5,XS,YS-0.5)
               CALL UIS$PLOT(VD_ID,IATRBB,XS,YS1+0.5,XS,YS1-0.5)
          ENDIF
         ENDIF
       DO 50 L=1,NCHANL
         XS=REAL(L)*DX+XL
         YS=BG-SNGL(YOBS(L))
        CALL UIS$CIRCLE(VD_ID,INDEXW,XS,YS,0.05)
   50 CONTINUE
      IATRBB=IATRBB+1
      IF(IATRBB.GT.8)IATRBB=3
```

```
      RETURN
      END
******************************************************************************
*
* Subroutine for solving the normal equations
*
      SUBROUTINE GAUSS(A,N,AN,EPS)
*
*  Gauss-Jordan method:
*    A:  coefficient matrix of the normal equations.
*    N:  dimension of matrix to be solved.
*    AN: column vector on the right hand-side of the normal equations.
*          The solution is stored in this vector.
*    EPS: precision to regard as zero elements.
*
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(226,226),AN(226),NWK(226)
      IF(EPS) 1,2,2
    1     EPS=1.0D-14
    2     EPSS=EPS*EPS
      DO 10 I=1,N
          NWK(I)=0
   10 CONTINUE
      DO 20 K=1,N
          BIGST=-1.0D+00
       DO 30 I=1,N
         IF(NWK(I)) 3,3,30
    3    IF(BIGST-DABS(A(I,K))) 4,30,30
    4      L=I
           BIGST=DABS(A(I,K))
   30  CONTINUE
         IF(BIGST-EPS) 5,5,6
    6      PIVOT=A(L,K)
           NWK(L)=K
           W=-1.0D+00/PIVOT
         IF(K-N) 7,8,8
    7      K1=K+1
       DO 40 J=K1,N
           X=A(L,J)*W
         IF(DABS(X)-EPSS) 9,9,11
   11     DO 50 I=1,N
           A(I,J)=A(I,J)+A(I,K)*X
   50     CONTINUE
    9     A(L,J)=-X
   40  CONTINUE
    8      X=AN(L)*W
         IF(DABS(X)-EPSS) 12,12,13
   13  DO 60 I=1,N
           AN(I)=AN(I)+A(I,K)*X
   60  CONTINUE
   12      AN(L)=-X
   20 CONTINUE
      DO 70 I=1,N
          A(I,1)=AN(I)
   70 CONTINUE
      DO 80 I=1,N
          AN(NWK(I))=A(I,1)
   80 CONTINUE
      RETURN
    5 WRITE(*,'('' Matrix is nearly singular.'')')
      RETURN
      END
*
* Hamiltonian Matrix formation and diagonalization
```

```
*     ---------------------------------------------------------------
*     The following subroutines MATRIX,UNIT,POEDER,SINGLE,POSITN,
*       and SPECTR are coded in ALGOL by Prof.Toshizo Fujita for a
*         computer set in Institute for Solid State Physics, The
*           University of Tokyo (1965). Their translation into
*             FORTRAN was done by N.Hashitsume for OKIMINITAK7000
*               computer (1974).
*     ---------------------------------------------------------------
*
      SUBROUTINE MATRIX
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 KT
      COMMON /MATRX/KT(8,8),U(8,8)
      COMMON /STATE/D(4,4,2),W(2,2,2),H(4),V(2)
      COMMON /PARAM1/CENTER,GG,QQ,E,XDEG,YDEG,XXG,YYG
*
* R3=square-root of 3, R3H=R3/2
* RAD=convertor from degree to radian
* R0=g(ground)/g(excited)/2, g(ground)=0.1808412,g(excited)=0.103273
* RK=convertor from kOe to mm/sec
*
      RAD=0.01745329252D+00
      R3=1.732050808D+00
      R3H=0.866025403D+00
      R0=0.8755492D+00
      RK=0.6777323088D-02
