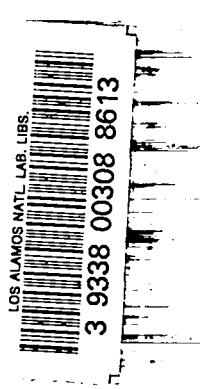
Photocomposition by Chris West

LA-9710

UC-34a
Issued: May 1983

A New Program for the Least Squares Calculation of Atomic Energy Levels

B. A. Palmer
R. Engleman, Jr.



Los Alamos National Laboratory
Los Alamos, New Mexico 87545

A NEW PROGRAM FOR THE LEAST SQUARES CALCULATION OF ATOMIC ENERGY LEVELS

by

B. A. Palmer and R. Engleman, Jr.

ABSTRACT

Calculation of energy levels from measured transitions is important in atomic spectroscopic research. We present an improved least squares program for such a calculation, a program that transfers easily to any modern computer using FORTRAN and that runs on minicomputers as well as large modern computers. We show in detail how to use this program.

I. INTRODUCTION

A traditional problem in classical optical spectroscopy is calculating best estimates of atomic energy level values from a given set of experimental assigned transitions. This problem can be expressed as

$$T_{ij} = A_i - B_j \ , \qquad (1)$$

where T_{ij} is the set of experimental transitions and A_i and B_j are sets of energy levels of opposite (odd and even) parity. In this formulation of the Rydberg-Ritz combination principle, we make T_{ij} either positive or negative, depending on the magnitude of A_i and B_j. However, all measurements of T_{ij} are, by definition, positive. The only inaccuracies in Eq. (1) arise from errors in the measurements of T_{ij}.

A 1970 report[1] gave computer programs, written for a CDC 6600 computer,* for two methods of solving this problem: an iterative and a least squares method. In addition, Radziemski et al.[2] published the least squares method in a journal article. Subsequently, we published corrections to their program.[3] Because the least squares method has been widely used, we have developed an improved version adapted for modern minicomputers that will be presented in this report with an abbreviated analysis of the solutions set forth in Refs. 1 and 2.

When the Radziemski, Fisher, and Steinhaus report[1] was written, the 60-bit computer word was common in the large computers used to calculate energy levels. Now the modern minicomputers can calculate these levels adequately. However, they commonly use 16- and 32-bit words. The original FORTRAN programs were written with much packing and unpacking to conserve memory space, but that is also tied to a 60-bit word. Unfortunately, changing to different word lengths is not simple. This new version is written in FORTRAN but does not do any packing or unpacking. It was originally written for a PDP-11/34* (16 bit), but later storage requirements (caused by the complex spectra of uranium and thorium) necessitated transferring the program to a VAX 11/780.* This transfer was readily accomplished by simply changing the appropriate DIMENSION statements to accommodate the larger data blocks. Our version also has some new features that enhance its versatility and usefulness.

*Control Data Corporation, 8100 34th Avenue South, Minneapolis, MN 55440.

*Digital Equipment Corporation, Maynard, MA 01754.

II. LEAST SQUARES SOLUTION

To calculate improved energy levels, we took a least squares approach. Let A_i be the i^{th} level of a given parity and B_j be the j^{th} level of the opposite parity. Let T_{ij} be the transition connecting A_i to B_j and ω_{ij} be the weight of the measurement T_{ij}. By minimizing the sum

$$R = \sum_i \sum_j (A_i - B_j - T_{ij})^2 \, \omega_{ij} \tag{2}$$

with respect to the level values A_i and B_j, we find the best estimate of these levels. Differentiating Eq. (2) with respect to A_i and B_j and setting these derivatives to zero yield the following equations:

$$\frac{\partial R}{\partial A_i} = 2 \sum_j (A_i - B_j - T_{ij}) \, \omega_{ij} = 0 \tag{3}$$

and

$$\frac{\partial R}{\partial B_j} = -2 \sum_i (A_i - B_j - T_{ij}) \, \omega_{ij} = 0 \ , \tag{4}$$

assuming that all B_j and A_i are independent. This treatment leads to a set of N + M linearly independent equations with N + M variables: the N levels of A_i and M levels of B_j. To obtain a unique solution, one level must be fixed, which is accomplished by setting B_j to zero. (In standard spectroscopic practice, the lowest energy level is assigned the value of zero.) This matrix is large, difficult to hold in the computer memory, and prone to large errors because of roundoff. To overcome this difficulty, we solve only for the B_j levels and then calculate the A_i levels from them. From Eq. (4) we obtain

$$\sum_i \omega_{ij} \, B_j = \sum_i \omega_{ij} \, A_i - \sum_i T_{ij} \, \omega_{ij} \ , \tag{5}$$

and from Eq. (3) we find

$$A_i = \frac{1}{\sum_k \omega_{ik}} \left(\sum_k T_{ik} \, \omega_{ik} + \sum_k B_k \, \omega_{ik} \right) \ . \tag{6}$$

Combining these two equations, we obtain

$$\sum_i \omega_{ij} \, B_j = \sum_i \left[\frac{\omega_{ij}}{\sum_k \omega_{ik}} \left(\sum_k T_{ik} \, \omega_{ik} + \sum_k B_k \, \omega_{ik} \right) \right] - \sum_i T_{ij} \, \omega_{ij} \ . \tag{7}$$

Letting

$$Q_j = \sum_i \omega_{ij} \left(\frac{1}{\sum_k \omega_{ik}} \sum_k T_{ik} \, \omega_{ik} \right) - \sum_i T_{ij} \, \omega_{ij} \ , \tag{8}$$

Eq. (7) becomes

$$\sum_i \omega_{ij} \, B_j - \sum_i \omega_{ij} \left(\frac{\sum_k \omega_{ik} \, B_k}{\sum_k \omega_{ik}} \right) = Q_j \ . \tag{9}$$

Equation (9) forms M − 1 linearly independent equations for B_j (because B_1 was set to zero). Rewriting Eq. (9) into matrix form, we obtain

$$\bar{C} \cdot \bar{B} = \bar{Q} \ , \tag{10}$$

where

$$C_{ij} = \sum_k \omega_{ki} \left(1 - \frac{\omega_{ki}}{\sum_\ell \omega_{k\ell}} \right) \ ; \qquad i = j \tag{11}$$

and

$$C_{ij} = - \sum_k \frac{\omega_{ki} \, \omega_{kj}}{\sum_\ell \omega_{k\ell}} \ ; \qquad i \neq j \ . \tag{12}$$

$$\bar{B} = \bar{C}^{-1} \cdot \bar{Q} \tag{13}$$

gives the solution for the level values B_j. The advantage of this formulation is that calculating improved energy level values is very fast and the uncertainties in the levels can be determined statistically from the $\bar{C}^{-1}$ matrix. More information about calculating uncertainties is in Ref. 1.

III. DESCRIPTION OF CLEVEL

We divided energy level calculations into three separate programs, which are listed in Appendix A. This separation was necessary to reduce the amount of memory needed. In the old program, equivalence in data storage (EQUIVALENCE statements) conserved space, but the use of virtual memory in our smaller computer prohibited such a technique. The first program reads in the control parameters and the data file; it sorts the data according to levels and performs several checks on the data, eliminating duplicate data and excluding the levels selected by the user from the calculation. It also computes the Q matrix as given by Eq. (8). Data to the second program are written out to two files.

The second program reads in the data from the first program and generates the C matrix. Only half of the C matrix is kept because $C_{ij} = C_{ji}$, which allows the C matrix to be stored efficiently and thus conserves memory. Next, this program computes the inverse of the C matrix and checks the inversion if a check is desired. The inversion check is performed by computing the identity matrix; the program multiplies the stored C matrix and the inverse. The program then looks for the largest deviation from the identity matrix and computes the root mean square (rms) of all deviations. If either the rms or the largest deviation is too large ($>10^{-5}$), one or more levels must be removed because they cannot be determined. This program next computes the correction to the energy levels and the variances associated with those levels and writes this information out as two files.

The third program in CLEVEL reads in the corrections to the energy levels; computes the final level values; and if we request it, prints out all the transitions, sorted by the A_i levels, along with variance and actual deviation from the calculated transitions. The third program then prints out the energy levels as well as writing them to a file. Finally, it prints the weight statistics that help us decide whether we used the appropriate weighting scheme.

These three programs should be run sequentially. If delaying one program is necessary, the intermediate files must be saved. The first two programs generate a small amount of print-out that is useful for diagnostics, but the third program generates most of the print-out. The programs are set to print to a printer and to type some information out to a terminal controlling the process. The programs have also been run in a batch environment with no terminal output. The intermediate data files may be deleted at the end. Table I gives the data files written by the program.

IV. PROGRAM INPUT

Appendix B contains examples of the two input files to the CLEVEL programs. (Data are from the "Atlas of the Thorium Spectrum."[4]) The first data file consists of parameters, and the second contains the transitions used to calculate energy levels. The parameter file contains a minimum of three lines. The first line consists of eight parameters, five logical inputs consisting of T or F, and three numeric constants. The first parameter IREV controls selection of the B_j level set, the level set used for the matrix inversion. If the parameter is T, the first level in the data file is used for the B_j levels. If it is F, the second level is used. The second parameter IABR controls which set of levels has the assumed zero value. If it is T, the A_i levels are assumed to have the zero level. The time spent in the matrix inversion is proportional to M^2 (M = number of B levels). The storage requirements depend on the magnitude of M. To decrease the time necessary for a given problem and the amount of memory needed, choose the smallest side of the array to be the B levels. The problem encountered here is that these levels may not have the ground state as a member and, as a result, B_1 will not be zero. Table II gives suggested values for these two parameters based on the data being used.

The third parameter tells the program what type of problem is being run, either an isotope shift calculation or a level calculation. If the parameter is T, an isotope shift calculation is run. The fourth parameter determines whether a matrix inversion check should be performed. A T instructs the program to do an inversion check; this check determines the largest error and the rms error in the identity matrix that is formed by multiplying the inverted matrix by the original matrix. The fifth logical parameter determines the amount of printed information. A T requests a full output listing, including all the transitions. If this parameter is set to F, only the level values are printed.

The remainder of the line consists of three parameters: IML, WTUNC, and SFL. IML determines the minimum number of transitions per level that are necessary to include the level in the matrix inversion. If IML = 0, all levels will be included; if IML = 1, any level with only one transition is excluded from the level fit. If IML is not

TABLE I. Files Generated by CLEVEL

File	Type	Write	Read	Description
11	Binary	1	2	Sorted list of levels and wave numbers, 0 matrix
12	Binary	1	2	Parameters to second program, levels
13	Binary	2	3	Level corrections and transitions
14	Binary	2	3	Parameter file, B levels
CLLEV.DAT	Text	3		Energy levels, A and B
15	Binary	3		Transitions, weights, and deviations
16	Text	3		Transitions marked with * or **

TABLE II. Suggested Level Control Parameters

Minimum Number of Levels	Zero Level	IREV	IABR
Col. 2	Col. 2	F	F
Col. 1	Col. 1	T	F
Col. 1	Col. 2	T	T
Col. 2	Col. 1	F	T

zero and levels must be discarded, eliminating those levels will require extra time because this is an iterative process. The data file of transitions is read in repeatedly until no more levels are excluded. Because any level determined to have only one transition adds very little to the determination of the rest of the levels, IML can be used to eliminate them.

