


11-1964

# LCR-2, a fortran lattice constant refinement program

Donald E. Williams  
*Iowa State University*

Follow this and additional works at: [http://lib.dr.iastate.edu/ameslab\\_isreports](http://lib.dr.iastate.edu/ameslab_isreports)

 Part of the [Chemistry Commons](#), and the [Computer Sciences Commons](#)

---

## Recommended Citation

Williams, Donald E., "LCR-2, a fortran lattice constant refinement program" (1964). *Ames Laboratory Technical Reports*. 102.  
[http://lib.dr.iastate.edu/ameslab\\_isreports/102](http://lib.dr.iastate.edu/ameslab_isreports/102)

This Report is brought to you for free and open access by the Ames Laboratory at Iowa State University Digital Repository. It has been accepted for inclusion in Ames Laboratory Technical Reports by an authorized administrator of Iowa State University Digital Repository. For more information, please contact [digirep@iastate.edu](mailto:digirep@iastate.edu).

---

# LCR-2, a fortran lattice constant refinement program

## **Abstract**

This report describes a computer program which refines lattice constants for any crystal symmetry by the least squares method. Several types of extrapolation functions are provided, and these functions can be used either singly or in combination. A complete error treatment is made, including covariances between parameters.

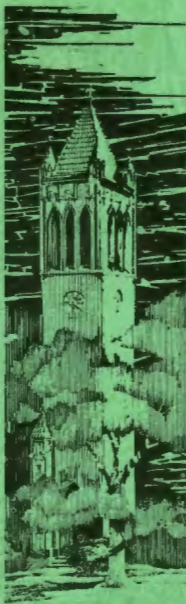
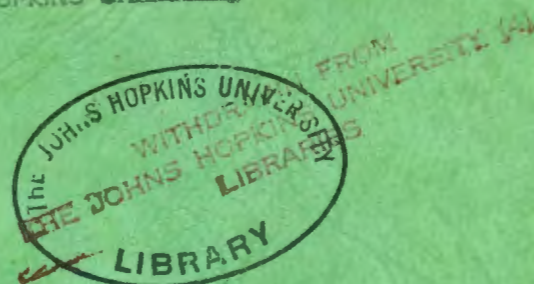
## **Disciplines**

Chemistry | Computer Sciences

Iowa  
SUST  
AL  
RDR  
IS-  
1052

THE LIBRARY  
THE JOHNS HOPKINS UNIVERSITY

IS-1052 ✓



IOWA STATE UNIVERSITY

LCR-2, A FORTRAN  
LATTICE CONSTANT REFINEMENT  
PROGRAM

by

Donald E. Williams

AMES LABORATORY

---

RESEARCH AND  
DEVELOPMENT  
REPORT

---

U.S.A.E.C.



IS-1052

Chemistry (UC-4)  
TID-4500, October 1, 1964

UNITED STATES ATOMIC ENERGY COMMISSION

Research and Development Report

LCR-2, A FORTRAN  
LATTICE CONSTANT REFINEMENT  
PROGRAM

by

Donald E. Williams

November, 1964

Ames Laboratory

at

Iowa State University of Science and Technology  
F. H. Spedding, Director  
Contract W-7405 eng-82

This report is distributed according to the category Chemistry (UC-4) as listed in TID-4500, October 1, 1964.

### 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.

Printed in USA. Price \$1.00 . Available from the

Office of Technical Services  
U. S. Department of Commerce  
Washington 25, D. C.

CONTENTS

	Page
ABSTRACT .....	5
MATHEMATICAL METHOD .....	5
EXTRAPOLATION FUNCTIONS .....	7
WEIGHTING FACTORS .....	8
INPUT DATA FORMATS .....	8

IS-1052

LCR-2, A FORTRAN LATTICE CONSTANT  
REFINEMENT PROGRAM

Donald E. Williams

ABSTRACT

This report describes a computer program which refines lattice constants for any crystal symmetry by the least squares method. Several types of extrapolation functions are provided, and these functions can be used either singly or in combination. A complete error treatment is made, including covariances between parameters.