       X=XDEG*RAD
       Y=YDEG*RAD
        SX=DSIN(X)
        CX=DCOS(X)
        SY=DSIN(Y)
        CY=DCOS(Y)
        SC=SX*CY
        SS=SX*SY
       G=GG*RK
       QS=QQ/6.0D+00
      N=8
          CALL UNIT(KT,U,N)
       P=G*CX
       Q=3.0D+00*QS
        WK=-1.5D+00*P+Q
         KT(1,1)=WK
         KT(5,5)=WK
        WK=-0.5D+00*P-Q
         KT(2,2)=WK
         KT(6,6)=WK
       WK=0.5D+00*P-Q
         KT(3,3)=WK
         KT(7,7)=WK
       WK=1.5D+00*P+Q
         KT(4,4)=WK
         KT(8,8)=WK
      P=-G*SC
         KT(2,3)=P
         KT(3,2)=P
         KT(6,7)=P
         KT(7,6)=P
      P=P*R3H
         KT(1,2)=P
         KT(2,1)=P
         KT(3,4)=P
         KT(4,3)=P
         KT(5,6)=P
         KT(6,5)=P
```

```
           KT(7,8)=P
           KT(8,7)=P
         WK=QS*E*R3
           KT(1,3)=WK
           KT(3,1)=WK
           KT(2,4)=WK
           KT(4,2)=WK
           KT(5,7)=WK
           KT(7,5)=WK
           KT(6,8)=WK
           KT(8,6)=WK
        P=-G*SS
           KT(2,7)=P
           KT(7,2)=P
           KT(3,6)=-P
           KT(6,3)=-P
        P=P*R3H
           KT(1,6)=P
           KT(6,1)=P
           KT(2,5)=-P
           KT(5,2)=-P
           KT(3,8)=P
           KT(8,3)=P
           KT(4,7)=-P
           KT(7,4)=-P
            CALL JACOBI(KT,U,N)
        I=0
        DO 30003 K=1,7
          DO 30004 L=K+1,8
            IF (DABS(KT(K,K)-KT(L,L)).LT.10.0D-10) THEN
              I=I+1
              H(I)=KT(K,K)
              DO 30005 J=1,4
                D(I,J,1)=U(K,J)
                D(I,J,2)=U(K,J+4)
30005         CONTINUE
            END IF
30004     CONTINUE
30003   CONTINUE
        N=4
            CALL UNIT(KT,U,N)
        P=G*R0
        Q=P*CX
           KT(1,1)=Q
           KT(2,2)=-Q
           KT(3,3)=Q
           KT(4,4)=-Q
         WK=P*SC
           KT(1,2)=WK
           KT(2,1)=WK
           KT(3,4)=WK
           KT(4,3)=WK
        Q=-P*SS
           KT(1,4)=-Q
           KT(4,1)=-Q
           KT(2,3)=Q
           KT(3,2)=Q
            CALL JACOBI(KT,U,N)
        I=0
        DO 30006 K=1,3
          DO 30007 L=K+1,4
            IF (DABS(KT(K,K)-KT(L,L)).LT.10.0D-10) THEN
              I=I+1
              V(I)=KT(K,K)
```

```
            DO 30008 J=1,2
              W(I,J,1)=U(K,J)
              W(I,J,2)=U(K,J+2)
30008       CONTINUE
          END IF
30007   CONTINUE
30006 CONTINUE
      RETURN
      END
*
* Unit matrices formation subroutine
*
      SUBROUTINE UNIT(KT,U,N)
*
      REAL*8 KT(8,8),U(8,8)
      DO 30009 I=1,N
        DO 30010 J=1,N
          IF (I.EQ.J) THEN
            U(I,I)=1.0
          ELSE
            U(I,J)=0.0
            KT(I,J)=0.0
          END IF
30010   CONTINUE
30009 CONTINUE
      RETURN
      END
*
* Matrix diagonalization subroutine
*
      SUBROUTINE JACOBI(KT,U,N)
*
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER H
      REAL*8 KT(8,8),U(8,8),MAX
*
* KT is a real symmetric N by N matrix.
* U is a unit N by N matrix.