WTUNC is the uncertainty associated with a weight of one. Thus, if WTUNC is set to 0.001, any transition with an uncertainty equal to 0.001 would have a weight of one. Those transitions with an uncertainty greater than WTUNC would be weighted less than one. If WTUNC is less than zero, all transitions will be weighted equally except those to be excluded.

The final parameter on this line is SFL, which controls the determination of the integer part of the wave number. The level identifications read in from the data file are multiplied by this value. Thus, if SFL is 0.1, the least significant digit is dropped from the level identification before calculation. If SFL is set to zero, the full wave number is used in the calculation. We do not recommend setting SFL to zero because the larger numbers can lead to larger roundoff errors and the possibility of overflow in the matrix inversion.

The next line in the parameter file is an 80-character description of the run; the first few characters are reserved for the run number or other identification. The next line contains the name of the data file that will be read. Subsequent lines are optional and are used to exclude levels from the energy level calculations. These lines are input as level identifications, and those levels with exactly the same identification will be excluded.

The data file contains the wave number of the transition, the row level (A_I) and column level (B_J) classification identifications, the uncertainty in the wave-number measurement, and (optionally) the signed isotope shift if an isotope shift run is desired. The weight for a transition is determined by the equation

$$\text{weight} = \left(\frac{\text{WTUNC}}{\text{uncertainty}}\right)^2 . \tag{14}$$

If the uncertainty is greater than 9.0, the weight is set to zero.

V. DATA OUTPUT

Although calculated energy levels are the major output from CLEVEL, it also provides diagnostic output to determine how well the problem was solved. Appendix C contains the listings from the run whose input is given in Appendix B.

The terminal output consists of the parameter inputs and an indication of the progress of the program. In the first program, the number of transitions retained is printed after levels requested by the user and levels with too few transitions (IML selected) are excluded. The second program sends only the matrix inversion check to the terminal. The third program prints only internal diagnostic information.

Output to the printer from the first program consists of the input parameters, the excluded lines, the total number of A_i and B_j levels, and a table. The table lists the identification label for each B_j level, the sum of the weights of all transitions to the B_j levels, Q_j as given in Eq. (7), and the number of transitions to the B_j levels. If the sum of the weights for any one level is zero, the matrix cannot be inverted correctly, and that level should be excluded.

The output from the second program consists of only the matrix inversion check if one was requested. The output from the third program starts with the transitions if a full print-out was requested. For each A_i level, all connecting B_j levels are given in the second column. The A_i levels are sorted in increasing order. The third column gives the weight used for the transition connecting the two levels. The fourth column contains the observed transition, and the next column shows the difference between the calculated and the observed transition. The final column gives the square root of the variance or the uncertainty of the calculated transition. After the column labeled C-O, flags indicate lines with excessive deviations from the predicted uncertainty from the data file. Those transitions with twice the deviation from the predicted uncertainty are marked with a single asterisk (*); transitions with three times the deviation are marked with a double asterisk (**). Lines with a weight close to zero are marked with two dashes (--) for easy location.

The A_i levels are listed next. The first column is the level identification; the second column is the calculated level; and the third column is the uncertainty of that level. The fourth column measures the relative accuracy of the level compared with other levels. The fifth column is the correction applied to the level identification to obtain the calculated level, and the sixth and final column lists the total number of the transitions to this level.

The B_j levels are given next in the same format. After the B_j levels, a line gives the accuracy of the level determinations, the number of B_j and A_i levels, and the number of transitions. SIGMA gives the standard deviation or measure of the least squares fit.

Weight statistics are the final output from CLEVEL. For each class of input uncertainty, the program enters the weight used. The rms for all the transitions in this weight class and the number of transitions are printed in the next two columns. This table is useful in determining the accuracy of the uncertainties assigned to each transition.

The output files consist of CLLEV.DAT, which is similar to the level listing; A_i levels are separated from the B_j levels by a -1 for the identification field, and the file is terminated by a -2 in the same field. Two other files are written and may be useful: file 16 has lines marked with single or double asterisks, and file 15 has all the transistions, their input uncertainty, and the deviation from the calculated value.

VI. USE OF OTHER COMPUTERS

The programs are written in FORTRAN and should move easily to most modern computers. The input and output may be different, and changes may have to be made in the OPEN and CLOSE statements. The other nonstandard statement is VIRTUAL. Variables dimensioned by these statements are kept in virtual memory, either on a disk or in memory that is not directly addressed. For computers that do not support VIRTUAL, change these statements to DIMENSION statements.

Changing dimension sizes for different problems is relatively simple. We have attempted to group the arrays needing a common size. A few statements in CLEV1 check to see if the user has exceeded the dimension size, and those statements must be changed. The C matrix is a little more complicated to change because it has two different dimensions; the program contains instructions on how to determine the size of the C matrix based on the number of B_j levels.

Most real variables are 32-bit floating-point numbers; variables that must have greater precision than that are listed in the REAL*8 statements and are 64-bit floating-point numbers. Because of the scaling used, only the wave numbers and the final level values must be double precision. We have detected no loss in accuracy using 32-bit floating-point numbers in any of the test cases we have tried (comparing the results with 60-bit computer runs).

REFERENCES

1. L. J. Radziemski, Jr., K. J. Fisher, and D. W. Steinhaus, "Calculation of Atomic-Energy-Level Values," Los Alamos Scientific Laboratory report LA-4402 (June 1970).

2. L. J. Radziemski, Jr., K. J. Fisher, D. W. Steinhaus, and A. S. Goldman, "Calculation of Atomic Energy Level Values," Comput. Phys. Commun. 3, 9-23 (1972).

3. R. Engleman, Jr., and B. A. Palmer, "Precision Isotope Shifts for the Heavy Elements. I. Neutral Uranium in the Visible and Near Infrared," J. Opt. Soc. Am. 70, 308-317 (1980).

4. B. A. Palmer and R. Engleman, Jr., "Atlas of the Thorium Spectrum," Los Alamos National Laboratory report LA-9615 (February 1983).