MATHEMATICAL METHOD

A residual function,  $R$ , of the parameters,  $p_k$ , is defined by the equation  $R = \sum w [\Delta(p_k)]^2$ . This function is minimized by expanding the residuals  $\Delta(p_k)$  in a first order Taylor's series about the trial values, setting the partial derivatives of  $R$  equal to zero, and solving the resulting set of linear equations for the parameter corrections  $\epsilon_k$ . The summation is over the observed reflection data, and the weighting factors  $w$  are defined by  $w = 1/\sigma^2(\Delta)$ , where  $\sigma(\Delta)$  is the estimated standard deviation of  $\Delta$ . The linear equations may be written in matrix notation as  $\underline{C} \underline{\epsilon} = \underline{q}$ , where

$$C_{ij} = \sum w \frac{\partial \Delta}{\partial p_i} \frac{\partial \Delta}{\partial p_j} \quad \text{and} \quad q_i = \sum -w \Delta \frac{\partial \Delta}{\partial p_i} .$$

The parameter list defined by this program is the following: the first three parameters are the lengths of the unit cell edges in  $\text{\AA}$ , the second three parameters are the cell angles in degrees, and the last



four parameters are extrapolation function coefficients, to be defined later. Any combination of these ten parameters may be refined by the program.

The residual  $\Delta$  is defined by the equation

$$\Delta = (d^*)^2 - 4 \sin^2 \theta / \lambda^2 - 4 \Delta \theta \sin 2\theta / \lambda^2 ,$$

$$\Delta \theta = \sum_n^4 K_n f_n(\theta) .$$

Thus the systematic error in  $\theta$ ,  $\Delta\theta$ , is expressed as the sum of contributions from various sources having the functional form  $f_n(\theta)$  and coefficient  $K_n$ .

We now must find expressions for the various derivatives of  $d^*$ . The quantity  $d^*$  may easily be expressed as a function of the reciprocal lattice constants; however, since we have chosen to refine the direct constants, we must express  $d^*$  and its derivatives as functions of the direct lattice constants. In terms of the metric tensor  $\underline{g}$ , where

$$\underline{g} = \begin{pmatrix} p_1 p_1 & p_1 p_2 \cos p_6 & p_1 p_3 \cos p_5 \\ p_1 p_2 \cos p_6 & p_2 p_2 & p_2 p_3 \cos p_4 \\ p_1 p_3 \cos p_5 & p_2 p_3 \cos p_4 & p_3 p_3 \end{pmatrix} , \quad (d^*)^2 = \underline{h} \underline{g}^{-1} \underline{h} , \quad \text{where}$$

$\underline{h}$  is the vector of the Miller indices of the reflection.

Although the expressions for the derivatives of  $\underline{g}^{-1}$  with respect to the  $p_k$  may be obtained in closed form, they are rather lengthy. This program evaluates these derivatives numerically by incrementing the  $p_k$ . The increments chosen are 0.005 Å for the cell edges, and 0.5° for the cell angles. The derivatives of  $\underline{g}^{-1}$  need be calculated only once,

since  $\frac{\partial (d^*)^2}{\partial p_k} = \sum_i^3 \sum_j^3 h_i h_j \frac{\partial g_{ij}^{-1}}{\partial p_k}$ . Thus a table may be constructed with the 54 derivatives of each of the elements of  $\underline{g}^{-1}$  with respect to  $p_k$ . Since  $\underline{g}^{-1}$  is symmetric, only 36 of these are independent.

The derivatives of  $\Delta$  with respect to the extrapolation coefficients  $K_n$  are simply the extrapolation functions  $f_n(\theta)$ , multiplied by  $4 \sin 2\theta / \lambda^2$ .

### EXTRAPOLATION FUNCTIONS

A discussion of extrapolation functions suitable for typical experimental arrangements is given in the International Tables for X-Ray Crystallography, Vol. II, p. 216 (1959). References to the extensive literature on this subject may also be found there.

Program LCR-2 provides a choice of any combination of the four extrapolation functions listed below. The user may alter the program, if necessary, to change these functions to suit his preference or experimental method.

