Copy 3

Submitted for Publication — J Math. Geol. Restricted until Published.

COMMONWEALTH OF AUSTRALIA

DEPARTMENT OF NATIONAL DEVELOPMENT

BUREAU OF MINERAL RESOURCES, GEOLOGY AND GEOPHYSICS

Record 1972/124

NONLINEAR MAPPING AND A RELATED R-MODE TECHNIQUE FOR COMPRESSION OF MULTIVARIATE DATA

by

S. HENLEY[1]

The information contained in this report has been obtained by the Department of National Development as part of the policy of the Commonwealth Government to assist in the exploration and development of mineral resources. It may not be published in any form or used in a company prospectus or statement without the permission in writing of the Director, Bureau of Mineral Resources, Geology & Geophysics.

NONLINEAR MAPPING AND A RELATED R-MODE TECHNIQUE FOR COMPRESSION OF MULTIVARIATE DATA

by

S. Henley

RECORD 1972/124

The information contained in this report has been obtained by the Department of National Development as part of the policy of the Commonwealth Government to assist in the exploration and development of mineral resources. It may not be published in any form or used in a company prospectus or statement without the permission in writing of the Director, Bureau of Mineral Resources, Geology and Geophysics.

Contents

Nonlinear Mapping and a Related R-mode Technique for Compression of Multivariate Data

by

S. Henley[1]

Abstract

Nonlinear mapping is a method of projecting points from L-dimensional space into a lower space with optimal preservation of data structure as measured by the inter-point distance matrices. An analogous method can be used to study relationships among variables, using, for example, the correlation matrix. Two possible geological applications of the techniques are illustrated.

KEY WORDS: multivariate analysis, classification, pattern recognition, factor analysis.

[1] Bureau of Mineral Resources, Geology and Geophysics, Canberra, Australia; published by permission of the Director.

INTRODUCTION

A problem frequently encountered in geochemical studies is the interpretation of large volumes of multivariate data; some form of data compression must be used to make the results intelligible and assist in their evaluation.

Factor analysis has often been used in an attempt to solve this problem, by linear transformation of the data into a number of (usually) orthogonal factors which is less than the original number of variables, so as to account for a maximum of some property of the data matrix, such as the variance/coveriance relationships or the correlations. In a typical case, 25 variables might be reduced to 4 factors which are considered significant; inevitably there are also factors containing little information which are usually ignored, and the 'true' dimensionality of the data is taken subjectively as the number of major factors.

The factor matrix is still often difficult to interpret despite various rotations of the factor co-ordinate system, and plotting the factor loadings or scores can be very helpful. When there are more than two factors, however, the data structure cannot be represented in a single scatter diagram, and graphical aids to interpretation become much less useful.

NONLINEAR MAPPING

Sammon (1969) has described a different approach to data compression, abandoning the requirement for linear transformation and thereby allowing much more flexible representation of data structure in a plane diagram. As Sammon envisaged it, the method of Nonlinear Mapping (NLM) operates in the Q-mode, expressing relationships among observations and maximally preserving the Euclidean distance matrix on reduction of the number of dimensions. One could immediately see applications for this type of analysis in discriminating groups of geochemical samples from different environments (Howarth, 1972); a less obvious application might be in the classification of rocks on their chemical and other characters, on the lines proposed by Hubaux (1971).

If, for example, we take a set of chemical analyses of standard rock types (e.g. Wedepohl, 1969), and try to place an 'unknown' rock in its correct position, the NLM plot should show which are the rocks nearest to it in composition. The results of such an exercise are shown in Figs 1 to 3, where a set of igneous rock standards is plotted. The first 'unknown' (S, Fig. 2) is a typical spilite; the second (OL, Fig.3) is an olivine leucitite from Devon, England (Knill, 1969). This test is discussed later.

R-MODE DATA SQUASHING

Another possibility which suggests itself is the R-mode application of an NLM-type method to investigate relationships among variables. One cannot now use the distance matrix, but the product-moment correlation matrix is suitable, and in fact leads to simpler equations. This type of analysis has not yet been named, but it is proposed to call it 'squashing' (non-linear projection into a plane is what one might expect if one were to sit on a heap of multi-dimensional observations and squash it).