APPENDIX A
CLEVEL PROGRAMS

Los Alamos Identification No. LP-1530

CLEV1

```
      PROGRAM CLEV1
C     ------------------------------------------------------------
C     CALCULATES THE LEVEL VALUES GIVEN A SET OF TRANSITIONS
C     CALCULATES THE ISOTOPE SHIFTS GIVEN A SET OF IS TRANSITIONS
C     THIS IS THE FIRST PROGRAM IN A SERIES OF THREE PROGRAMS
C     ------------------------------------------------------------
C     DIMENSION STATEMENTS FOR THE COLUMN (B LEVELS)
      REAL*4 LEVC
      VIRTUAL YV(280),WV(280),B(280)
      DIMENSION JV(280),LEVC(280),QTAB(280),WTJ(280)
C     DIMENSION STATEMENTS FOR THE TRANSISTIONS
      REAL*8 WM,WN
      VIRTUAL WT(6300),WN(6300),RIT(2000)
      REAL*4 LR,LC
      VIRTUAL LR(6300),LC(6300)
      DIMENSION IT(6300),IDT(12),IRUN(40),IFIL(8),NTB(280)
      LOGICAL PRNT,ISOTOP,INVCK,LDEL,IABR
C     ------------------------------------------------------------
C     INPUT CONTROL FILE
C     ------------------------------------------------------------
C     IREV,IABR,ISOTOP,INVCK,PRNT,IML,WTUNC,SFL
C     RUN DESCRIPTION
C     FILE
C     LEVELS TO BE EXCLUDED (OPTIONAL)
C    --- FORMAT STATEMENTS ---
C    (5L1,I5,F10.0,F5.0/40A2,8A2)
C    (F10.0)
C    --- PARAMETER DESCRIPTION
C     IREV=T USE FIRST LEVEL FOR B AND SECOND FOR A
C     IABR=T A LEVELS HAVE GROUND LEVEL INCLUDED
C     ISOTOP=T IF ISOTOPE SHIFT RUN
C     INVCK=T  IF INVERSE CHECK IS DESIRED
C     PRNT=T   IF ALL DATA IS PRINTED
C     IML= NUMBER OF LINES NEEDED PER LEVEL - 1
C     WTUNC=UNCERTAINTY ASSOCIATED WITH WEIGHT OF ONE
C     IF WTUNC<0 THEN USE UNITS WEIGHTS EXCEPT FOR THOSE > 9.0
C     SFL=SCALE FACTOR FOR LEVEL ID'S
C      MULTIPLIES LEVEL ID'S TO GET VALUES
C     NAME=80 CHARACTER DESCRIPTION FOR THIS RUN
C     FILE=THE FILE NAME TO BE USED FOR WAVENUMBER INPUT
C
C     THESE ARE THEN FOLLOWED BY A SERIES OF CARDS CONTAINING
C     THE LEVELS TO EXCLUDE IN THE RUN (F10.0)
C     THESE ARE OPTIONAL
C
```

```
C     ------------------------------------------------------------
C      THE WAVENUMBER FILE
C     ------------------------------------------------------------
C      COL 1-13 (F13.4) WAVE NUMBER
C      COL 14-23 (F10.0) ROW LEVEL CLASSIFICATION NAME (A)
C      COL 24-33 (F10.0) COLUMN LEVEL CLASSIFICATION NAME  (B)
C      COL 33-40 (F8.5) UNCERTAINTY OF THE WAVENUMBER
C      COL 34-42 (F10.5) SIGNED ISOTOPE SHIFT IF AN ISOTOPE RUN
C
C     ------------------------------------------------------------
C      FILES USED:
C     1-DATA INPUT
C     11-SORTED LIST OF LEVELS AND WAVENUMBERS (FTN11.DAT)
C     12-PARAMETERS TO SECOND PROGRAM (FTN12.DAT)
C
C     ------------------------------------------------------------
       RL(X)=AINT(SFL*X)
       CALL DATE(IDT)
       CALL TIME(IDT(6))
       TYPE 2000
2000   FORMAT(' INPUT FILE-',$)
       ACCEPT 2001,IFIL
2001   FORMAT(8A2)
       OPEN(UNIT=1,NAME=IFIL,TYPE='OLD')
       TSTAR=SECNDS(0.0)
       READ(1,2002) IREV,IABR,ISOTOP,INVCK,PRNT,IML,SFL,WTUNC,IRUN,IFIL
2002   FORMAT(5L1,I5,2F10.0/40A2/8A2)
       PRINT 2003,IREV,IABR,ISOTOP,INVCK,PRNT,IML,SFL,(IDT(I),I=1,5),
      1 (IDT(I),I=6,10),IFIL,IRUN
       TYPE 2003,IREV,IABR,ISOTOP,INVCK,PRNT,IML,SFL,(IDT(I),I=1,5),
      1 (IDT(I),I=6,10),IFIL,IRUN
2003   FORMAT(2X,'IREV=',L1,'  IABR=',L1,'  ISOTOP=',L1,
      1 '  INVCK=',L1,'  PRNT=',L1,
      2 '  IML=',I5/'  LEVEL SCALE FACTOR=',F10.3,/'  DATE-',5A2,
      3 '  TIME-',5A2,' FILE-',8A2/'  RUN-',40A2/)
       IF(WTUNC.EQ.0.) WTUNC=1.
       IF(ISOTOP) PRINT 2004, WTUNC
2004   FORMAT(' ISOTOPE SHIFT RUN WTUNC= ',F9.5)
       IF(.NOT.ISOTOP) PRINT 2005,WTUNC
2005   FORMAT(' WAVE NUMBER LEVEL RUN  WTUNC= ',F9.5)
C      READ IN THE LEVELS TO BE DELETED FROM RUN
       ID=0
100    ID=ID+1
       READ (1,2100,END=110) RIT(ID)
2100   FORMAT(F10.0)
       IF(RIT(ID).NE.0) GO TO 100
110    ID=ID-1
       CLOSE(UNIT=1)
C      READ IN THE WAVENUMBERS
120    OPEN(UNIT=1,NAME=IFIL,TYPE='OLD')
       IX=1
```

```
C     READ IN DATA CARDS
130   IF(.NOT.IREV) READ(1,2130,END=160) WM,LR(IX),LC(IX),WT(IX),SFT
      IF(IREV) READ(1,2130,END=160) WM,LC(IX),LR(IX),WT(IX),SFT
2130  FORMAT(F13.4,2F10.0,F8.5,F10.2)
      IF(WM.EQ.0.0) GO TO 160
      IF(WT(IX).EQ.0.0) WT(IX)=1.0
      IF(ID.EQ.0) GO TO 150
      DO 140 J=1,ID
      IF(LR(IX).NE.RIT(J).AND.LC(IX).NE.RIT(J)) GO TO 140
      PRINT 2131,WM,LR(IX),LC(IX),SFT
2131  FORMAT(F15.4,2F10.0,F12.5,' EXCLUDED LINE')
      GO TO 130
140   CONTINUE
150   CONTINUE
      WN(IX)=WM
      IF(ISOTOP) WN(IX)=SFT
      IX=IX+1
      IF(IX.GT.6300) GO TO 160
      GO TO 130
160   IX=IX-1
      CLOSE(UNIT=1)
C     FINISHED LOADING TRANSITIONS, START SORTING
      TEND=SECNDS(TSTAR)
      TMIN=AINT(TEND/60.0)
      TEND=TEND-TMIN*60.
      PRINT 2160,IX,TMIN,TEND
      TYPE 2160,IX,TMIN,TEND
2160  FORMAT(' FINISHED LOADING ',I10,'TRANSITIONS, TIME=',
     1 F6.0,':',F4.0)
C     GENERATE B(J) LEVEL LIST
2161  FORMAT(1H1(1H ,151I7))
      CALL SORT2(IX,LC,LR,IT)
      LEVC(1)=LC(IT(1))
      RKX=LEVC(1)
      LC(IT(1))=1
      N=1
      DO 170 I=2,IX
      IF(RKX.EQ.LC(IT(I))) GO TO 170
      N=N+1
C     IF THIS MESSAGE IS RECEIVED THEN TRY REVERSING ODD AND EVEN
      IF(N.GT.281) STOP 'LEVC ARRAY LENGTH EXCEEDED'
      LEVC(N)=LC(IT(I))
      RKX=LEVC(N)
170   LC(IT(I))=N
      CALL SORT2 (IX,LR,LC,IT)
C     ORDERS SORTED TABLES
      DO 180 I=1,IX
      K=IT(I)
180   WRITE(11) LC(K),LR(K),WN(K),WT(K)
      REWIND 11
C     READ IN SORTED LIST
      DO 190 I=1,IX
```

```
190   READ(11) LC(I),LR(I),WN(I),WT(I)
      REWIND 11
C     ELIMINATE DUPLICATE TRANSITIONS, SUM UP DATA
      M=0
      NX1=1
      NTRAN=0
C     SET TABLES TO ZERO
      DO 200 I=1,N
      WTJ(I)=0.
      NTB(I)=0
200   QTAB(I)=0.
      RKX=LR(1)
C     FIND NUMBER OF TRANSITIONS TO THIS LEVEL
      I=1
210   I=I+1
      IF(I.GT.IX) GO TO 220
      IF(LR(I).EQ.RKX) GO TO 210
220   NX2=I-1
      IF(NX2-NX1.LT.IML) GO TO 260
      SNI=0.
      YI=0.
      K=0
      DO 240 J=NX1,NX2
      K=K+1
      IF(K.EQ.1) GO TO 230
      IF(LC(J-1).NE.LC(J)) GO TO 230
C REMOVE DUPLICATE CLASSIFICATIONS
      TEMP=RL(RKX)-RL(LEVC(LC(J)))
      PRINT 2220,RKX,LEVC(LC(J)),TEMP,WN(J-1),WN(J)
2220  FORMAT(36HODUPLICATE CLASSIFICATION, 2ND ENTRY
     1 8H IGNORED2F10.0,3F14.4)
      K=K-1
      GO TO 240
C     WV CONTAINS THE WEIGHT OF THE TRANSITION
230   WV(K)=(WTUNC/WT(J))**2
      IF(WTUNC.LT.0.0) WV(K)=1.0
      IF(WT(J).GT.9.0) WV(K)=0.0
C     SNI CONTAINS THE SUM OF THE W(IJ) FOR LEVEL A(I)
      SNI=SNI+WV(K)
C     SUBTRACT INTEGER DIFFERENCE OF ENERGY LEVELS FROM TRANSITION
      TEMP=1.0
      IF(LR(J).LT.LEVC(LC(J))) TEMP=-1.0
      IF(ISOTOP) YV(K)=TEMP*WN(J)
      IF(.NOT.ISOTOP)YV(K)=TEMP*WN(J)-(RL(LR(J))-RL(LEVC(LC(J))))
C     YI CONTAINS THE SUM OF THE WEIGHTED TRANSITIONS IN THE ROW
      YI=YI+WV(K)*YV(K)
      JV(K)=LC(J)
240   CONTINUE
      IF(K.EQ.0) GO TO 260
      IF(SNI.EQ.0) TYPE 2240,LR(J)
```

```
2240  FORMAT(' Remove level',F10.1)
      DTEMP=YI/SNI
      DO 250 J=1,K
      JX=JV(J)
C     QTAB CONTAINS THE Q(I),I=1,...,N
      QTAB(JX)=QTAB(JX)+WV(J)*(DTEMP-YV(J))
C     WTJ CONTAINS THE SUM OF WEIGHTS IN THE COLUMN
      NTB(JX)=NTB(JX)+1
250   WTJ(JX)=WTJ(JX)+WV(J)
      M=M+1
C     STORE A(I) LEVEL DATA ON DISC WITH A RECORD FOR EACH LEVEL
      WRITE(11)K,YI,SNI,RKX,(YV(J),JV(J),WV(J),J=1,K)
C     NTRAN CONTAINS NUMBER OF TRANSITIONS
      NTRAN=NTRAN+K
260   IF(I.GT.IX) GO TO 270
      NX1=I
      RKX=LR(I)
      GO TO 210
C     WRITE OUT Q PART
270   ENDFILE 11
      WRITE(11)(QTAB(I),I=1,N)
      REWIND 11
      TEND=SECNDS(TSTAR)
      TMIN=AINT(TEND/60.)
      TEND=TEND-TMIN*60.
      PRINT 2270,N,M,NTRAN,TMIN,TEND
2270  FORMAT(1H0I4,12H B(J) LEVELS/1H0I4,12H A(I) LEVELS/1H0I5,
     1 12H TRANSITIONS,' TIME=',F6.0,':',F4.0/1H1)
      LDEL=.FALSE.
      DO 280 I=1,N
      IF(NTB(I).GT.IML) GO TO 280
      LDEL=.TRUE.
      ID=ID+1
      RIT(ID)=LEVC(I)
280   CONTINUE
      IF(LDEL) GO TO 120
      PRINT 2280,(LEVC(I),WTJ(I),QTAB(I),NTB(I),I=1,N)
2280  FORMAT(1XF10.0,2F15.4,I5)
      PRINT 2281
2281  FORMAT(15(/))
      WRITE(12) 2,N,INVCK,ISOTOP,SFL,PRNT,IABR,TSTAR,IRUN,IDT,WTUNC
      WRITE(12) (LEVC(I),I=1,N)
      WRITE(12) (NTB(I),I=1,N)
      ENDFILE 12