* R2I=Inverse of square root of 2
*
      R2I=0.7071067812D+00
30011 CONTINUE
        MAX=0.0
        DO 30012 H=2,N
          DO 30013 K=1,H-1
            AKT=DABS(KT(H,K))
            IF (AKT  .GE.MAX) THEN
              MAX=AKT
              I=H
              J=K
            END IF
30013     CONTINUE
30012   CONTINUE
        IF (MAX.LE.   10.0D-10) THEN
          RETURN
        END IF
        P=(KT(J,J)-KT(I,I))/KT(I,J)*0.5D+00
        C2=DABS(P)/DSQRT(1.0+P*P)
        C1=DSQRT(0.5D+00*(1.0+C2))
        IF (DABS(P) .LT.10.0D-8) THEN
          S1=R2I
        ELSE
          S1=C2/(2.0*C1*P)
        END IF
```

```
          DO 30014 K=1,N
            IF (K.EQ.J) THEN
              P=KT(J,J)
              Q=KT(I,I)
              KT(J,J)=P*C1*C1+Q*S1*S1+KT(I,J)*2.0*S1*C1
              KT(I,I)=P+Q-KT(J,J)
            ELSE
              IF (K.NE.I) THEN
                P=KT(J,K)
                Q=KT(I,K)
                WK=P*C1+Q*S1
                KT(J,K)=WK
                KT(K,J)=WK
                WK=Q*C1-P*S1
                KT(I,K)=WK
                KT(K,I)=WK
              END IF
            END IF
30014     CONTINUE
          KT(I,J)=0.0
          KT(J,I)=0.0
          DO 30015 K=1,N
            P=U(J,K)
            Q=U(I,K)
            U(J,K)=P*C1+Q*S1
            U(I,K)=Q*C1-P*S1
30015     CONTINUE
          EPS=0.0
          DO 30016 I=2,N
            DO 30017 J=1,I-1
              AKT=DABS(KT(I,J))
              IF (AKT  .GE.EPS) THEN
                EPS=AKT
              END IF
30017       CONTINUE
30016     CONTINUE
        IF (.NOT.(EPS.LT.10.0D-10))GOTO 30011
        RETURN
        END
*
* Spectral intensity calculation in case of powder
*
        SUBROUTINE POWDER
*
        IMPLICIT REAL*8(A-H,O-Z)
        REAL*8 IT
        COMMON /POSIT/IT(8,8)
        COMMON /STATE/D(4,4,2),W(2,2,2),H(4),V(2)
         KINDSL=0
*
* R2=square root of 2
* R6D=2/(square root of 6)
* R6Q=2*R6D
*
        R2=1.414213562D+00
        R6D=0.81649658D+00
        R6Q=1.632993162D+00
        DO 30020 I=1,4
          DO 30021 K=1,2
            P=R2*(D(I,4,1)*W(K,2,1)+D(I,4,2)*W(K,2,2)        -D(I,1,1)*W(K,
     *      1,1)-D(I,1,2)*W(K,1,2)) +R6D*(D(I,3,1)*W(K,1,1)+D(I,3,2)*W(K,1
     *      ,2)        -D(I,2,1)*W(K,2,1)-D(I,2,2)*W(K,2,2))
            Q=P*P
            P=R2*(D(I,4,1)*W(K,2,2)-D(I,4,2)*W(K,2,1)        -D(I,1,1)*W(K,
```

```
     *     1,2)+D(I,1,2)*W(K,1,1))  +R6D*(D(I,3,1)*W(K,1,2)-D(I,3,2)*W(K,1
     *     ,1)        -D(I,2,1)*W(K,2,2)+D(I,2,2)*W(K,2,1))
           Q=Q+P*P
           P=R6Q*(D(I,2,1)*W(K,1,1)+D(I,2,2)*W(K,1,2)         +D(I,3,1)*W(
     *     K,2,1)+D(I,3,2)*W(K,2,2))
           Q=Q+P*P
           P=R6Q*(D(I,2,1)*W(K,1,2)-D(I,2,2)*W(K,1,1)         +D(I,3,1)*W(
     *     K,2,2)-D(I,3,2)*W(K,2,1))
           Q=Q+P*P
           P=-R2*(D(I,1,1)*W(K,1,1)+D(I,1,2)*W(K,1,2)         +D(I,4,1)*W(
     *     K,2,1)+D(I,4,2)*W(K,2,2))  -R6D*(D(I,2,1)*W(K,2,1)+D(I,2,2)*W(