Both NLM and squashing analysis work by the generation of a triangular matrix as close as possible to the original triangular matrix derived from n-dimensional data, using only 2-dimensional matrix. The method of solution adopted by Sammon - an iterative, steepest descent method - is followed in both the programs (NLM and SQUASH) listed in Appendices C and D. The derivations of the algorithms are presented in Appendices A and B.

TEST 1 - ROCK CLASSIFICATION

It has been suggested (Hubaux, 1971) that a useful method of classifying rocks might be based on a series of standards. Ignoring, for the purposes of the test, the problems of scaling non-comparable units (petrographic, geochemical, physical properties....) and taking simply chemical analyses, we may construct a set of chemical standards of named rocks from the literature. The method proposed by Hubaux takes the three standards nearest to the unknown rock, that enclose it in a triangle, and the rock is expressed as x percent of standard 1, y percent of standard 2, and z percent of standard 3. Since it is a plane figure which is used to express a rock composition, NLM is an appropriate method to adopt, as it can reduce the dimensionality of a multivariate data matrix.

Points closest together will suffer least distortion, in general, and thus the method of NLM will usually be adequate to define the three closest standards. Having done this, one may use a linear programming approach (Wright and Doherty, 1971) to compute the relative proportions of the standard compositions required to make up the 'unknown' rock. Alternatively, one may compute these proportions directly from the distance matrix printed out by the NLM program.

In the first run, we have taken 17 analyses from Wedepohl (1969, p. 236-239), of named rock types, and performed NLM on the data (twelve oxides: SiO_2, TiO_2, Al_2O_3, Fe_2O_3, FeO, MnO, MgO, CaO, Na_2O, K_2O, P_2O_5, H_2O+). The plot (Fig.1) of the resulting two NLM co-ordinates shows a basic-acid axis from alkali basalt to alkali granite, with a number of outlying rocks of unusual composition (peridotite, olivine melilitite, etc.).

If we now wish to find the position of a typical spilite in this field of igneous rock compositions, we perform another NLM analysis, with our spilite analysis included. The plot (Fig.2) shows that the spilite composition lies closest tomdiorite and andesite, and not far from gabbro. The smallest triangle which contains the spilite is the diorite-andesite-ijolite triangle.

The third run (Fig.3) was an attempt to classify an olivine leucitite (Knill, 1969) according to the limited range of standards that were selected.

TEST 2 - GEOCHEMISTRY OF DEVONIAN SEDIMENTS

A set of major and trace element analyses of 96 samples of clastic sediments from Cornwall, England (Henley, 1970) was used to test the squashing algorithm. Results are plotted in Fig. 4, and a plot of the first two principal components is shown for comparison in Fig.5.

The squashing algorithm gives a picture very similar to the principal components diagram; differences are caused by the attempt of the squashing algorithm to present all the significant inter-variable relationships in a single two dimensional diagram. For a quick examination of a multivariate data matrix, therefore, squashing is more useful than principal components; it is also cheaper to compute.

PERFORMANCE OF THE NLM ALGORITHM

Sammon (1969) did not discuss the performance of the algorithm except in the most general terms. Points which require clarification, before use of the method in specific problems, are the stability of the iterative method, the optimal 'magic factor', the number of iterations required (and possible means of automating the selection of this number), and possible ways of controlling the algorithm to produce more stable and (if possible) faster convergence to minimum error.

(i) Stability Sammon mentions that care must be taken to prevent any two points becoming identical, in order to avoid 'blowing up' the partial derivatives. Even with close approach of two points, however, very large values may be obtained for the derivatives, which can temporarily drive one or both points a spuriously large distance out of the area of interest. When this happens, the error increases sharply and the total number of iterations required is much greater than it need be; a partial solution is discussed below in the section on methods of control.