      CLOSE(UNIT=11)
      CLOSE(UNIT=12)
      STOP 'CLEV1 complete'
      END
```

```
      SUBROUTINE SORT2 (N,IY1,IY2,IS)
C     ------------------------------------------------------------
C     TWO DIMENSIONAL SORT ROUTINE, SORTS FIRST ON IY1, THEN ON IY2
C     IF IY1 HAS EQUAL ENTRIES
C     INDEX OF SORTED TABLE ENTRIES IS STORED IN IS
C     IY1 AND IY2 REMAIN UNCHANGED
C     ------------------------------------------------------------
      REAL*4 IY1,IY2,IA,IB
      VIRTUAL IY1(1),IY2(1)
      DIMENSION IS(1)
      DO 100 J=1,N
100   IS(J)=J
      IF(N.EQ.1) RETURN
      M=N
      M1=8
150   IF(M.EQ.1) GO TO 450
      IF(M.LE.15) M1=4
      M=2*(M/M1)+1
      N1=N-M
      DO 400 L1=1,N1
      K1=L1
200   J=IS(K1)
      K=IS(K1+M)
      IA=IY1(J)
      IB=IY1(K)
      IF(IA.NE.IB) GO TO 250
      IA=IY2(J)
      IB=IY2(K)
250   IF(ABS(IA).LT.ABS(IB)) GO TO 350
      IF(IA.LT.0.0) GO TO 400
300   IS(K1)=K
      IS(K1+M)=J
      K1=K1-M
      IF(K1.GT.0) GO TO 200
      GO TO 400
350   IF(IB.LT.0.0) GO TO 300
400   CONTINUE
      GO TO 150
450   RETURN
      END
```

```
      PROGRAM CLEV2
C     ------------------------------------------------------------
C     CALCULATES THE LEVEL VALUES GIVEN A SET OF TRANSISTIONS
C     CALCULATES THE ISOTOPE SHIFTS GIVEN A SET OF IS TRANSISTIONS
C     SECOND PROGRAM IN THE CLEVEL SERIES
C     ------------------------------------------------------------
C     DIMENSION STATEMENTS FOR THE COLUMN (B LEVELS)
      REAL*8 B,AT
      DIMENSION BVAR(255),ROW(255),RMULT(255),QTAB(255)
      DIMENSION JV(255),YV(150),WV(150),YVAR(150)
      VIRTUAL B(254)
      EQUIVALENCE (BVAR(1),ROW(1)),(QTAB(1),RMULT(1))
C     DIMENSION STATEMENTS FOR THE TRANSISTIONS
      VIRTUAL C(129,254)
C     DIMENSION FOR WEIGHT CLASS STORAGE, LOGICAL VARIABLES
      LOGICAL PRNT,ISOTOP,INVCK,IABR
      DIMENSION WU(38),WTCLAS(38),WRMS(38),WS(38),NRMS(38),
     1 IDT(12),IRUN(40)
      DATA WU/.0001,.0002,.0003,.0004,.0005,.0006,.0007,
     1 .0008,.0009,.001,.002,.003,.004,.005,.006,.007,
     2 .008,.009,.01,.02,.03,.04,.05,.06,.07,.08,.09,.1,
     3 .2,.3,.4,.5,.6,.7,.8,.9,1.,100./
C     DIMENSION C(NCX1,NCX2) WHERE NCX1 > NCX2/2+1
C     NCX2 IS DIMENSION OF C IN B(I) LEVELS
      DATA NCX1,NCX2/129,254/
C     ------------------------------------------------------------
      READ(12) NP,N,INVCK,ISOTOP,SFL,PRNT,IABR,TSTAR,IRUN,IDT,WTUNC
      IF(NP.NE.2) STOP 'DATA FILE NOT CORRECT, R CLEV1 FIRST'
      N1=N-1
      IF(N1.GT.NCX2) STOP 'TOO MANY B LEVELS'
      DO 100 J=1,NCX2
      DO 100 I=1,NCX1
100   C(I,J)=0.
C     COMPUTE ELEMENTS OF THE C MATRIX
C     C-MATRIX IS SYMMETRIC SO ONLY C(I,J) WHERE I.LE.J, IS KEPT IN STORAGE
C     C(I,J) CONTAINED IN C(I,J) FOR I.LE.NCX1
C     C(I,J) CONTAINED IN C(LXE,MX) FOR I.GT.NCX1
C     WHERE LXE=NCX2-I+2 AND MX=J-I+1
C     DIMENSION C(NCX1,NCX2) WHERE NCX1 > NCX2/2+1
110   READ(11,END=170)L,TEMP,SNI,RTEM,(YV(I),JV(I),WV(I),I=1,L)
      L1=1
      IF(JV(1).EQ.1)L1=2
      IF(L1.GT.L) GO TO 110
      DO 130 I=L1,L
      J=JV(I)-1
      IF(J.GT.NCX1) GO TO 140
      C(J,J)=C(J,J)+WV(I)*(1.-WV(I)/SNI)
      IF(I.EQ.L) GO TO 130
      L2=I+1
      DO 120 K=L2,L
      JJ=JV(K)-1
```

```
120   C(J,JJ)=C(J,JJ)-WV(I)*WV(K)/SNI
130   CONTINUE
      GO TO 110
140   DO 160 M=I,L
      LXE=NCX2-JV(M)+3
      C(LXE,1)=C(LXE,1)+WV(M)*(1.-WV(M)/SNI)
      IF(M.EQ.L) GO TO 160
      L2=M+1
      DO 150 K=L2,L
      MX=JV(K)-JV(M)+1
150   C(LXE,MX)=C(LXE,MX)-WV(M)*WV(K)/SNI
160   CONTINUE
      GO TO 110
170   CONTINUE
C     UNIT 10 IS USED TO STORE THE ENTIRE C MATRIX
C     IF A CHECK ON THE INVERSION IS REQUESTED
      IF(.NOT.INVCK) GO TO 250
      DO 240 I=1,N1
      NX=MIN0(NCX1,I)
      DO 180 J=1,NX
180   ROW(J)=C(J,I)
      NX=NX+1
      IF(NX.LE.I) GO TO 200
      IF(NX.GT.N1) GO TO 230
      DO 190 J=NX,N1
190   ROW(J)=C(I,J)
      GO TO 230
200   LXE=NCX2+3-NX
      MX=I-NX+2
      DO 210 J=NX,I
      LXE=LXE-1
      MX=MX-1
210   ROW(J)=C(LXE,MX)
      NX=I+1
      IF(NX.GT.N1) GO TO 230
      DO 220 J=NX,N1
      MX=MX+1
220   ROW(J)=C(LXE,MX)
230   CONTINUE
      WRITE(13) (ROW(IL),IL=1,N1)
240   CONTINUE
      REWIND 13
250   CONTINUE
      ASSIGN 260 TO LEXIT
C INVERT C-MATRIX
C
C STEPS EXECUTED FOR EACH ROW I IN MATRIX
C 1. DMULT=1./C(I,I)   C(I,I) SET TO 1.
C 2. RMULT(IX)=C(IX,I) FOR IX=1,...,I-1
C    RMULT(IX)=C(I,I)=1. FOR IX=I
C    RMULT(IX)=C(I,IX) FOR IX=I+1,...,N1
C 3. ROW(IX)=-DMULT*RMULT(IX) FOR IX=1,...,I-1
C    ROW(IX)=DMULT*RMULT(IX)FORIX=I,...,N1
```

```
C 4. C(IX,I) SET TO 0. FOR IX=1,...,I-1
C    C(I,IX)=ROW(IX) FOR IX=I,...,N1
C 5. FOR ALL ROWS IX WHERE IX.NE.1
C    C(IX,J)=C(IX,J)-RMULT(IX)*ROW(J) FOR J=IX,...,N1
C
      I=0
260   I=I+1
      IF(I.GT.NCX1) GO TO 370
C     WHEN C(I,I)=0, THE B(I+1)LEVEL IS NOT CONNECTED TO THE REFERENCE
C     LEVEL.
      IF(C(I,I).EQ.0.)DMULT=0.
      IF(C(I,I).NE.0.)DMULT=1./C(I,I)
      C(I,I)=1.
      IXN=I-1
      IF(IXN.EQ.0) GO TO 280
      DO 270 IX=1,IXN
      RMULT(IX)=C(IX,I)
      C(IX,I)=0.
270   ROW(IX)=-DMULT*RMULT(IX)
280   DO 290 IX=I,N1
      RMULT(IX)=C(I,IX)
      ROW(IX)=DMULT*RMULT(IX)
290   C(I,IX)=ROW(IX)
300   RMULT(I)=0.
      NX=MIN0(NCX1,N1)
      DO 320 IX=1,NX
      IF(RMULT(IX).EQ.0.) GO TO 320
      DO 310 J=IX,N1
310   C(IX,J)=C(IX,J)-RMULT(IX)*ROW(J)
320   CONTINUE
      IF(NX.EQ.N1) GO TO 360
330   IX1=NCX1+1
      LXE=NCX2-NCX1+2
      DO 350 IX=IX1,N1
      LXE=LXE-1
      IF(RMULT(IX).EQ.0.) GO TO 350
      MX=0
      DO 340 J=IX,N1
      MX=MX+1
340   C(LXE,MX)=C(LXE,MX)-RMULT(IX)*ROW(J)
350   CONTINUE
360   IF(I.GE.N1) GO TO 440
      GO TO LEXIT,(260,380)
370   ASSIGN 380 TO LEXIT
      GO TO 390
380   I=I+1
390   LXE=NCX2-I+2
      IF(C(LXE,1).EQ.0.) DMULT=0.
      IF(C(LXE,1).NE.0.) DMULT=1./C(LXE,1)
      C(LXE,1)=1.
      DO 400 IX=1,NCX1
      RMULT(IX)=C(IX,I)
      C(IX,I)=0.
```

```
400   ROW(IX)=-DMULT*RMULT(IX)
      IXN=I-1
      IF(IXN.EQ.NCX1) GO TO 420
      IX1=NCX1+1
      DO 410 IX=IX1,IXN
      LXE=NCX2-IX+2
      MX=I-IX+1
      RMULT(IX)=C(LXE,MX)
      C(LXE,MX)=0.
410   ROW(IX)=-DMULT*RMULT(IX)
420   LXE=NCX2-I+2
      MX=0
      DO 430 IX=I,N1
      MX=MX+1
      RMULT(IX)=C(LXE,MX)
      ROW(IX)=DMULT*RMULT(IX)
430   C(LXE,MX)=ROW(IX)
      GO TO 300
440   CONTINUE
      IF(.NOT.INVCK) GO TO 520
      RMS=0.
      BIG=0.
      DO 510 I=1,N1
      READ(13) (ROW(IL),IL=1,N1)
      E=-1.
      DO 510 J=I,N1
      NX=MINO(NCX1,J)
      DO 450 K=1,NX
450   E=E+ROW(K)*C(K,J)
      NX=NX+1
      IF(NX.LE.J) GO TO 470
      IF(NX.GT.N1) GO TO 500
      DO 460 K=NX,N1
460   E=E+ROW(K)*C(J,K)
      GO TO 500
470   LXE=NCX2+3-NX
      MX=J-NX+2
      DO 480 K=NX,J
      LXE=LXE-1
      MX=MX-1
480   E=E+ROW(K)*C(LXE,MX)
      NX=J+1
      IF(NX.GT.N1) GO TO 500
      DO 490 K=NX,N1
      MX=MX+1
490   E=E+ROW(K)*C(LXE,MX)
500   CONTINUE
      RMS=RMS+E**2
      BIG=AMAX1(BIG,ABS(E))
      E=0.
```

```
510   CONTINUE
      RMS=SQRT(RMS/(.5*N1*N))
      TEND=SECNDS(TSTAR)
      TMIN=AINT(TEND/60.0)
      TEND=TEND-TMIN*60.
      PRINT 2510,IRUN,IDT,RMS,BIG,TMIN,TEND
2510  FORMAT(' RUN-',40A2/' DATE-',5A2,' TIME-',7A2,
     1 ' MATRIX INVERSION CHECK'/' RMS=',E13.6,'   LARGEST',
     1 ' DEVIATION='E13.6,'  TIME=',F6.0,':',F4.0,15(/))
      TYPE 2520,RMS,BIG,TMIN,TEND
2520  FORMAT(' MATRIX INVERSION CHECK   RMS=',E13.6,
     1 '   LARGEST DEVIATION='E13.6/'  TIME=',F6.0,':',F4.0)
      REWIND 13
520   CONTINUE
      READ(11)(QTAB(I),I=1,N)
      REWIND 11
C COMPUTE CORRECTIONS TO B(J) LEVELS
      B(1)=0.
      NX=MIN0(NCX1,N1)
      DO 550 I=1,NX
      B(I+1)=0.
      IF(C(I,I).EQ.0.) GO TO 550
      DO 530 J=1,I
530   B(I+1)=B(I+1)+C(J,I)*QTAB(J+1)
      IF(I.EO.N1) GO TO 600
      NXX=I+1
      DO 540 J=NXX,N1
540   B(I+1)=B(I+1)+C(I,J)*QTAB(J+1)
550   CONTINUE
      IF(NX.EQ.N1) GO TO 600
      NX=NX+1
      DO 590 I=NX,N1
      LXE=NCX2+2-I
      B(I+1)=0.
      IF(C(LXE,1).EQ.0.) GO TO 590
      DO 560 J=1,NCX1
560   B(I+1)=B(I+1)+C(J,I)*QTAB(J+1)
      NXX=NCX1+1
      LXE=NCX2+3-NXX
      MX=I-NXX+2
      DO 570 J=NXX,I
      LXE=LXE-1
      MX=MX-1
570   B(I+1)=B(I+1)+C(LXE,MX)*QTAB(J+1)
      IF(I.EQ.N1) GO TO 600
      NXX=I+1
      DO 580 J=NXX,N1