     *     K,2,2)         +D(I,3,1)*W(K,1,1)+D(I,3,2)*W(K,1,2))
           Q=Q+P*P
           P=-R2*(D(I,1,1)*W(K,1,2)-D(I,1,2)*W(K,1,1)         +D(I,4,1)*W(
     *     K,2,2)-D(I,4,2)*W(K,2,1))  -R6D*(D(I,2,1)*W(K,2,2)-D(I,2,2)*W(
     *     K,2,1)         +D(I,3,1)*W(K,1,2)-D(I,3,2)*W(K,1,1))
           IT(I,K)=Q+P*P
30021    CONTINUE
30020 CONTINUE
      RETURN
      END
*
* Spectral intensity calculation in case of single crystal
*
      SUBROUTINE SINGLE
*
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 IT
      COMMON /POSIT/IT(8,8)
      COMMON /STATE/D(4,4,2),W(2,2,2),H(4),V(2)
      COMMON /PARAM1/CENTER,GG,QQ,E,XDEG,YDEG,XXG,YYG
      DIMENSION A(3,4,2),B(3,4,2),O1(4,2),O2(4,2)
       KINDSL=1
*
* R6I=inverse of square root of 6
* R12I=inverse of square root of 12
*
        RAD=0.01745329252D+00
        R2I=0.7071067812D+00
        R6I=0.4082482905D+00
        R12I=0.2886751346D+00
      XG=XXG*RAD
       S=DSIN(XG)
       C=DCOS(XG)
        SC=S*C
        SS=S*S
      YG=YYG*RAD
       ES=DSIN(YG)
       EC=DCOS(YG)
        ES2=2.0D+00*ES*EC
        EC2=2.0D+00*EC*EC-1.0D+00
       A(1,1,1)=0.5D+00*R2I*SC*EC
      WK=R6I*SS
       A(1,2,1)=WK
       A(1,3,2)=WK
       A(1,3,1)=-0.5D+00*R6I*SC*EC
       A(1,4,1)=0.0
       A(1,1,2)=0.0
       A(1,4,2)=-A(1,1,1)
       A(1,2,2)=-A(1,3,1)
      WK=-0.5D+00*R2I*SC*ES
       B(1,1,1)=WK
       B(1,4,2)=WK
      WK=-0.5D+00*R6I*SC*ES
```

```
       B(1,3,1)=WK
       B(1,2,2)=WK
       B(1,2,1)=0.0
       B(1,4,1)=0.0
       B(1,1,2)=0.0
       B(1,3,2)=0.0
       A(2,1,1)=0.5D+00*(1.0-0.5D+00*SS)
      WK=R12I*SC*EC
       A(2,2,1)=WK
       A(2,3,2)=WK
       A(2,3,1)=0.5D+00*R12I*SS*EC2
       A(2,4,1)=0.0
       A(2,1,2)=0.0
       A(2,2,2)=R12I*(1.0-0.5D+00*SS)
       A(2,4,2)=0.25D+00*SS*EC2
       B(2,1,1)=0.0
       B(2,4,1)=0.0
       B(2,1,2)=0.0
       B(2,2,2)=0.0
      WK=R12I*SC*ES
       B(2,2,1)=WK
       B(2,3,2)=WK
       B(2,3,1)=0.5D+00*R12I*SS*ES2
       B(2,4,2)=0.25D+00*SS*ES2
       A(3,1,1)=0.25D+00*SS*EC2
      WK=-R12I*SC*EC
       A(3,2,1)=WK
       A(3,3,2)=WK
       A(3,3,1)=R12I*(1.0-0.5D+00*SS)
       A(3,4,1)=0.0
       A(3,1,2)=0.0
       A(3,2,2)=0.5D+00*R12I*SS*EC2
       A(3,4,2)=0.5D+00*(1.0-0.5D+00*SS)
       B(3,1,1)=-0.25D+00*SS*ES2
      WK=R12I*SC*ES
       B(3,2,1)=WK
       B(3,3,2)=WK
       B(3,3,1)=0.0
       B(3,4,1)=0.0
       B(3,1,2)=0.0
       B(3,4,2)=0.0
       B(3,2,2)=-0.5D+00*R12I*SS*ES2
      DO 30023 I=1,4
        DO 30024 K=1,2
          TIK=0.0
          DO 30025 M=1,3
            S=0.0
            P=0.0
            DO 30026 J=1,4
              DO 30027 L=1,2
                O1(J,L)=D(I,J,1)*W(K,L,1)+D(I,J,2)*W(K,L,2)
                O2(J,L)=D(I,J,1)*W(K,L,2)-D(I,J,2)*W(K,L,1)
                S=S+O1(J,L)*A(M,J,L)-O2(J,L)*B(M,J,L)
                P=P+O1(J,L)*B(M,J,L)+O2(J,L)*A(M,J,L)
30027         CONTINUE
30026       CONTINUE
            TIK=TIK+S*S+P*P
30025     CONTINUE
          IT(I,K)=24.0D+00*TIK
30024   CONTINUE