A second point concerns the starting condition of the d-space. Sammon mentions two possible starting configurations: (a) random point distribution, and (b) using the d variables with highest variance. Theoretically either method will give perfectly adequate results; the second is used in NLM program, and the first in the squashing program. However, to avoid having to add a random number generator, a starting configuration is used which is fixed for every problem; only d = 2 is used. The FORTRAN integer divide operation is employed to define any point Y (i,1), Y(i,2) by

$$Y(i,1) = \mathrm{FLOAT}\ (i\ /\ 2)$$
$$Y(i,2) = \mathrm{FLOAT}((i-1)\ /\ 2),$$

giving a 'stepped' array of points. Either method will give a stable result in nearly every case: however, one must be aware that there is always a possibility of an incorrect solution being found by the points being 'locked' into a linear state.

(ii) The magic factor Sammon found the optional MF to lie between 0.3 and 0.4, and study of synthetic data appears to confirm this, with 0.35 perhaps being close to the best value. Lower values give slower convergence, while higher values cause each iteration to 'overshoot' the path of steepest descent. Either effect results in less efficient convergence.

(iii) Number of iterations For relatively simple synthetic data (cube and 4 dimensional hypercube) satisfactory convergence with MF = 0.35 took up to 30 iterations; more complex data (rock classification tests) required 40 - 50 iterations; though the time taken by each iteration is greater for problems with more points, the number of iterations required is not so variable; it depends more on the magic factor and on methods used to control the algorithm (thus in a large problem it is essential to find the best possible magic factor).

If one could define an objectively satisfactory solution, valid in all cases, it would be possible to cut off the analysis automatically. There are a number of ways by which one might attempt this:

(a) specifying a maximum acceptable error. This has the disadvantage that very many iterations could be required to reach it (or it may never be reached at all) - alternatively a solution with much lower error might be possible after relatively few more iterations.

(b) defining an absolute or relative reduction of error between successive iterations. This criterion could be useful if safeguards were incorporated to ensure that a stable (i.e. minimum error) solution had been attained: for example, one might check three or more successive iterations by this criterion. An absolute reduction of error would suffer the same drawbacks as (a) above; a relative reduction (e.g. 0.1%) would not.

(c) computing a fixed number of iterations after either criterion (a) or (b) has been satisfied, and re-check, to confirm that a stable solution has been reached.

(d) satisfaction of criteria (a) and (b) simultaneously.

Either (c) or (d) willprevent the possibility of the analysis being cut off at a high-error unstable configuration.

(iv) Control of the algorithm

(a) When large values of the partial derivatives tend to drive points too far, it can be useful to limit the distance to which they can be sent, by incorporating a check, reducing unacceptably high values of delta (f_1/f_2).

(b) It is possible that a variable magic factor give faster convergence: near the beginning a small MF gives sufficiently rapid error reduction (and a large MF could lead to large deviations from the optimal path), but later in the procedure a larger MF could give faster convergence. The equation for one possible variable MF function is

$$MF = 0.4 - 0.1 / \sqrt{M}$$

where M is the number of iterations.
This suggestion is not incorporated into the existing program, since reasonably satisfactory results have been obtained with a fixed MF = 0.35, but for problems with a large number of points a variable MF might result in significant savings.

(v) A method of improving stability and convergence

The program was first tested with new y values not being used until the start of the succeeding iteration, but it has been found that by using new y values as soon as they have been computed, very much faster convergence is obtained and the algorithm appears to be more stable. Further theoretical work requires to be done, however, on the mathematical basis of the NLM method. Meanwhile, the program exists in two versions, both of which give satisfactory results.

REFERENCES

HENLEY, S., 1970 - The geology and geochemistry of an area around Perranporth, Cornwall: PhD thesis, University of Nottingham.

HOWARTH, R.J., 1972 - Empirical discriminant classification of regional stream-sediment geochemistry in Devon and east Cornwall (discussion): Trans. Inst. Min. Metall., v. 81, B116-119.

HUBAUX, A., 1971 - Scheme for a quick description of rocks: Mathematical Geology, v. 3, p. 317-322.

KNILL, D.C., 1969 - The Permian igneous rocks of Devon: Bull. Geol. Surv. G.B. 29, 115-138.

SAMMON, J.W., 1969 - A nonlinear mapping for data structure analysis : IEEE Trans. on Computers, v. C-18, p. 401-409.