1.  $\Delta\theta_1 = K_1 \sin 2\theta$ . This is equivalent to a linear extrapolation of a plot of  $d \underline{vs} \cos^2 \theta$  to  $\theta = 90^\circ$ . This function may be used to correct for eccentricity of the sample.
2.  $\Delta\theta_2 = K_2 \cos \theta$ . This is equivalent to a linear extrapolation of a plot of  $d \underline{vs} \cos \theta \cot \theta$  to  $\theta = 90^\circ$ .
3.  $\Delta\theta_3 = K_3 \cos \theta (1/2 + \sin \theta / 2\theta)$ . This is equivalent to a linear extrapolation of a plot of  $d \underline{vs} \frac{1}{2}(\cos^2 \theta / \sin \theta + \cos^2 \theta / \theta)$  to  $\theta = 90^\circ$ . This is the procedure suggested by Nelson and Riley to correct for absorption errors in the sample.

4.  $\Delta\theta_4 = K_4$ . This is equivalent to a linear extrapolation of a plot of  $d \sin \theta$  to  $\theta = 90^\circ$ . This function may be used to correct for a constant calibration error persisting at  $2\theta = 180^\circ$ . Prior calibration of the experimental apparatus is highly desirable. For example, a substance with accurately known lattice constants, such as Al, may be used for calibration.

It should be noted that these extrapolation functions have a similar mathematical form as  $\theta$  approaches  $90^\circ$ . The lack of orthogonality between these functions will result in a large correlation between the extrapolation coefficients. For this reason, the results obtained with the program when more than one extrapolation coefficient is varied should be cautiously interpreted.

#### WEIGHTING FACTORS

After correction for the systematic error  $\Delta\theta$ , it is presumed that only random errors in  $\theta$  will remain. Assuming the systematic errors are small, the random error in  $\Delta$  is given by the equation

$$\sigma(\Delta) = 8 \sigma(\theta) \sin \theta \cos \theta / \lambda^2, \text{ and therefore } w = \lambda^4 / 64 \sigma(\theta) \sin^2 \theta \cos^2 \theta.$$

The effect of these weighting factors on the final values of the parameters is not negligible, and it is therefore important that they be included.

#### INPUT DATA FORMATS

1. Title card, any alphanumerics in col. 1-70, col. 71-72 blank.
2. Initial lattice constants in angstroms and degrees: FORMAT (6F10.4);  
a, b, c, alpha, beta, and gamma.

3. Control card. Col. 1-10 have parameter selection information for the six lattice constants and the four extrapolation functions; a 1 in the corresponding column includes the parameter in the least squares treatment, otherwise put zero in the column.

Col. 20 has the lattice type code. Triclinic=1, monoclinic=2, orthorhombic=3, tetragonal=4, rhombohedral=5, hexagonal=6, or cubic=7.

Col. 29-30 has the maximum number of least squares cycles desired.

Col. 31-40, 41-50, and 51-60 have damping factors for the shifts in the magnitudes of the lattice constants, the angular constants, and the extrapolation coefficients. These are normally punched 1.0 unless convergence difficulties are encountered.

Col. 69-70 has the number of data cards to be considered.

Control card is FORMAT (10I1, 2I10, 3F10.2, I10).

4. Data cards, FORMAT (6F10.0). These contain  $h$ ,  $k$ ,  $l$ , two theta, estimated error in two theta, and the wave length.

Any number of sets of data may be processed without reloading the program. Each data set consists of items 1 to 4 above.