30023 CONTINUE
      RETURN
      END
*
```

```
* Determination of line positions and intensities
*
      SUBROUTINE POSITN
*
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 IT,MAX,MIN,MM
      COMMON /POSIT/IT(8,8)
      COMMON /STATE/D(4,4,2),W(2,2,2),H(4),V(2)
      COMMON /SPECT/AIN(8),HM(8),HD(8),IN(8),JN(8)
        MAX=0.0
        MIN=0.0
      DO 30028 I=1,2
        DO 30029 J=1,4
          HV=H(J)-V(I)
          IF (HV.GT.MAX) THEN
            MAX=HV
          END IF
          IF (HV.LT.MIN) THEN
            MIN=HV
          END IF
30029   CONTINUE
30028 CONTINUE
      MM=MAX-MIN
      DO 30030 I=1,2
        DO 30031 J=1,4
          K=J+(I-1)*4
          IN(K)=I
          JN(K)=J
          AIN(K)=IT(J,I)
          HV=H(J)-V(I)
          HM(K)=HV/MM
          HD(K)=HV
30031   CONTINUE
30030 CONTINUE
      DO 30032 K=1,7
        DO 30033 J=K+1,8
          IF (HD(K).LT.HD(J)) THEN
            HDW=HD(K)
            HD(K)=HD(J)
            HD(J)=HDW
            INW=IN(K)
            IN(K)=IN(J)
            IN(J)=INW
            JNW=JN(K)
            JN(K)=JN(J)
            JN(J)=JNW
            HMW=HM(K)
            HM(K)=HM(J)
            HM(J)=HMW
            AINW=AIN(K)
            AIN(K)=AIN(J)
            AIN(J)=AINW
          END IF
30033   CONTINUE
30032 CONTINUE
      RETURN
      END
*
* Spectral line shape calculation
*
      SUBROUTINE SPECTR
*
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /PARAM1/CENTER,GG,QQ,E,XDEG,YDEG,XXG,YYG
```

```
      COMMON /REGION/IFI,IFF,IWI,IWF,WW
      COMMON /SPECT/AIN(8),HM(8),HD(8),IN(8),JN(8)
      COMMON /LINES/CV,WD(8),SF(512)
      COMMON /OBSDT/YOBS(1024),NCHANL,CP
      DO 30037 K=1,512
          SF(K)=0
30037 CONTINUE
      DO 30038 I=1,8
          WD2=(WD(I)*CV)**2*0.25D+00
          AI=AIN(I)/WD(I)
          CENTRM=CENTER
        DO 30039 K=IFI,IFF
          X=(DBLE(K)-CENTRM-CV*HD(I))**2
          SF(K)=SF(K)+AI*WD2/(X+WD2)
30039   CONTINUE
30038 CONTINUE
      RETURN
      END
*
* Estimated probability distribution plotting subroutine
*
      SUBROUTINE PRBPLT1
*
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /PARAM1/CENTER,GG,QQ,E,XDEG,YDEG,XXG,YYG
      COMMON /PARAM2/GAMMA,IPS
      COMMON /PARAM3/AMIN,AMAX,BMIN,BMAX,NPA,NPB,NP,KSWTCH,LSWTCH(2)
      COMMON /REGION/IFI,IFF,IWI,IWF,WW
      CHARACTER YN*1
      COMMON /LINES/CV,WD(8),SF(512),YFIT(512),SQKAI
      COMMON /PROB/P(226),AAV
      INTEGER VD_ID,WD_ID
      INTEGER*2 KEYBUF(2)
      COMMON /SCREN/ VD_ID,WD_ID,IATRBB,INDEXW,KEYBUF
      REAL*4 XL,XR,YL,YU,XS,YS,XS1,YS1,BG,SCL,DX,PMAX,PIP
      CHARACTER STRING*80
      CHARACTER*6 C(4)
        INCLUDE 'SYS$LIBRARY:UISENTRY'
        INCLUDE 'SYS$LIBRARY:UISUSRDEF'
        CALL UIS$POP_VIEWPORT(WD_ID)
*-------------------------------------------------------------------
        PMAX=0.0
      DO 10 IP=1,NP
           PIP=SNGL(P(IP))
        IF(PIP.GT.PMAX) PMAX=PIP
   10 CONTINUE
         XL=4.5
         XR=14.5