WEDEPOHL, K.H., 1969 - Composition and abundance of common igneous rocks: *in* Handbook of geochemistry, vol. 1, Springer-Verlag, Berlin, p. 227-249.

WRIGHT, T.L., and DOHERTY, P.C., 1960 - A linear programming and least squares computer method for solving petrologic mixing problems: Bull. Geol. Soc. Amer., v. 81, p. 1995-2008.

APPENDIX A - DERIVATION OF THE NLM ALGORITHM (after Sammon, 1969).

Given N vectors in an L-dimensional space, designated X_i, i=1, ..., N, let us define a corresponding N vectors in d-space ($d \ll L$), designated Y_i, i=1, ..., N.

Let the distance between vectors X_i and X_j be defined as

$$d^*_{ij} \equiv \text{dist}\ (X_i, X_j),$$

and the distance between corresponding vectors Y_i and Y_j in the d-space be defined as

$$d_{ij} \equiv \text{dist}\ (Y_i, Y_j).$$

Given an initial random d-space configuration for the Y vectors as follows:

$$Y_1 = \begin{bmatrix} y_{11} \\ \vdots \\ y_{1d} \end{bmatrix}, \ldots\ldots, Y_i = \begin{bmatrix} y_{i1} \\ \vdots \\ y_{id} \end{bmatrix}, \ldots\ldots, Y_N = \begin{bmatrix} y_{N1} \\ \vdots \\ y_{Nd} \end{bmatrix}$$

the error E, which represents how well the present configuration of the N points in d-space fits the structure of the N points in L-space, is given by

$$E = \frac{1}{\sum\limits_{i<j} d^*_{ij}} \sum_{i<j}^{N} \frac{(d^*_{ij} - d_{ij})^2}{d^*_{ij}}$$

Since the values of d_{ij} are defined by the values of y_{pq}, p = 1, ...,N and q = 1, . . . ,d, the next step is to adjust the values of y_{pq} so as to decrease the error. An iterative, steepest descent procedure outlined by Sammon (1969) is used for this purpose; the new d-space configuration after the mth iteration is given by

$$y_{pq}(m+1) = y_{pq}(m) - \text{MF}.\Delta_{pq}(m),$$

where $\Delta_{pq}(m) = \dfrac{\partial E(m)}{\partial y_{pq}\ (m)} \Bigg/ \left|\dfrac{\partial^2 E(m)}{\partial y_{pq}\ (m)^2}\right|$

and MF is the 'magic factor' or stepping coefficient which controls the rate and stability of convergence.

The partial derivatives are given by

$$\frac{\partial E}{\partial y_{pq}} = \frac{-2}{c} \sum_{\substack{j \neq p \\ j=1}}^{N} \left[\frac{d^*_{pj} - d_{pj}}{d^*_{pj} d_{pj}} \right] (y_{pq} - y_{jq})$$

and

$$\frac{\partial^2 E}{\partial y^2_{pq}} = \frac{-2}{c} \sum_{\substack{j \neq p \\ j=1}}^{N} \frac{1}{d^*_{pj} d_{pj}} \left[(d^*_{pj} - d_{pj}) - \frac{(y_{pq} - y_{jq})^2}{d_{pj}} \left(\frac{d^*_{pj}}{d_{pj}} \right) \right]$$

where

$$c = \sum_{i<j}^{N} d^*_{ij}.$$

If the distance between any two points y_{iq} and y_{jq} in the d-space becomes too small, the partial derivatives tend to infinity; thus in the program, if this distance falls to less than 0.05 at any stage, the points are separated by adding 0.1 to y_{i1} before entering the next iteration. If any differential is still large, the appropriate distance could be adjusted further by slight alteration of the program, though this has not been found necessary by the author.

APPENDIX B - DERIVATION OF THE SQUASHING ALGORITHM

Given L variables sampled by N observations, with values defined as x_{pq}, $p = 1, \ldots\ldots, L$ and $q = 1, \ldots\ldots, N$, let us define a corresponding set of L variables in a lower d-space (i.e. sampled by only d observations), with values y_{pq}, $p = 1 \ldots\ldots, L$ and $q = 1, \ldots, d$.