5. Stopper card, any alphanumerics in col. 1-70, 1 in col. 72.

```

START      ACCNTA0015 @C.WILLIAMS@
CCMPILE    RUN  FORTRAN LIST
C          LCR-2 LATTICE CONSTANT REFINEMENT D.WILLIAMS  APRIL 1964
C          INPUT IS AS FOLLOWS
C          TITLE CARD, FORMAT(14A5,I2)
C          INITIAL LATTICE CONSTANTS IN ANGSTROMS AND DEGREES, FORMAT(6F10.4)
C          CONTROL CARD, FORMAT(1CI1,2I10,3F10.2,I1C)
C          COL. 1-10 HAS PARAMETER SELECTION INFORMATION FOR THE SIX LATTICE
C          CONSTANTS AND FOUR EXTRAPOLATION FUNCTIONS, 1=SELECT, 0=CC NCT
C          SELECT
C          COL. 11-20 HAS LATTICE TYPE CODE
C          1=TRICLINIC
C          2=MONOCLINIC
C          3=ORTHORHOMBIC
C          4=TETRAGONAL
C          5=RHOMBOHEDRAL
C          6=HEXAGONAL
C          7=CUBIC
C          COL. 21-30 HAS NUMBER OF LEAST SQUARES CYCLES DESIRED
C          COL. 31-40 HAS DAMPING FACTOR FOR TRANSLATIONAL CONSTANT SHIFTS
C          COL. 41-50 HAS DAMPING FACTOR FOR ANGULAR CONSTANT SHIFTS
C          COL. 51-60 HAS DAMPING FACTOR FOR EXTRAPOLATION CONSTANTS
C          COL. 61-70 HAS NUMBER OF DATA CARDS TO BE CONSIDERED
C          DATA CARDS, FORMAT(6F10.C), H, K, L, TWO THETA, ESTIMATED ERROR IN
C          TWO THETA, WAVE LENGTH
C          STOPPER CARD, FORMAT(14A5,I2), +1 IN COL. 71-72
C          EXTRAPOLATION FUNCTIONS
C          1. DELTA THETA=K1*SINF(2.0*THETA)
C          2. DELTA THETA=K2*COSE(THETA)
C          3. NELSON-RILEY, DELTA THETA=K3*COSE(THETA)*(0.5+SINF(THETA)/(2.0*
C          THETA))
C          4. DELTA THETA=K4
C
          DIMENSION TITLE(14), ALAT(10), ISEL(10), XH(3,500), THETA(500),
          1  ERROR(500), ROW(10), DC(3,3,6), ALINC(6), ALDEL(6), DP(10),
          2  ANORM(65), AVEC(10), WAVE(500), CORR(10), SCEV(10), ALCUT(10)
          99 READ 101, (TITLE(I), I=1,14), KEY
          101 FORMAT (14A5,I2)
             IF (KEY) 98, 100, 98
          100 PRINT 102, (TITLE(I), I=1,14)
          102 FORMAT (1HC14A5)
             READ 103, (ALAT(I), I=1,6)
          103 .FORMAT (6F10.4)
             PRINT 104, (ALAT(I), I=1,6)
          104 FORMAT (1HC30H INITIAL LATTICE CONSTANTS
          1           /1H02HA=F8.4,4H B=F8.4,4H C=F8.4,
          2           8H ALPHA=F6.2,7H BETA=F6.2,8H GAMMA=F6.2)
             READ 105, (ISEL(I), I=1,10), LCCDE, NC, CPM, CPA, DPK, NDATA
          105 FORMAT (10I1,2I1C,3F10.2,I10)
             PRINT 106, (ISEL(I), I=1,10), NC, DPM, CPA, DPK, NDATA
          106 FORMAT (1HC 8HISEL(I)=10I1,5H NC=12,6H DPM=F4.2,
          1           6H CPA=F4.2,6H DPK=F4.2,8H NDATA=14)
             GO TO (126,132,123,129,128,130,121),LCCDE
          121 PRINT 115
          115 FORMAT (1HC15H CUBIC LATTICE)
             GO TO 135
          128 PRINT 116
          116 FORMAT (1HC25H RHOMBOHEDRAL LATTICE )
             GO TO 135
          129 PRINT 117