         YL=1.0
         YU=6.5
          ENCODE(8,100,C(1)) AMIN
          ENCODE(8,100,C(2)) AMAX
          XAV=XL+10.0*(AAV-AMIN)/(AMAX-AMIN)
          DX=(XR-XL)/REAL(NPA)
          SCL=5.0/PMAX
  100   FORMAT(F6.2)
       CALL UIS$TEXT(VD_ID,2,C(1),4.0,0.8)
       CALL UIS$TEXT(VD_ID,2,C(2),14.0,0.8)
       CALL UIS$PLOT(VD_ID,0,XAV,YL,XAV,YU)
          XS=XL
          XS1=XR
       CALL UIS$PLOT(VD_ID,0,XS,YL,XS1,YL)
       CALL UIS$PLOT(VD_ID,0,XS,YU,XS1,YU)
        DO 20 IP=1,NP
```

```
          XS=XL+REAL(IP-1)*DX
          YS=YL+0.2
          YS1=YU-0.2
          CALL UIS$PLOT(VD_ID,0,XS,YL,XS,YS)
          CALL UIS$PLOT(VD_ID,0,XS,YU,XS,YS1)
   20   CONTINUE
          XS=XL
          XS1=XL+0.5*DX
          YS=YL+SNGL(P(1))*SCL
          YS1=YL+SNGL(P(2))*SCL
          CALL UIS$PLOT(VD_ID,IATRBB,XS,YL,XS,YS,XS1,YS,XS1,YS1)
        DO 30 IP=2,NP-1
          XS=XL+REAL(IP-1)*DX-0.5*DX
          XS1=XS+DX
          YS=YL+SNGL(P(IP))*SCL
          YS1=YL+SNGL(P(IP+1))*SCL
          CALL UIS$PLOT(VD_ID,IATRBB,XS,YS,XS1,YS,XS1,YS1)
   30   CONTINUE
          XS=XR-0.5*DX
          XS1=XR
          YS=YL+SNGL(P(NP))*SCL
          CALL UIS$PLOT(VD_ID,IATRBB,XS,YS,XS1,YS,XS1,YL)
      IATRBB=IATRBB+1
      IF(IATRBB.GT.8)IATRBB=3
      RETURN
      END
******************************************************************
      SUBROUTINE PRBPLT2
*
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 PB,PBMAX,XL,XR,YL,YU,XS,YS,XS1,YS1,SCLA,SCLB,HEIGHT,
     *       DELTAH,VH,UA1,UA2,VA1,VA2,SLA,SLB,ELA,ELB,RT,PROBIP
      COMMON /CONTR/PB(45,45),KPB(45,45),SCLA,SCLB,HEIGHT,SLA,SLB,
     *              KA,KB,LA,LB,KS,KV,IREP,XL,XR,YL,YU,UA1,UA2
      COMMON /PARAM3/AMIN,AMAX,BMIN,BMAX,NPA,NPB,NP,KSWTCH,LSWTCH(2)
      COMMON /PROB/P(226),AAV
      CHARACTER*6 C(4)
      INTEGER VD_ID,WD_ID
      COMMON /SCREN/ VD_ID,WD_ID,IATRBB,INDEXW
        INCLUDE 'SYS$LIBRARY:UISENTRY'
        INCLUDE 'SYS$LIBRARY:UISUSRDEF'
            PBMAX=0.0
      DO 1 IP=1,NP
            PROBIP=SNGL(P(IP))
          IF(PROBIP.GT.PBMAX) PBMAX=PROBIP
    1 CONTINUE
            XL=4.5
            XR=14.5
            YL=1.0
            YU=6.0
          CALL UIS$PLOT(VD_ID,0,XL,YL,XL,YU,XR,YU,XR,YL,XL,YL)
          ENCODE(8,100,C(1)) AMIN
          ENCODE(8,100,C(2)) AMAX
  100       FORMAT(F6.2)
           CALL UIS$TEXT(VD_ID,2,C(1),4.0,0.8)
           CALL UIS$TEXT(VD_ID,2,C(2),14.0,0.8)
          ENCODE(8,100,C(3)) BMIN
          ENCODE(8,100,C(4)) BMAX
           CALL UIS$TEXT(VD_ID,2,C(3),3.2,1.2)
           CALL UIS$TEXT(VD_ID,2,C(4),3.2,6.2)
        DO 3 IA=1,NPA+1
          DO 4 IB=1,NPB+1
            L=IB+(NPB+1)*(IA-1)
            PB(IA,IB)=SNGL(P(L))
```

```
    4     CONTINUE
    3   CONTINUE
*     -----------------------------------------------------
*      The following part of this program and the next
*         subroutine SEARCH are coded by Prof.Masatake Mori
*           ( "Kyokusen to kyokumen",Kyoiku-shuppan,1974,pp.