The correlation coefficient between the ith and jth variables, normalised to zero mean and unit variance, is

$$r^*_{ij} = \frac{1}{N} \sum_{k=1}^{N} x_{ik}\, x_{jk}$$

and in the d-space, the correlation coefficient is

$$r_{ij} = \frac{1}{d} \sum_{k=1}^{d} y_{ik}\, y_{jk}.$$

The error E which is to be minimised, can be defined as

$$E = \frac{1}{d(d-1)} \sum_{i<j}^{d} (r^*_{ij} - r_{ij})^2$$

(or the mean cell squared error - the root of this is the mean error in each cell of the correlation matrix).

An iterative, steepest descent procedure analogous to that used in nonlinear mapping may be adopted, with the new d-space oonfiguration after the mth iteration being given by

$$y_{pq}(m+1) = Y_{pq}(m) - MF.\Delta_{pq}(m)$$

where $$\Delta_{pq} = \frac{\partial E(m)}{\partial y_{pq}(m)} \Bigg/ \left| \frac{\partial^2 E(m)}{\partial y_{pq}(m)^2} \right|$$

and MF is the magic factor.

The partial derivatives are:

$$\frac{\partial E}{\partial y_{pq}} = 2 \sum_{i \neq p}^{d} y_{iq} (r_{ip} - r^*_{ip})$$

and $$\frac{\partial^2 E}{\partial y^2_{pq}} = 2 \sum_{i \neq p}^{d} y^2_{iq} \left(\frac{1}{d-1} \right)$$

There is no necessity to prevent points from becoming identical, since there is no danger of the derivatives 'blowing up'; the only requirement is that the starting configuration must contain at least two non-zero values of y_{pq} for each q.

CAPTIONS TO THE FIGURES

Figure 1. NLM projection of the silicate analyses of 17 rock standards from Wedepohl (1969) into a plane: a=alkali granite, b=quartz monzonite, c=granodiorite, d=quartz diorite, e=alkali syenite, f=syenite, g=monzonite, h=diorite, i=gabbro, j=andesite, k=tholeiitic basalt, l=alkali basalt, m=peridotite, n=olivine melilitite, o=anorthosite, p=phonolite, q=ijolite.

Figure 2. NLM projection of the silicate analyses of 17 rock standards and an 'unknown' spilite into a plane. Key to standards as in Fig. 1; S=spilite.

Figure 3. NLM projection of the silicate analyses of 17 rock standards and an 'unknown' olivine leucitite into a plane. Key to standards as in Fig. 1; OL=olivine leucitite.

Figure 4. Squashing analysis of 35 chemical variables measured on 96 clastic Devonian sediments from Cornwall, England; optimal plane representation of the correlation matrix.

Figure 5. Principal component analysis of 35 chemical variables measured on 96 clastic Devonian sediments from Cornwall, England; first two components plotted.