```

```

117 FORMAT (1HC20H TETRAGONAL LATTICE)
GO TO 135
130 PRINT 118
118 FORMAT (1HC20H HEXAGONAL LATTICE )
GO TO 135
123 PRINT 119
119 FORMAT (1HC25H ORTHORHOMBIC LATTICE )
GO TO 135
132 PRINT 120
120 FORMAT (1HC20H MONOCLINIC LATTICE)
GO TO 135
126 PRINT 134
134 FORMAT (1HC20H TRICLINIC LATTICE )
135 CONTINUE
READ 107, ((XH(J,I),J=1,3),THETA(I),ERRCR(I), WAVE(I), I=1, NDATA)
107 FORMAT (6F10.0)
DO 108 I=7, 10
108 ALAT(I)=0.0
NV=0
DO 111 I=1, 10
111 NV=NV+ISEL(I)
NP=NV*(NV+3)/2
NAN=NV+1
DO 113 I=1, 3
ALDEL(I)=0.005
ALAT(I+3)=ALAT(I+3)/57.295800
113 ALDEL(I+3)=0.5/57.295800
IC=1
C START REFINEMENT CYCLES
176 IF (IC-NC) 1176, 1176, 97
1176 DO 109 I=1, 10
109 AVEC(I)=0.0
DO 110 I=1, NP
110 ANGRM(I)=0.0
RSLM=0.0
C OBTAIN DC MATRIX, UPPER TRIANGULAR PORTION, DOUBLED OFF-DIAGONALS
DO 136 I=1, 3
DO 136 J=I, 3
CIJO=CIJ(I,J,ALAT)
DO 136 K=1, 6
CC(I,J,K)=0.0
DO 137 L=1, 6
137 ALINC(L)=ALAT(L)
IF (ISEL(K)) 136, 136, 206
206 ALINC(K)=ALAT(K)+ALDEL(K)
C SET SYMMETRY RELATED PARAMETERS
GO TO (201,201,201,202,203,202,205), LCCDE
202 ALINC(2)=ALINC(1)
GO TO 201
203 ALINC(5)=ALINC(4)
ALINC(6)=ALINC(4)
205 ALINC(3)=ALINC(1)
GO TO 202
201 CC(I,J,K)=(CIJ(I,J,ALINC)-CIJO)/ALDEL(K)
136 CONTINUE
C START LOOP THROUGH REFLECTION DATA
DO 141 M=1, NDATA
DELTA=0.0
DO 143 I=1, 3
DO 143 J=I, 3
143 DELTA=DELTA+CIJ(I,J,ALAT)*XH(I,M)*XH(J,M)
C CALCULATE DERIVATIVES OF DELTA WITH RESPECT TO LATTICE PARAMETERS

```

```

DO 142 K=1, 6
DP(K)=0.0
DO 142 I=1, 3
DO 142 J=I, 3
142 CP(K)=DP(K)+DC(I,J,K)*XH(I,M)*XH(J,M)
WTHE=THETA(M)/114.59160
WS2TH=SINF(2.0*WTHE)
WSTH=SINF(WTHE)
WCTH=COSE(WTHE)
WAV2=WAVE(M)*WAVE(M)
FLAMB=4.0/WAV2
WDELTH=ALAT(7)*WS2TH+ALAT(8)*WCTH+ALAT(9)*WCTH*
1 (0.5+WSTH/(2.0*WTHE))+ALAT(10)
DELTA=DELTA-FLAMB*(WSTH*WSTH+WDELTH*WS2TH)
CP(7)=-WS2TH*WS2TH*FLAMB
CP(8)=-WS2TH*WCTH*FLAMB
CP(9)=-WS2TH*WCTH*(0.5+WSTH/(2.0*WTHE))*FLAMB
CP(10)=-WS2TH*FLAMB
WATE=h*WAV2/(ERROR(M)*WS2TH)
DO 144 I=1, 10
144 CP(I)=CP(I)*WATE
DELTA=DELTA*WATE
C ADE TERMS TO MATRIX
RSM=RSM+DELTA*DELTA
KK=NV
K=C
DO 145 I=1, 10
IF (ISEL(I)) 145, 145, 146
146 DO 147 J=I, 10
IF (ISEL(J)) 147, 147, 148
148 KK=KK+1
ANORM(KK)=ANORM(KK)+DP(I)*DP(J)
147 CONTINUE
K=K+1
AVEC(K)=AVEC(K)-DELTA*DP(I)
145 CONTINUE
141 CONTINUE
CALL SYMLIN(ANORM,NV,AVEC,NG)
IF (NG) 152, 151, 152
152 PRINT 150
150 FORMAT (1H035H NORMAL EQUATIONS ARE SINGULAR )
GO TO 99
151 h=RSM/(FLCATF(NDATA)-FLOATF(NV))
DO 161 I=1, NV
161 ANORM(I)=ANORM(I)*h
DO 153 I=1, 10
IF (ISEL(I)) 154, 154, 153
154 CORR(I)=0.0
SDEV(I)=0.0
153 CONTINUE
K=0
DO 155 I=1, 3
IF (ISEL(I)) 155, 155, 156
156 K=K+1
ALAT(I)=ALAT(I)+ANORM(K)*DPM
CORR(I)=ANORM(K)*DPM
KK=K+NV+K-K*(K-1)/2
SDEV(I)=SQRTF(ANORM(KK))
155 ALCUT(I)=ALAT(I)
DO 157 I=4, 6
IF (ISEL(I)) 157, 157, 158
158 K=K+1