*              136-137)
*     -----------------------------------------------------
            SCLA=(XR-XL)/REAL(NPA)
            SCLB=(YU-YL)/REAL(NPB)
       DO 1000 IA=1,NPA
            XS=SCLA*(IA-1)+XL
            CALL UIS$PLOT(VD_ID,0,XS,YL,XS,YU)
 1000  CONTINUE
       DO 2000 IB=1,NPB
            YS=SCLB*(IB-1)+YL
            CALL UIS$PLOT(VD_ID,0,XL,YS,XR,YS)
 2000 CONTINUE
       DO 5 IA=1,NPA+1
         DO 6 IB=1,NPB+1
            KPB(IA,IB)=0
    6    CONTINUE
    5  CONTINUE
            DELTAH=PBMAX/10.0
            HEIGHT=DELTAH*2.0
            IATRBB=2
       DO 7 IH=3,9
            KERR=0
            KV=IH
            HEIGHT=HEIGHT+DELTAH
            IATRBB=IATRBB+1
            IF(IATRBB.EQ.9) IATRBB=2
         DO 8 IA=1,NPA+1
            VH=PB(IA,1)-HEIGHT
          IF(VH) 10,11,9
    9       KG=1
            GO TO 12
   10       KG=2
            GO TO 12
   11       KG=3
   12     CONTINUE
         IF(KPB(IA,1).NE.KV) GO TO 13
         IF((PB(IA,2)-HEIGHT).LE.0.0) KG=2
   13    CONTINUE
         DO 14 IB=2,NPB+1
            VH=PB(IA,IB)-HEIGHT
           IF(KPB(IA,IB).EQ.KV) GO TO 15
            GO TO (16,17,18), KG
   16      IF(VH) 19,19,14
   19      CONTINUE
            IREP=0
           DO 21 MH=1,2
            KA=IA
            KB=IB-1
             GO TO (22,23), MH
   22          IF(IA.EQ.NPA+1) GO TO 21
            LA=1
            KS=-1
             GO TO 24
   23          IF(IA.EQ.1) GO TO 21
            IF(IREP.GE.2) GO TO 21
            LA=-1
            KS=1
            KPB(KA,KB)=0
```

```
24         LB=0
           UA1=PB(KA,KB)
           UA2=PB(KA,KB+1)
           RT=(UA2-HEIGHT)/(UA1-UA2)
           SLA=SCLA*REAL(KA-1)
           SLB=SCLB*(REAL(KB)+RT)
           KERR=1
          CALL SEARCH
21      CONTINUE
           KG=2
         GO TO 14
17       IF(VH) 14,14,25
25       CONTINUE
        IREP=0
          DO 27 MH=1,2
           KA=IA
           KB=IB
           GO TO (28,29), MH
28         IF(IA.EQ.NPA+1) GO TO 27
           LA=1
           KS=1
           GO TO 30
29         IF(IA.EQ.1) GO TO 27
           IF(IREP.GE.2) GO TO 27
           LA=-1
           KS=-1
           KPB(KA,KB)=0
30         LB=0
           UA1=PB(KA,KB)
           UA2=PB(KA,KB-1)
           RT=(UA1-HEIGHT)/(UA1-UA2)
           SLA=SCLA*REAL(KA-1)
           SLB=SCLB*(REAL(KB-1)-RT)
           KERR=1
          CALL SEARCH
27        CONTINUE
           KG=1
          GO TO 14
15        GO TO (31,32,14), KG
31         KG=2
          IF(IB.EQ.NPB+1) GO TO 14
          IF((PB(IA,IB+1)-HEIGHT).GT.0.0) KG=1
          GO TO 14
32         KG=1
          IF(VH.LT.0.0) KG=2
          GO TO 14
18        IF(VH) 33,34,35
35         KG=1
           GO TO 36
33         KG=2
           GO TO 36
34         KG=3
36        CONTINUE
14       CONTINUE
 8      CONTINUE
          IF(KERR.EQ.0) THEN
           WRITE(*,'(''  Contour line HEIGHT='',F8.5,
  *                  ''  is not found.'')') HEIGHT
          ENDIF
  7   CONTINUE