      MX=MX+1
580   B(I+1)=B(I+1)+C(LXE,MX)*QTAB(J+1)
590   CONTINUE
```

```
600   CONTINUE
      IM=0
      N1=N-1
      REWIND 11
C CLEAR WEIGHT STATISTICS STORAGE
      DO 610 I=1,38
      WTCLAS(I)=(WTUNC/WU(I))**2
      WRMS(I)=0.
      WS(I)=0.
610   NRMS(I)=0
      SIGMA=0.
      NTRAN=0
      M=0
      BVAR(1)=0.
      NX=MIN0(NCX1,N1)
      DO 620 I=1,NX
620   BVAR(I+1)=C(I,I)
      IF(NX.EQ.N1) GO TO 640
      NX=NX+1
      DO 630 I=NX,N1
      LXE=NCX2+2-I
630   BVAR(I+1)=C(LXE,1)
640   CONTINUE
C
C FOR EACH LEVEL, DETERMINE THE CORRECTION TO THE A(I) LEVEL IN AX
C AND THE VARIANCE TERM IN AVX
650   READ(11,END=760)L,YI,SNI,RLAB,(YV(I),JV(I),WV(I),I=1,L)
660   CONTINUE
      AX=YI
      AVX=SNI
      M=M+1
      DO 670 I=1,L
670   YVAR(I)=0.
      DO 710 I=1,L
      J=JV(I)
      AX=AX+WV(I)*B(J)
      YVAR(I)=YVAR(I)+WV(I)*BVAR(J)
      AVX=AVX+WV(I)*WV(I)*BVAR(J)
      IF(I.EQ.L) GO TO 710
      IF(J.EQ.1) GO TO 710
      NX=I+1
      J=J-1
      DO 700 IX=NX,L
      JX=JV(IX)
      JX=JX-1
      IF(J.GT.NCX1) GO TO 680
      TEMP=C(J,JX)
      GO TO 690
680   LXE=NCX2+2-J
      MX=JX-J+1
      TEMP=C(LXE,MX)
```

```
690   CONTINUE
      AVX=AVX+2.*WV(I)*WV(IX)*TEMP
      YVAR(I)=YVAR(I)+WV(IX)*TEMP
      YVAR(IX)=YVAR(IX)+WV(I)*TEMP
700   CONTINUE
710   CONTINUE
      AVX=AVX/(SNI*SNI)
      AX=AX/SNI
      DO 720 I=1,L
720   YVAR(I)=YVAR(I)/SNI
      AT=AX
      IF(.NOT.ISOTOP)AT=DBLE(AX)+DBLE(AINT(SFL*RLAB))
      WRITE(13)L,AT,AX,AVX,RLAB,(YV(I),JV(I),WV(I),YVAR(I),I=1,L)
      DO 750 I=1,L
      J=JV(I)
C     SUM THE SQUARES OF THE DIFFERENCES BETWEEN THE OBSERVED AND CALCULATED
C     TRANSITIONS
      TEMP=(AX-B(J)-YV(I))**2*WV(I)
      SIGMA=SIGMA+TEMP
      DO 730 IWY=1,37
730   IF(WV(I).GE.WTCLAS(IWY)) GO TO 740
      IWY=38
C
C WRMS CONTAINS THE SUM OF THE WEIGHTED SQUARES OF THE DIFFERENCES
C BETWEEN CALCULATED AND OBSERVED TRANSITIONS FOR A GIVEN WEIGHT CLASS
C NRMS CONTAINS THE NUMBER OF TRANSITIONS IN A GIVEN WEIGHT CLASS
740   WRMS(IWY)=WRMS(IWY)+TEMP
      WS(IWY)=WS(IWY)+WV(I)
      NRMS(IWY)=NRMS(IWY)+1
750   CONTINUE
C NTRAN CONTAINS THE NUMBER OF TRANSITIONS
      NTRAN=NTRAN+L
      GO TO 650
760   READ(12) (QTAB(I),I=1,N)
      READ(12) (JV(I),I=1,N)
      WRITE(14) 3,N,N1,M,NCX1,ISOTOP,INVCK,PRNT,IABR,SFL,TSTAR,WTUNC,
     1 IDT,IRUN,NTRAN,SIGMA
      WRITE(14) NRMS,WRMS,WS,WTCLAS
      WRITE(14) (QTAB(I),I=1,N)
      WRITE(14) (B(I),I=1,N)
      WRITE(14) (BVAR(I),I=1,N)
      WRITE(14) (JV(I),I=1,N)
      ENDFILE 14
      CLOSE(UNIT=12)
      CLOSE(UNIT=11)
      CLOSE(UNIT=14)
      CLOSE(UNIT=13)
      STOP 'CLEV2 Complete'
      END
```

```
      PROGRAM CLEV3
C     ------------------------------------------------------------
C     CALCULATES THE LEVEL VALUES GIVEN A SET OF TRANSISTIONS
C     CALCULATES THE ISOTOPE SHIFTS GIVEN A SET OF IS TRANSISTIONS
C     THRID PROGRAM IN CLEVEL SERIES
C     ------------------------------------------------------------
      REAL*8 WM,WN,A,B,CWN
      REAL*4 LEVR,LEVC
C     DIMENSION STATEMENTS FOR THE ROW (A LEVELS)
      VIRTUAL A(1000),AXP(1000),AVAR(1000),LEVR(1000)
C     DIMENSION STATEMENTS FOR THE COLUMN (B LEVELS)
      VIRTUAL YV(254),WV(254),BVAR(254),YVAR(254),BX(254)
      VIRTUAL B(254)
      VIRTUAL JV(254),LEVC(254),NTB(255),NTA(1000)
C     DIMENSION FOR WEIGHT CLASS STORAGE, LOGICAL VARIABLES
      DIMENSION WU(38),WTCLAS(38),WRMS(38),WS(38),NRMS(38),IDT(12),
     1 IRUN(40)
      DATA WU/.0001,.0002,.0003,.0004,.0005,.0006,.0007,
     1 .0008,.0009,.001,.002,.003,.004,.005,.006,.007,
     2 .008,.009,.01,.02,.03,.04,.05,.06,.07,.08,.09,.1,
     3 .2,.3,.4,.5,.6,.7,.8,.9,1.,100./
      LOGICAL PRNT,ISOTOP,INVCK,IABR
      READ(14) NP,N,N1,M,NCX1,ISOTOP,INVCK,PRNT,IABR,SFL,TSTAR,
     1 WTUNC,IDT,
     1 IRUN,NTRAN,SIGMA
      IF(NP.NE.3) STOP 'PROGRAM SEQUENCE OUT OF ORDER'
      ACOR=0.0
      IF(.NOT.IABR) GO TO 100
      READ(13) L,ACOR
      REWIND 13
      PRINT 2000,ACOR
2000  FORMAT('   Correction for reversal=',F10.5)
100   READ(14) NRMS,WRMS,WS,WTCLAS
      READ(14) (LEVC(I),I=1,N)
      READ(14) (B(I),I=1,N)
      READ(14) (BVAR(I),I=1,N)
      READ(14) (NTB(I),I=1,N)
      NLEV=N1+M
C     COMPUTE SIGMA SQUARED
      SIGMS=SIGMA/(NTRAN-NLEV+1)
C     COMPUTE SIGMA
      TYPE *,SIGMA,SIGMS,SQRT(SIGMS),NTRAN,NLEV
      SIGMA=SQRT(SIGMS)
      DO 150 I=1,N
      BX(I)=B(I)-ACOR
      IF(.NOT.ISOTOP)B(I)=B(I)+AINT(SFL*LEVC(I))-ACOR
150   CONTINUE
      IF(PRNT)PRINT 2150,IRUN,IDT
2150  FORMAT(' RUN-',40A2/' DATE-',5A2,' TIME-',7A2/
     1 3X,'LEVEL     LEVEL WEIGHT'
     2 8X,'SIG OBS      C-O',10X,'(VAR)1/2'  )
```

```
C PRINT TRANSITION, CLASSIFICATION,AND OTHER ASSOCIATED DATA
      K=0
      DO 300 IX=1,M
      READ(13)L,A(IX),AXP(IX),AVAR(IX),LEVR(IX),(YV(I),JV(I),WV(I),
     1 YVAR(I),I=1,L)
      A(IX)=A(IX)-ACOR
      AXP(IX)=AXP(IX)-ACOR
      IF(.NOT.PRNT) GO TO 300
      NTA(IX)=L
C FOR EACH TRANSITION COMPUTE THE CALCULATED TRANSITON AND ITS
C VARIANCE AND PRINT ALL THE DATA ASSOCIATED WITH THE TRANSITION
      DO 250 I=1,L
      J=JV(I)
      CWN=A(IX)-B(J)
      IF(LEVR(IX)-LEVC(J).LT.0) CWN=-CWN
      DIFF=AXP(IX)-BX(J)-YV(I)
      IF(LEVR(IX)-LEVC(J).LT.0) DIFF=-DIFF
      WM=CWN-DIFF
      COM=' '
      IF(WV(I).LT.0.00001) COM='--'
      IF(DIFF.EQ.0.) GO TO 200
      TEMP=(SIGMA/DIFF)**2
C IF THE DIFFERENCE IS TWICE THE UNCERTAINTY STAR THE PRINTOUT
C IF THE DIFFERENCE IS THREE TIMES THE UNCERTAINTY DOUBLE STAR THE
C PRINTOUT
      IF(4.*TEMP.LT.WV(I)) COM='* '
      IF(9.*TEMP.LT.WV(I)) COM='**'
200   YVAR(I)=AVAR(IX)+BVAR(J)-2.*YVAR(I)
      WRITE(15) WM,WV(I),DIFF
      VARRT=SIGN(SQRT(ABS(YVAR(I))*SIGMS),YVAR(I))
      PRINT 2200,LEVR(IX),LEVC(J),WV(I),WM,DIFF,COM,VARRT
2200  FORMAT(2F10.0,2X,F11.4,2X,F13.4,1X,F10.5,A2,2X,F8.5)
      IF(4.*TEMP.LT.WV(I)) WRITE(16,2200) LEVR(IX),LEVC(J),
     1 WV(I),WM,DIFF,COM,VARRT
      IF(IFL.EQ.0.OR.DIFF.EQ.0.0) GO TO 250
      K=K+1
250   CONTINUE
      PRINT 2250
2250  FORMAT(' ')
300   CONTINUE
      CLOSE(UNIT=13)
      CLOSE(UNIT=14)
      CLOSE(UNIT=15)
      WRITE(16,2200) 0.0,0.0,0.0
      CLOSE(UNIT=16)
      OPEN(UNIT=1,NAME='DK1:CLLEV.DAT',TYPE='NEW')
      PRINT 2300
2300  FORMAT(1H1,3X,5HLEVEL,3X,16HCALCULATED LEVEL,3X,
     1 9HSQRT(VAR)10X12HVAR/SIGMA**25X,'CORRECTION',2X,'NUM.')
      DO 350 I=1,M
      VARRT=SQRT(ABS(AVAR(I))*SIGMS)
      WRITE(1,2350)LEVR(I),A(I),VARRT,AVAR(I),AXP(I),NTA(I)
350   PRINT 2350,LEVR(I),A(I),VARRT,AVAR(I),AXP(I),NTA(I)
```

```
2350  FORMAT(F10.0,4XF13.5,4XF8.5,4XF20.5 ,2X,F12.6,I6)
C
C B LEVELS THAT HAVE ZERO VARIANCES ARE NOT CONNECTED TO THE
C REFERENCE LEVEL   SIGMA IS INCORRECT SO THE UNCONNECTED LEVELS
C SHOULD BE REMOVED
C
      PRINT 2300
      WRITE(1,2350) -1.
      PRINT 2350,LEVC(1),B(1),0.0,0.0,0.0,NTB(1)
      WRITE(1,2350) LEVC(1),B(1),0.0,0.0,0.0,NTB(1)
      DO 450 I=2,N
      IF(BVAR(I).NE.0.) GO TO 400
      PRINT 2351,LEVC(I)
2351  FORMAT(F10.0,4X,'LEVEL NOT CONNECTED TO THE REFERENCE LEVEL.')
      GO TO 450
400   VARRT=SQRT(ABS(BVAR(I))*SIGMS)
      WRITE(1,2350) LEVC(I),B(I),VARRT,BVAR(I),BX(I),NTB(I)
      PRINT 2350,LEVC(I),B(I),VARRT,BVAR(I),BX(I),NTB(I)
450   CONTINUE
      WRITE(1,2350) -2.
      PRINT 2450,SIGMA,SIGMS,N,M,NTRAN
2450  FORMAT(7HOSIGMA=F9.6,3X14HSIGMA SQUARED=F15.12,3XI3,
     1 12H B(J) LEVELS3XI4,12H A(I) LEVELS3XI5,12H TRANSITIONS)
      PRINT 2451
2451  FORMAT(1HO/1HO,1X,5HCLASS,7X,6HWEIGHT,9X,3HRMS,7X,
     1 8HQUANTITY)
C
C PRINT WEIGHT STATISTICS
C
      DO 500 I=1,37
      IF(NRMS(I).EQ.0) GO TO 500
      IF(WS(I).EQ.0.0) WRMS(I)=999.999
      IF(WS(I).NE.0.0) WRMS(I)=SQRT(WRMS(I)/WS(I))
      PRINT 2452,WU(I),WTCLAS(I),WRMS(I),NRMS(I)
2452  FORMAT(1X,F6.4,4X,F13.4,4X,F9.6,4X,I5)
500   CONTINUE
      IF(NRMS(38).EQ.0) GO TO 550
      IF(WS(38).EQ.0) WRMS(38)=999.999
      IF(WS(38).NE.0) WRMS(38)=SQRT(WRMS(38)/WS(38))
      PRINT 2500,WRMS(38),NRMS(38)
2500  FORMAT(1X15HGREATER THAN 1.10XF9.6,4XI5)
550   CONTINUE
      TEND=SECNDS(TSTAR)
      TMIN=AINT(TEND/60.)
      TEND=TEND-TMIN*60.0
      PRINT 2550,TMIN,TEND
2550  FORMAT('  TOTAL TIME FOR CLEVEL RUN=',F5.0,':',F4.0,
     1 ' SECONDS',15(/))
      CLOSE(UNIT=1)
      STOP 'CLEVEL finished'
      END
```