```

```

CORR(I)=ANCRM(K)*DPA*57.295800
ALAT(I)=ALAT(I)+ANCRM(K)*DPA
KK=K*(NV+1)-K*(K-1)/2
SDEV(I)=57.295800*SQRTF(ANORM(KK))
157 ALCUT(I)=ALAT(I)*57.295800
DO 159 I=7, 10
IF (ISEL(I)) 159, 159, 160
160 K=K+1
CORR(I)=ANCRM(K)*DPK*57.295800
ALAT(I)=ALAT(I)+ANCRM(K)*DPK
KK=K*(NV+1)-K*(K-1)/2
SDEV(I)=57.295800*SQRTF(ANORM(KK))
159 ALCUT(I)=ALAT(I)*57.295800
C SET SYMMETRY RELATED PARAMETERS
GO TO (1201,1201,1201,1201,1202,1203,1202,1205), LCCDE
1202 ALAT(2)=ALAT(1)
ALCUT(2)=ALCUT(1)
GO TO 1201
1203 ALAT(5)=ALAT(4)
ALCUT(5)=ALCUT(4)
ALAT(6)=ALAT(4)
ALCUT(6)=ALCUT(4)
1205 ALAT(3)=ALAT(1)
ALCUT(3)=ALCUT(1)
GO TO 1202
1201 CONTINUE
C OUTPUT RESULTS FOR CYCLE IN ANGSTROMS AND DEGREES
PRINT 166, IC
166 FORMAT (1H05HCYCLEI3, 32H A B C
1 40H ALPHA BETA GAMMA K(1)
2 39H K(2) K(3) K(4) RSUM/(NDATA-NV))
PRINT 170, (CCRR(I), I=1, 10)
170 FORMAT (1HC10HCHANGE 3F10.4,3F10.3,4F08.4)
PRINT 171, (ALCUT(I), I=1, 10), h
171 FORMAT (1HC10HNEW VALUE 3F10.4,3F10.3,4F08.4,E15.5)
PRINT 172, (SDEV(I), I=1, 10)
172 FORMAT (1H010HESC 3F10.4,3F10.3,4F08.4)
C TEST FOR END OF REFINEMENT
K=0
DO 501 I=1, 10
IF (ISEL(I)) 501, 501, 502
502 K=K+1
KK=K*(NV+1)-K*(K-1)/2
IF ((ABSF(ANORM(K))/SQRTF(ANORM(KK)))-C.1) 501, 503, 503
501 CONTINUE
GO TO 97
503 IC=IC+1
GO TO 176
C TERMINATE REFINEMENT
97 IF (NC) 14C1, 14C1, 1197
1197 PRINT 301, (ANCRM(I), I=NAN, NP)
301 FORMAT(1H0 50H VARIANCE-COVARIANCE MATRIX IN ANGSTROMS AND RADI
1 SHANS /(1H07E15.6))
DO 302 K=1, NV
KK=K*(NV+1)-K*(K-1)/2
302 SDEV(K)=1.C/SQRTF(ANORM(KK))
PRINT 303
303 FORMAT(1H0 20H CORRELATION MATRIX)
KK=NAN
DO 304 I=1, NV
DO 305 J=1, NV
305 ROW(J)=0.0