      RETURN
      END
*
      SUBROUTINE SEARCH
```

```
*
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 PB,PBMAX,XL,XR,YL,YU,XS,YS,XS1,YS1,SCLA,SCLB,HEIGHT,
     *       DELTAH,VH,UA1,UA2,VA1,VA2,SLA,SLB,ELA,ELB,RT,PROBIP
      COMMON /CONTR/PB(45,45),KPB(45,45),SCLA,SCLB,HEIGHT,SLA,SLB,
     *              KA,KB,LA,LB,KS,KV,IREP,XL,XR,YL,YU,UA1,UA2
      COMMON /PARAM3/AMIN,AMAX,BMIN,BMAX,NPA,NPB,NP,KSWTCH,LSWTCH(2)
      COMMON /PROB/P(226),AAV
      INTEGER VD_ID,WD_ID
      COMMON /SCREN/ VD_ID,WD_ID,IATRBB,INDEXW
        INCLUDE 'SYS$LIBRARY:UISENTRY'
        INCLUDE 'SYS$LIBRARY:UISUSRDEF'
*-----------------------------------------------------
            LN=800
            IREP=0
      DO 10 L=2,LN
         IF(KPB(KA,KB).EQ.KV) GO TO 99
            KAV=KA+LA
         IF(KAV.LE.0) GO TO 98
         IF(KAV.GT.NPA+1) GO TO 98
            KBV=KB+LB
         IF(KBV.LE.0) GO TO 98
         IF(KBV.GT.NPB+1) GO TO 98
            VA1=PB(KAV,KBV)
         IF(VA1-HEIGHT) 1,1,2
1           IREP=IREP+1
            RT=(UA1-HEIGHT)/(UA1-VA1)
            ELA=SCLA*(REAL(KA-1)+RT*REAL(LA))
            ELB=SCLB*(REAL(KB-1)+RT*REAL(LB))
             XS=SLA+XL
             YS=SLB+YL
             XS1=ELA+XL
             YS1=ELB+YL
           CALL UIS$PLOT(VD_ID,IATRBB,XS,YS,XS1,YS1)
            SLA=ELA
            SLB=ELB
               LAV=LA
               LA=-KS*LB
               LB=KS*LAV
               UA2=VA1
         IF(IREP.GE.4) GO TO 98
        GO TO 10
2           IREP=0
            LAV=KS*LB
            LBV=-KS*LA
            KAV=KAV+LAV
            KBV=KBV+LBV
            VA2=PB(KAV,KBV)
         IF(VA2-HEIGHT) 3,3,4
3           RT=(VA1-HEIGHT)/(VA1-VA2)
            ELA=SCLA*(REAL(KA+LA-1)+RT*REAL(LAV))
            ELB=SCLB*(REAL(KB+LB-1)+RT*REAL(LBV))
             XS=SLA+XL
             YS=SLB+YL
             XS1=ELA+XL
             YS1=ELB+YL
           CALL UIS$PLOT(VD_ID,IATRBB,XS,YS,XS1,YS1)
             SLA=ELA
             SLB=ELB
            KPB(KA,KB)=KV
            KA=KA+LA
            KB=KB+LB
            UA1=VA1
            UA2=VA2
```

```
      GO TO 10
 4        LAV=KS*LB
          LBV=-KS*LA
          RT=(UA2-HEIGHT)/(UA2-VA2)
          ELA=SCLA*(REAL(KA+LAV-1)+RT*REAL(LA))
          ELB=SCLB*(REAL(KB+LBV-1)+RT*REAL(LB))
           XS=SLA+XL
           YS=SLB+YL
           XS1=ELA+XL
           YS1=ELB+YL
         CALL UIS$PLOT(VD_ID,IATRBB,XS,YS,XS1,YS1)
           SLA=ELA
           SLB=ELB
          KPB(KA,KB)=KV
          KA=KA+LA+LAV
          KB=KB+LB+LBV
          LA=LAV
          LB=LBV
          UA1=VA2
10 CONTINUE
      GO TO 99
98        KPB(KA,KB)=KV
99    CONTINUE
     RETURN
     END
*
* Moessbauer spectrum window deleting subroutine
*
     SUBROUTINE SCRFIN
*
     CHARACTER*1 YN
     INTEGER VD_ID,WD_ID
     COMMON /SCREN/ VD_ID,WD_ID,IATRBB,INDEXW
       INCLUDE 'SYS$LIBRARY:UISENTRY'
       INCLUDE 'SYS$LIBRARY:UISUSRDEF'
       CALL UIS$DELETE_DISPLAY(VD_ID)
     RETURN
     END
```