APPENDIX B
CLEVEL INPUT

Parameter Input

```
TFFTT 1,0.1,0.002
EXAMPLE - Th III
DK1:TH3CIN.DAT
207101.0
```

Data File

```
 7437.3384     75003       632 0.00725
 7632.2743    132082    208403 0.01000
 7740.6893    111231    188632 0.00676
 7857.8207     79211       632 0.00193
 8116.5532     79211    160372 0.00231
 8277.3039    154533     71762 0.01136
 8467.0104    190095    105424 0.00055
 8531.7816    132082     46762 0.01282
 8683.1031    101802    188632 0.00926
 8749.8164    207101    119610 0.01163
 8906.8513     89804    178885 0.00048
 8915.5934    154533     65374 0.00431
 9450.5833     84376    178885 0.01389
 9494.0884    112765    207716 0.00016
 9745.6595     81414    178885 0.00014
 9749.4190     62882    160372 0.01020
 9953.4382         4     99533 0.01064
10087.9751     50603    151484 0.00011
10099.3388    107413    208403 0.01250
10321.6927     48263    151484 0.00040
10507.0618    112765    217834 0.00116
10518.7611    154533    259724 0.00102
10542.8975         4    105424 0.01613
10658.8778     44895    151484 0.00532
10659.7221    101802    208403 0.00065
10677.8835    107413       632 0.01515
10776.9790    154533     46762 0.00108
10942.7808     79211    188632 0.00085
10977.0970     50603    160372 0.00212
11042.7190    107413    217834 0.00045
11210.8149     48263    160372 0.00215
11576.6015     63104    178885 0.00227
12334.0707     84376    207716 0.00031
12472.0928    190095     65374 0.00244
12575.6488     62882    188632 0.00321
12621.4246     25273    151484 0.00115
12698.7394     81414    208403 0.00056
```

12803.3122	89804	217834	0.00113
13339.8834	75003	208403	0.00287
13397.7684	44895	178885	0.00016
13803.3255	50603	188632	0.00157
14037.0431	48263	188632	0.00083
14283.2639	75003	217834	0.00152
14529.6812	63104	208403	0.00185
14552.2684	62882	208403	0.00095
14699.1067	31884	178885	0.00187
15148.5195	4	151484	0.00009
15473.0614	63104	217834	0.00108
15526.8828	5102	160372	0.00424
15682.3683	31812	188632	0.00144
15779.9447	50603	208403	0.00031
16013.6625	48263	208403	0.00032
16336.7737	25273	188632	0.00093
16957.0451	48263	217834	0.00562
16991.6162	89804	259724	0.00040
17109.3252	111231	282322	0.00254
17413.8628	154533	328672	0.00159
17491.3528	107413	282322	0.00082
17652.1873	31884	208403	0.00694
18176.3358	111231	292990	0.00543
18313.3926	25273	208403	0.00029
18353.1109	5102	188632	0.00008
18471.5673	75003	259724	0.00266
18595.5683	31884	217834	0.00012
18943.8881	112765	302206	0.01351
19170.2398	112320	304021	0.00195
19256.7739	25273	217834	0.00118
19279.6756	111231	304021	0.00177
19570.6945	190095	385804	9.9
19659.0608	132082	328672	0.00072
19661.3654	63104	259724	0.00042
20222.0892	101802	304021	0.00081
20287.5502	392801	189930	0.00352
20329.7308	5102	208403	0.00182
20647.6817	207101	632	0.01316
20911.6283	50603	259724	0.00549
21378.4262	79211	292990	0.00137
21744.0957	111231	328672	0.00376
21783.8694	4	217834	0.00035
21783.8694	84376	302206	0.00035
21944.2809	62882	282322	0.00031
22126.1245	107413	328672	0.00296
22481.7673	79211	304021	0.00725
22686.5079	101802	328672	0.01515
22759.8706	509921	282322	0.00667
22783.8718	31884	259724	0.00061
23405.6772	48263	282322	0.02000
24691.7572	190095	437014	0.01563
24946.1861	79211	328672	0.02941
25050.9963	31812	282322	0.00163

25972.1719	4	259724	0.00051
26840.3681	372802	104402	0.00309
27075.0000	79211	349962	0.00127
27303.7941	112765	385804	0.00260
27319.7375	392801	119610	0.00595
27319.7375	555523	282322	0.00595
27690.4911	107413	384313	0.00735
27721.7427	5102	282322	0.00065
27839.4448	107413	385804	9.9
28250.8716	101802	384313	0.00224
28251.1914	190095	472604	0.00303
28427.3200	444653	160372	0.00962
28501.8054	50603	335623	0.00038
28735.5269	48263	335623	0.00382
29451.0833	89804	384313	0.00562
29600.0445	89804	385804	0.00123
29935.5437	50603	349962	0.00082
30104.1219	372802	71762	0.00147
30169.2606	48263	349962	0.00029
30289.8898	81414	384313	0.00060
30374.0458	31884	335623	0.00336
30380.8473	31812	335623	0.00013
30931.0359	75003	384313	0.00078
31035.2539	25273	335623	0.00012
31079.9939	75003	385804	0.00066
31117.1341	499803	188632	0.00538
31571.9098	107413	423133	9.9
31756.3480	372802	55241	0.00032
31814.5839	31812	349962	0.00072
32104.7555	392801	71762	0.00291
32120.8318	63104	384313	0.00088
32269.7916	63104	385804	0.00046
32356.5127	5102	328672	0.00342
32424.8562	112765	437014	0.00164
32468.9915	25273	349962	0.00044
32603.7968	372802	46762	0.00059
32636.0437	111231	437592	0.00450
33018.0724	107413	437592	0.00500
33224.7990	372802	40563	0.00362
33332.4970	89804	423133	0.00538
33370.4894	112320	446031	0.00633
33371.0940	50603	384313	0.00216
33479.9195	111231	446031	0.00376
33562.3491	4	335623	0.00010
33578.4555	101802	437592	0.00521
33604.8118	48263	384313	0.00197
33753.7722	48263	385804	0.00467
33756.9827	392801	55241	0.00087
34090.9574	44895	385804	0.00061
34134.7008	440882	99533	0.00176
34171.3048	81414	423133	0.00243
34190.8061	392801	50890	0.00129
34263.0623	132082	474713	0.00568

34352.2663	154533	498052	0.00459
34383.8873	422590	78751	0.00331
34485.3285	5102	349962	0.00018
34604.4312	392801	46762	0.00333
34812.4512	75003	423133	0.00336
35243.3423	31884	384313	0.00388
35392.2971	31884	385804	0.00107
35559.9159	81414	437014	0.00365
35904.5453	25273	384313	0.00196
35984.2960	112765	472604	0.00543
36024.8328	62882	423133	0.00195
36145.1322	112765	474215	0.00373
36212.4537	440882	78751	0.00352

APPENDIX C
CLEVEL OUTPUT

Terminal Output

```
.RU CLEV1
INPUT FILE-TH3C2.DAT
 IREV=T  IABR=F  ISOTOP=F  INVCK=T  PRNT=T  IML=    1
 LEVEL SCALE FACTOR=     0.100
 DATE-22-MAR-83

 RUN-EXAMPLE - Th III


FINISHED LOADING        154TRANSITIONS, TIME=    0.:  5.
FINISHED LOADING        149TRANSITIONS, TIME=    0.: 15.
FINISHED LOADING        147TRANSITIONS, TIME=    0.: 25.
STOP -- CLEV1 complete
.RU CLEV2
MATRIX INVERSION CHECK RMS= 0.239143E-06   LARGEST DEVIATION= 0.315160E-05  TIME
=    3.: 43.
STOP -- CLEV2 Complete
.RU CLEV3
 1.9248277E-04  2.2124457E-06  1.4874293E-03    138     52
STOP -- CLEVEL finished

STOP -- NORMAL
```