```



```

      DO 306 J=1, NV
      ROW(J)=ANORM(KK)*SDEV(I)*SDEV(J)
306  KK=KK+1
      PRINT 307, (ROW(J), J=1, NV)
307  FORMAT(1H0 10F10.4)
304  CONTINUE
C     OUTPUT REFLECTION DATA
1401 PRINT 401
401  FORMAT(1H0 45H
1     1          45H      ESTIMATED      CALCULATED      OBSERVED
2     2          45H      H                K                CALCULATED /1H
3     3          45H      ERROR           TWO THETA      TWO THETA
4     4          15H      LAMBDA        )          DRIFT
      DO 402 M=1, NDATA
      DELTA=0.0
      DO 403 I=1, 3
      DO 403 J=1, 3
403  DELTA=DELTA+CIJ(I,J,ALAT)*XH(I,M)*XH(J,M)
      WTPE=THETA(M)/114.59160
      WS2TH=SINF(2.0*WTPE)
      WSTH=SINF(WTPE)
      WCTH=CCSF(WTPE)
      WWAV2=WAVE(M)*WAVE(M)
      WDELTH=ALAT(7)*WS2TH+ALAT(8)*WCTH+ALAT(9)*WCTH*
1     (0.5+WSTH/(2.0*WTPE))+ALAT(10)
      CALC=114.59160*ASINF(SQRTF(WWAV2*DELTA/4.0-WDELTH*WS2TH))
      DRIFT=114.59160*WDELTH
      PRINT 404, (XH(I,M), I=1,3), THETA(M), ERROR(M), CALC, DRIFT, WAVE(M)
404  FORMAT(1H0 3F10.0, F12.2, 3F15.2, F15.6)
402  CONTINUE
      GO TO 99
98  STOP 87
      END

```

```

      END          ACCNTA0015 @D.WILLIAMS@          PCGC
      START       ACCNTA0015 @D.WILLIAMS@          PUGO
      CCMPILE     RUN FORTRAN LIST                 PCGO
C     FUNCTION CIJ(I,J,ALAT), J IS EQUAL OR GREATER THAN I
      FUNCTION CIJ(I,J,ALAT)
      DIMENSION ALAT(6)
      COSA=COSF(ALAT(4))
      COSB=COSF(ALAT(5))
      COSC=COSF(ALAT(6))
      DET=1.0/(1.0-CCSA*COSA-COSB*COSB-COSC*COSC+2.0*COSA*COSB*CCSC)
      GO TO (7,8,6), I
7     GO TO (1,2,3), J
8     GO TO (9,4,5), J
1     CIJ=DET*(1.0-CCSA*COSA)/(ALAT(1)*ALAT(1))
      GO TO 9
2     CIJ=DET*(COSA*COSB-COSC)*2.0/(ALAT(1)*ALAT(2))
      GO TO 9
3     CIJ=DET*(COSA*COSC-COSB)*2.0/(ALAT(1)*ALAT(3))
      GO TO 9
4     CIJ=DET*(1.0-CCSB*COSB)/(ALAT(2)*ALAT(2))
      GO TO 9
5     CIJ=DET*(COSB*COSC-COSA)*2.0/(ALAT(2)*ALAT(3))
      GO TO 9
6     CIJ=DET*(1.0-CCSC*COSC)/(ALAT(3)*ALAT(3))
9     RETURN
      END

```

```

END      ACCNTA0015  @D.WILLIAMS@          PGGC
START    ACCNTA0015  @D.WILLIAMS@          PGGC
CCMPILE  RUN  FORTRAN LIST                  PGGC
C        SYMMETRIC LINEAR EQUATION SOLVER USING SYMINV
SUBROUTINE SYMLIN(AM, NV, VEC, ISING)
DIMENSION AM(9869), VEC(139)
CALL SYMINV (AM, NV, ISING)
IF (ISING) 99, 2, 99
2  DO 3 I=1, NV
   AM(I)=0.0
   DO 3 J=1, NV
    IF (J-I) 4, 5, 5
5   IJMUL=I*NV+J-(I*(I-1))/2
   GO TO 6
4   IJMUL=J*NV+I-(J*(J-1))/2
6   AM(I)=AM(I)+AM(IJMUL)*VEC(J)
3  CONTINUE
99 RETURN
END

END      ACCNTA0015  @D.WILLIAMS@          PGGC
START    ACCNTA0015  @D.WILLIAMS@          PGGC
CCMPILE  RUN  FORTRAN LIST                  PGGC
C        INVERT POSITIVE DEFINITE SYMMETRIC MATRIX OF ORDER NV  D.WILLIAMS
C        MATRIX IS STORED AS FOLLOWS
C        N STORAGE CELLS TO BE USED AS WORK CELLS BY INVERTER, FOLLOWED BY
C        MATRIX ELEMENTS BY ROWS, GIVING ONLY UPPER TRIANGULAR PART
C        A TOTAL OF NV*(NV+3)/2 STORAGE CELLS IS REQUIRED
C        LOCATION AM(NV+1) IS STARTING ADDRESS OF MATRIX
C        NG=0 IS NORMAL RETURN, NG=1 IS RETURN FOR SINGULAR MATRIX
C        INVERSE MATRIX IS RETURNED TO SAME LOCATION I.E. AM(NV+1)
C        METHOD-BUSING AND LEVY, COMM. ACM, AUGUST 1962
SUBROUTINE SYMINV(AM, NV, NG)
DIMENSION AM(9869)
C        DIAGONALIZE
DO 24 I=1, NV
IO=I*NV-(I*(I-1))/2
II=IO+I
LI=(I-1)*NV+I-((I-1)*(I-2))/2
AM(LI)=FLOATF(I)
BIGAJJ=0.0
DO 2 J=I, NV
JJ=J*NV+J-(J*(J-1))/2
IF (ABSF(AM(JJ))-BIGAJJ) 2, 2, 3
3  BIGAJJ=ABSF(AM(JJ))
AM(LI)=FLOATF(J)
2  CONTINUE
CALL AINTCH(AM,NV,I,IO,II,LI)
C        RESUME DIAGONALIZE
20 IF(AM(II)) 22, 22, 21
22 NG=1
GO TO 62
21 AM(II)=1.0/AM(II)
IQQ=(I-1)*NV-((I-1)*(I-2))/2
II=I+1
IF(II-NV) 28, 28, 29
28 DO 23 K=II, NV
IK=IO+K
IKC=IQQ+K

```

```

23 AM(IKQ)=-AM(II)*AM(IK)
   DO 24 J=I1, NV
     JO=J*NV-(J*(J-1))/2
     IJ=IO+J
     DO 24 K=J, NV
       JK=JO+K
       IKC=IQC+K
24 AM(JK)=AM(JK)+AM(IKQ)*AM(IJ)
C   RESTORE
29 I=NV-1
55 IO=I*NV-(I*(I-1))/2
   II=IO+I
   LI=(I-1)*NV+I-((I-1)*(I-2))/2
   I1=I+1
   DO 50 J=I1, NV
     IJ=IO+J
50 AM(IJ)=0.0
   IQC=(I-1)*NV-((I-1)*(I-2))/2
   DO 52 J=I1, NV
     JO=J*NV-(J*(J-1))/2
     IJ=IO+J
     IJQ=IQC+J
     DO 30 K=J, NV
       IK=IO+K
       JK=JO+K
       AM(IK)=AM(IK)+AM(IJQ)*AM(JK)
       IF (J-K) 53, 30, 53
53 IKC=IQC+K
   AM(IJ)=AM(IJ)+AM(IKQ)*AM(JK)
30 CONTINUE
52 AM(II)=AM(II)+AM(IJQ)*AM(IJ)
C   INTERCHANGE
   CALL AINTCH(AM,NV,I,IO,II,LI)
   I=I-1
   IF (I) 61, 61, 55
61 NG=0
62 RETURN
   END

END      ACCNTA0015  @D.WILLIAMS@          POGO
START   ACCNTA0015  @D.WILLIAMS@          POGO
CCMPLE  RUN  FORTRAN LIST                  POGO
C   INTERCHANGE SUBROUTINE FOR SYMINV      D.WILLIAMS
   SUBROUTINE AINTCH(AM,NV,I,IO,II,LI).
   DIMENSION AM(9869)
10 LI=XINTF(AM(LI))
   IF(LI-I) 15, 16, 15
15 ILI=IO+LI
   LIO=LI*NV-(LI*(LI-1))/2
   DO 11 J=I, NV
     IJ=IO+J
     WORK=AM(IJ)
     IF(LI-J) 13, 13, 12
12 JLI=J*NV+LI-(J*(J-1))/2
   AM(IJ)=AM(JLI)
   AM(JLI)=WORK
   GO TO 11
13 LIJ=LIO+J
   AM(IJ)=AM(LIJ)
   AM(LIJ)=WORK
11 CONTINUE

```

```
WORK=AM(II)  
AM(II)=AM(ILI)  
AM(ILI)=WORK  
16 RETURN  
ENC
```

END ACCNTAOC15 @D.WILLIAMS@

PUGO