CLEV1 Output

```
 IREV=T  IABR=F  ISOTOP=F  INVCK=T  PRNT=T  IML=    1
 LEVEL SCALE FACTOR=     0.100
 DATE-22-MAR-83  TIME-09:33:06 FILE-DK1:TH3CIN.DAT
 RUN-EXAMPLE - Th III

WAVE NUMBER LEVEL RUN  WTUNC=   0.00200
    8749.8164   119610.   207101.     0.00000 EXCLUDED LINE
   20647.6817      632.   207101.     0.00000 EXCLUDED LINE
FINISHED LOADING        154TRANSITIONS, TIME=    0.:  5.

  31 B(J) LEVELS

  30 A(I) LEVELS

  146 TRANSITIONS TIME=    0.: 10.

    8749.8164   119610.   207101.     0.00000 EXCLUDED LINE
   20647.6817      632.   207101.     0.00000 EXCLUDED LINE
   22759.8706   282322.   509921.     0.00000 EXCLUDED LINE
   27319.7375   282322.   555523.     0.00000 EXCLUDED LINE
   28427.3200   160372.   444653.     0.00000 EXCLUDED LINE
   31117.1341   188632.   499803.     0.00000 EXCLUDED LINE
   34383.8873    78751.   422590.     0.00000 EXCLUDED LINE
FINISHED LOADING        149TRANSITIONS, TIME=    0.: 15.
```

```
 26 B(J) LEVELS
 29 A(I) LEVELS

 140 TRANSITIONS TIME=    0.: 20.

    8749.8164   119610.   207101.     0.00000 EXCLUDED LINE
   20647.6817      632.   207101.     0.00000 EXCLUDED LINE
   22759.8706   282322.   509921.     0.00000 EXCLUDED LINE
   27319.7375   282322.   555523.     0.00000 EXCLUDED LINE
   28427.3200   160372.   444653.     0.00000 EXCLUDED LINE
   31117.1341   188632.   499803.     0.00000 EXCLUDED LINE
   34134.7008    99533.   440882.     0.00000 EXCLUDED LINE
   34383.8873    78751.   422590.     0.00000 EXCLUDED LINE
   36212.4537    78751.   440882.     0.00000 EXCLUDED LINE
FINISHED LOADING        147TRANSITIONS, TIME=    0.: 25.

 25 B(J) LEVELS

 28 A(I) LEVELS

 138 TRANSITIONS TIME=    0.: 29.

       4.        941.8745      -200.9296    5
    5102.        759.6964        21.2068    6
   25273.        357.5648       -56.9852    7
   31812.        247.8370        76.8748    4
   31884.        293.8683         3.3335    7
   44895.        167.1412        -8.7400    3
   48263.        119.9215        35.2894   10
   50603.        409.3543       112.7991    8
   62882.         47.5340        -5.2947    5
   63104.         52.1190        12.8491    6
   75003.         18.9700        -0.4287    7
   79211.         12.0517        -5.5439    7
   81414.        228.9255        15.1775    5
   84376.         74.2971       -32.3255    3
   89804.         48.4025         0.1414    6
  101802.         16.5727         3.3154    6
  107413.         26.4355        -3.5413    9
  111231.          2.8834        -0.8100    7
  112320.          1.1518        -0.0081    2
  112765.        161.4592        33.9684    6
  132082.          7.7804        -0.3625    3
  154533.          9.1026         0.3396    5
  190095.         14.3471         0.1080    5
  372802.         52.4045        -3.8349    3
  392801.          6.1178         3.4013    3
```

CLEV2 Output

```
RUN-EXAMPLE - Th III
DATE-22-MAR-83 TIME-09:33:06 MATRIX INVERSION CHECK
RMS= 0.239143E-06   LARGEST DEVIATION= 0.315160E-05  TIME=    3.: 43.
```

CLEV3 Output

RUN-EXAMPLE - Th III
DATE-22-MAR-83 TIME-09:33:06

LEVEL	LEVEL	WEIGHT	SIG OBS	C-O	(VAR)1/2
632.	75003.	0.0761	7437.3384	-0.00151	0.00151
632.	79211.	1.0739	7857.8207	0.00012	0.00138
632.	107413.	0.0174	10677.8835	-0.00100	0.00150
46762.	132082.	0.0243	8531.7816	-0.00083	0.00134
46762.	154533.	3.4294	10776.9790	0.00001	0.00080
46762.	372802.	11.4910	32603.7968	-0.00001	0.00043
46762.	392801.	0.3607	34604.4312	0.00020	0.00076
55241.	372802.	39.0625	31756.3480	0.00001	0.00024
55241.	392801.	5.2847	33756.9827	-0.00008	0.00061
65374.	154533.	0.2153	8915.5934	0.00005	0.00262
65374.	190095.	0.6719	12472.0928	-0.00002	0.00172
71762.	154533.	0.0310	8277.3039	0.00001	0.00132
71762.	372802.	1.8511	30104.1219	-0.00020	0.00098
71762.	392801.	0.4724	32104.7555	0.00080	0.00110
105424.	4.	0.0154	10542.8975	0.00260	0.00280
105424.	190095.	13.2231	8467.0104	0.00000	0.00041
151484.	4.	493.8272	15148.5195	-0.00008	0.00006
151484.	25273.	3.0246	12621.4246	-0.00058	0.00011
151484.	44895.	0.1413	10658.8778	0.00148	0.00031
151484.	48263.	25.0000	10321.6927	0.00041	0.00017
151484.	50603.	330.5785	10087.9751	0.00010	0.00008
160372.	5102.	0.2225	15526.8828	-0.00005	0.00091
160372.	48263.	0.8653	11210.8149	-0.00001	0.00091
160372.	50603.	0.8900	10977.0970	-0.00003	0.00091
160372.	62882.	0.0384	9749.4190	0.00171	0.00097
160372.	79211.	0.7496	8116.5532	-0.00003	0.00097
178885.	31884.	1.1439	14699.1067	0.00102	0.00031
178885.	44895.	156.2500	13397.7684	-0.00006	0.00012
178885.	63104.	0.7763	11576.6015	-0.00000	0.00031
178885.	81414.	204.0816	9745.6595	-0.00002	0.00010

178885.	84376.	0.0207	9450.5833	0.00061	0.00084
178885.	89804.	17.3611	8906.8513	0.00072*	0.00027
188632.	5102.	625.0001	18353.1109	-0.00000	0.00006
188632.	25273.	4.6248	16336.7737	0.00024	0.00021
188632.	31812.	1.9290	15682.3683	-0.00093	0.00022
188632.	48263.	5.8064	14037.0431	-0.00007	0.00020
188632.	50603.	1.6228	13803.3255	-0.00038	0.00022
188632.	62882.	0.3882	12575.6488	0.00006	0.00038
188632.	79211.	5.5363	10942.7808	0.00051	0.00050
188632.	101802.	0.0466	8683.1031	-0.00012	0.00050
188632.	111231.	0.0875	7740.6893	0.00134	0.00102
207716.	84376.	41.6233	12334.0707	0.00000	0.00023
207716.	112765.	156.2500	9494.0884	-0.00000	0.00012
208403.	5102.	1.2076	20329.7308	-0.00070	0.00022
208403.	25273.	47.5624	18313.3926	0.00054*	0.00015
208403.	31884.	0.0831	17652.1873	0.00048	0.00025
208403.	48263.	39.0625	16013.6625	-0.00027	0.00017
208403.	50603.	41.6233	15779.9447	-0.00038	0.00015
208403.	62882.	4.4321	14552.2684	-0.00034	0.00038
208403.	63104.	1.1687	14529.6812	0.00036	0.00032
208403.	75003.	0.4856	13339.8834	0.00104	0.00043
208403.	81414.	12.7551	12698.7394	0.00013	0.00028
208403.	101802.	9.4675	10659.7221	0.00008	0.00045
208403.	107413.	0.0256	10099.3388	0.00003	0.00036
208403.	132082.	0.0400	7632.2743	0.00130	0.00114
217834.	4.	32.6531	21783.8694	-0.00035	0.00021
217834.	25273.	2.8727	19256.7739	-0.00025	0.00022
217834.	31884.	277.7778	18595.5683	-0.00000	0.00009
217834.	48263.	0.1266	16957.0451	-0.00235	0.00026
217834.	63104.	3.4294	15473.0614	0.00067	0.00030
217834.	75003.	1.7313	14283.2639	0.00106	0.00043
217834.	89804.	3.1326	12803.3122	0.00039	0.00031
217834.	107413.	19.7531	11042.7190	0.00035	0.00030
217834.	112765.	2.9727	10507.0618	0.00038	0.00076
259724.	4.	15.3787	25972.1719	0.00048	0.00025
259724.	31884.	10.7498	22783.8718	-0.00017	0.00026
259724.	50603.	0.1327	20911.6283	-0.00013	0.00026
259724.	63104.	22.6757	19661.3654	0.00001	0.00025
259724.	75003.	0.5653	18471.5673	0.00100	0.00042
259724.	89804.	25.0000	16991.6162	-0.00027	0.00025
259724.	154533.	3.8447	10518.7611	0.00011	0.00070
282322.	5102.	9.4675	27721.7427	0.00026	0.00034
282322.	31812.	1.5055	25050.9963	0.00314*	0.00036
282322.	48263.	0.0100	23405.6772	-0.00210	0.00036
282322.	62882.	41.6233	21944.2809	0.00002	0.00022
282322.	107413.	5.9488	17491.3528	-0.00110	0.00041
282322.	111231.	0.6200	17109.3252	-0.00250	0.00103

292990.	79211.	2.1312	21378.4262	0.00002	0.00099
292990.	111231.	0.1357	18176.3358	-0.00026	0.00144
302206.	84376.	32.6531	21783.8694	0.00000	0.00026
302206.	112765.	0.0219	18943.8881	-0.00100	0.00037
304021.	79211.	0.0761	22481.7673	-0.00005	0.00087
304021.	101802.	6.0966	20222.0892	-0.00028	0.00057
304021.	111231.	1.2768	19279.6756	0.00098	0.00099
304021.	112320.	1.0519	19170.2398	0.00044	0.00140
328672.	5102.	0.3420	32356.5127	0.00259	0.00102
328672.	79211.	0.0046	24946.1861	-0.00040	0.00113
328672.	101802.	0.0174	22686.5079	-0.00053	0.00109
328672.	107413.	0.4565	22126.1245	-0.00048	0.00103
328672.	111231.	0.2829	21744.0957	-0.00067	0.00133
328672.	132082.	7.7161	19659.0608	-0.00001	0.00053
328672.	154533.	1.5822	17413.8628	-0.00025	0.00095
335623.	4.	400.0001	33562.3491	0.00011	0.00007
335623.	25273.	277.7778	31035.2539	-0.00008	0.00008
335623.	31812.	236.6864	30380.8473	-0.00005	0.00009
335623.	31884.	0.3543	30374.0458	0.00266	0.00023
335623.	48263.	0.2741	28735.5269	-0.00399	0.00018
335623.	50603.	27.7008	28501.8054	-0.00040	0.00011
349962.	5102.	123.4568	34485.3285	-0.00001	0.00013
349962.	25273.	20.6612	32468.9915	0.00003	0.00018
349962.	31812.	7.7161	31814.5839	0.00106	0.00020
349962.	48263.	47.5624	30169.2606	0.00002	0.00017
349962.	50603.	5.9488	29935.5437	-0.00099	0.00019
349962.	79211.	2.4800	27075.0000	-0.00110	0.00051
384313.	25273.	1.0412	35904.5453	-0.00156	0.00037
384313.	31884.	0.2657	35243.3423	-0.00391	0.00038
384313.	48263.	1.0307	33604.8118	0.00104	0.00038
384313.	50603.	0.8573	33371.0940	0.00092	0.00037
384313.	63104.	5.1653	32120.8318	0.00036	0.00036
384313.	75003.	6.5746	30931.0359	-0.00085	0.00041
384313.	81414.	11.1111	30289.8898	0.00034	0.00033
384313.	89804.	0.1266	29451.0833	-0.00062	0.00038
384313.	101802.	0.7972	28250.8716	0.00118	0.00056
384313.	107413.	0.0740	27690.4911	-0.00166	0.00047
385804.	31884.	3.4938	35392.2971	0.00047	0.00033
385804.	44895.	10.7498	34090.9574	0.00079	0.00030
385804.	48263.	0.1834	33753.7722	-0.00018	0.00035
385804.	63104.	18.9036	32269.7916	-0.00025	0.00027
385804.	75003.	9.1827	31079.9939	0.00034	0.00038
385804.	89804.	2.6439	29600.0445	-0.00263*	0.00033
385804.	107413.	0.0000	27839.4448	0.00382—	0.00043
385804.	112765.	0.5917	27303.7941	-0.00264	0.00080
385804.	190095.	0.0000	19570.6945	-0.00668—	0.00278

423133.	62882.	1.0519	36024.8328	0.00049	0.00102
423133.	75003.	0.3543	34812.4512	-0.00152	0.00106
423133.	81414.	0.6774	34171.3048	-0.00003	0.00103
423133.	89804.	0.1382	33332.4970	0.00031	0.00104
423133.	107413.	0.0000	31571.9098	-0.00573	0.00106
437014.	81414.	0.3002	35559.9159	-0.00149	0.00129
437014.	112765.	1.4872	32424.8562	0.00035	0.00112
437014.	190095.	0.0164	24691.7572	-0.00429	0.00296
437592.	101802.	0.1474	33578.4555	0.00029	0.00214
437592.	107413.	0.1600	33018.0724	0.00004	0.00215
437592.	111231.	0.1975	32636.0437	-0.00025	0.00218
446031.	111231.	0.2829	33479.9195	0.00163	0.00244
446031.	112320.	0.0998	33370.4894	-0.00461	0.00271
472604.	112765.	0.1357	35984.2960	-0.00073	0.00290
472604.	190095.	0.4357	28251.1914	0.00023	0.00208

LEVEL	CALCULATED LEVEL	SQRT(VAR)	VAR/SIGMA**2	CORRECTION	NUM.
632.	63.26720	0.00146	0.96580	0.267198	3
46762.	4676.43217	0.00108	0.52589	0.432168	4
55241.	5523.88095	0.00118	0.63252	-0.119050	2
65374.	6537.81772	0.00265	3.18416	0.817722	2
71762.	7176.10727	0.00151	1.02894	0.107266	3
105424.	10542.90010	0.00280	3.54192	0.900102	2
151484.	15148.51942	0.00006	0.00183	0.519416	5
160372.	16037.64119	0.00091	0.37619	0.641191	5
178885.	17887.40848	0.00029	0.03759	-0.591524	6
188632.	18863.86933	0.00021	0.02020	0.869334	9
207716.	20770.89527	0.00079	0.27871	-0.104732	2
208403.	20840.48853	0.00015	0.01016	0.488533	12
217834.	21783.86905	0.00021	0.01918	0.869050	9
259724.	25972.17238	0.00025	0.02802	0.172384	7
282322.	28232.50140	0.00035	0.05554	0.501399	6
292990.	29299.51424	0.00111	0.55760	0.514238	2
302206.	30220.69397	0.00086	0.33300	0.693968	2
304021.	30402.85528	0.00071	0.22899	0.855276	4
328672.	32867.27372	0.00101	0.45843	0.273725	7
335623.	33562.34921	0.00007	0.00219	0.349214	6
349962.	34996.08693	0.00019	0.01553	0.086926	6
384313.	38431.63914	0.00036	0.05948	0.639140	10
385804.	38580.59833	0.00032	0.04768	0.598326	9
423133.	42313.05377	0.00103	0.47622	0.053768	5
437014.	43701.66341	0.00128	0.74570	0.663415	3
437592.	43759.22215	0.00215	2.08121	0.222146	3
446031.	44603.09983	0.00258	3.00034	0.099826	2
472604.	47261.10213	0.00291	3.82740	1.102134	2

LEVEL	CALCULATED LEVEL	SQRT(VAR)	VAR/SIGMA**2	CORRECTION	NUM.
4.	0.00000	0.00000	0.00000	0.000000	5
5102.	510.75844	0.00021	0.01934	0.758436	6
25273.	2527.09540	0.00010	0.00485	0.095397	7
31812.	3181.50196	0.00012	0.00617	0.501961	4
31884.	3188.30075	0.00022	0.02149	0.300753	7
44895.	4489.64013	0.00030	0.04168	0.640132	3
48263.	4826.82630	0.00018	0.01414	0.826304	10
50603.	5060.54422	0.00009	0.00400	0.544216	8
62882.	6288.22048	0.00038	0.06590	0.220477	5
63104.	6310.80698	0.00030	0.04123	0.806977	6
75003.	7500.60409	0.00043	0.08181	0.604089	7
79211.	7921.08802	0.00053	0.12801	0.088021	7
81414.	8141.74900	0.00029	0.03711	0.749000	5
84376.	8436.82457	0.00082	0.30243	-0.175433	3
89804.	8980.55645	0.00030	0.04142	0.556455	6
101802.	10180.76636	0.00047	0.09980	0.766355	6
107413.	10741.14970	0.00035	0.05388	0.149702	9
111231.	11123.17870	0.00102	0.46708	0.178698	7
112320.	11232.61504	0.00156	1.10575	0.615038	2
112765.	11276.80687	0.00078	0.27238	0.806868	6
132082.	13208.21293	0.00113	0.58036	0.212934	3
154533.	15453.41117	0.00073	0.24019	0.411174	5
190095.	19009.91051	0.00277	3.47444	0.910505	5
372802.	37280.22896	0.00116	0.60877	0.228962	3
392801.	39280.86357	0.00132	0.78254	0.863566	3

SIGMA= 0.001487 SIGMA SQUARED= 0.000002212446 25 B(J) LEVELS 28 A(I) LEVELS
138 TRANSITIONS

CLASS	WEIGHT	RMS	QUANTITY
0.0001	400.0001	0.000076	3
0.0002	100.0000	0.000060	8
0.0003	44.4444	0.000379	2
0.0004	25.0000	0.000261	10
0.0005	16.0000	0.000357	5
0.0006	11.1111	0.000283	5
0.0007	8.1633	0.000422	5
0.0008	6.2500	0.000783	3
0.0009	4.9383	0.000629	7
0.0010	4.0000	0.000294	2
0.0020	1.0000	0.000971	27
0.0030	0.4444	0.001094	14
0.0040	0.2500	0.002072	11
0.0050	0.1600	0.000142	5
0.0060	0.1111	0.001039	8
0.0070	0.0816	0.002916	3
0.0080	0.0625	0.001293	3
0.0100	0.0400	0.000887	2
0.0200	0.0100	0.001627	11
0.0300	0.0044	0.000396	1
GREATER THAN 1.		*********	3

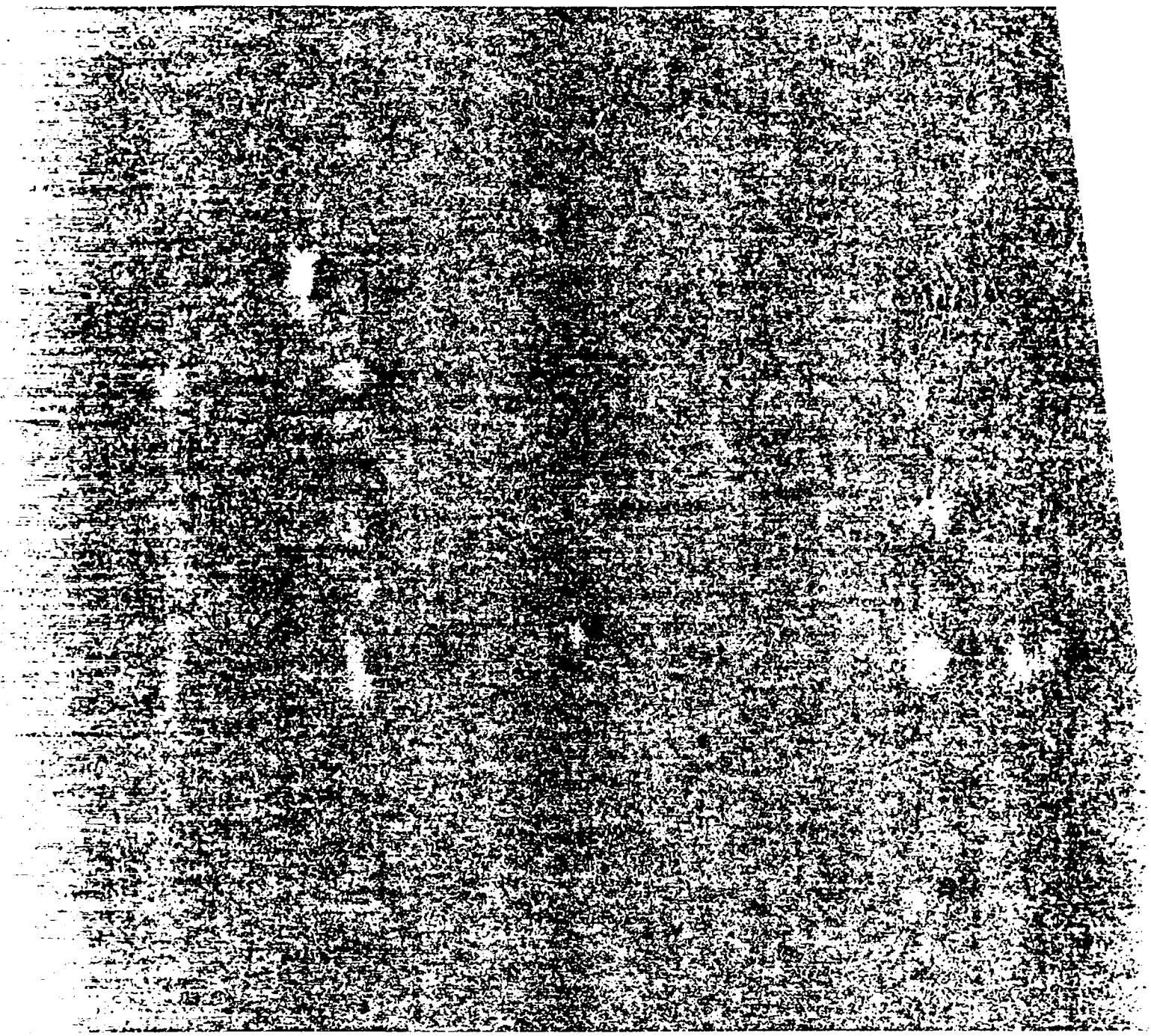
TOTAL TIME FOR CLEVEL RUN= 4.: 6. SECONDS

Printed in the United States of America
Available from
National Technical Information Service
US Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

Microfiche (A01)

Page Range	NTIS Price Code	Page Range	NTIS Price Code	Page Range	NTIS Price Code	Page Range	NTIS Price Code
001-025	A02	151-175	A08	301-325	A14	451-475	A20
026-050	A03	176-200	A09	326-350	A15	476-500	A21
051-075	A04	201-225	A10	351-375	A16	501-525	A22
076-100	A05	226-250	A11	376-400	A17	526-550	A23
101-125	A06	251-275	A12	401-425	A18	551-575	A24
126-150	A07	276-300	A13	426-450	A19	576-600	A25
						601-up*	A99

*Contact NTIS for a price quote.



Los Alamos