Fig 1

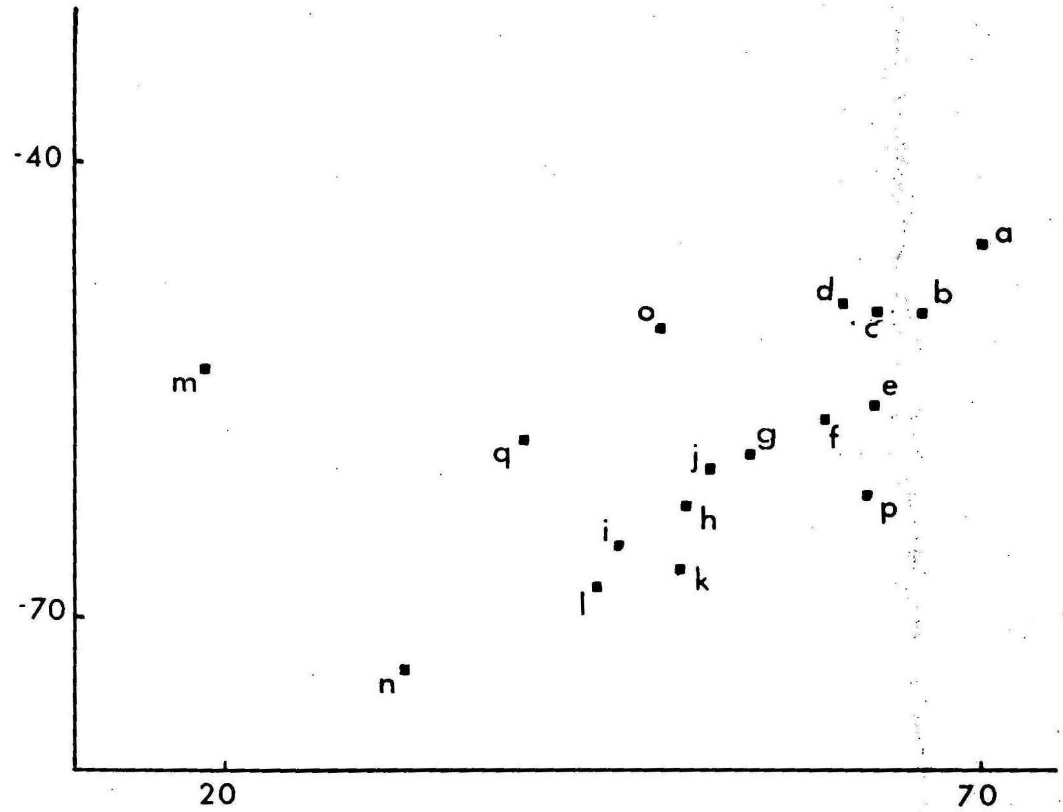


Fig 2

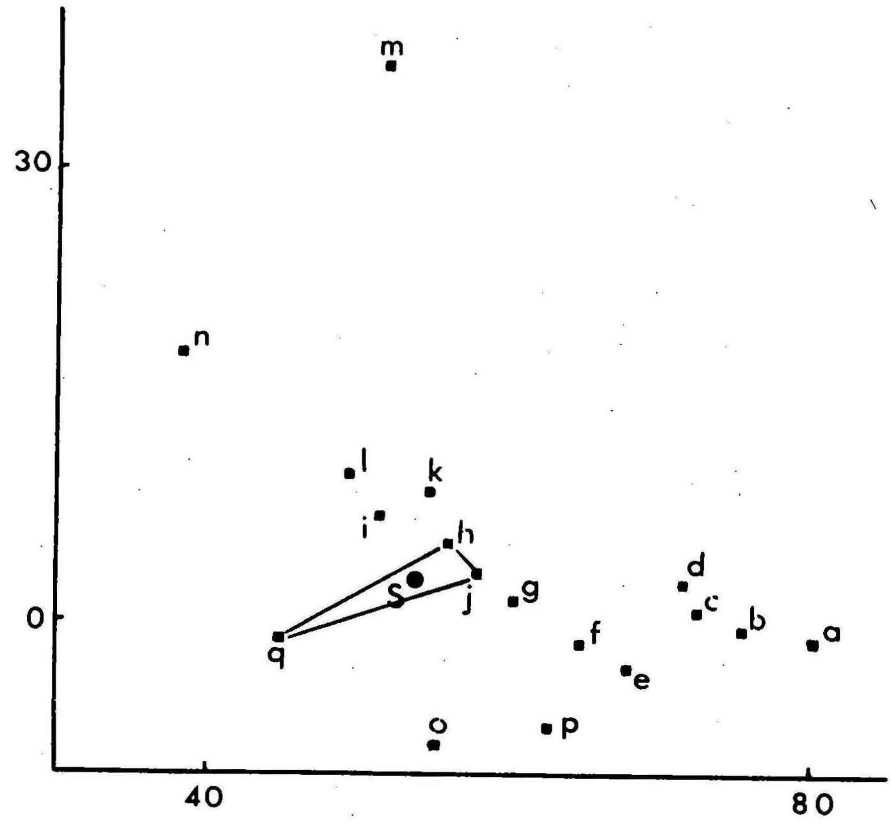


Fig 3

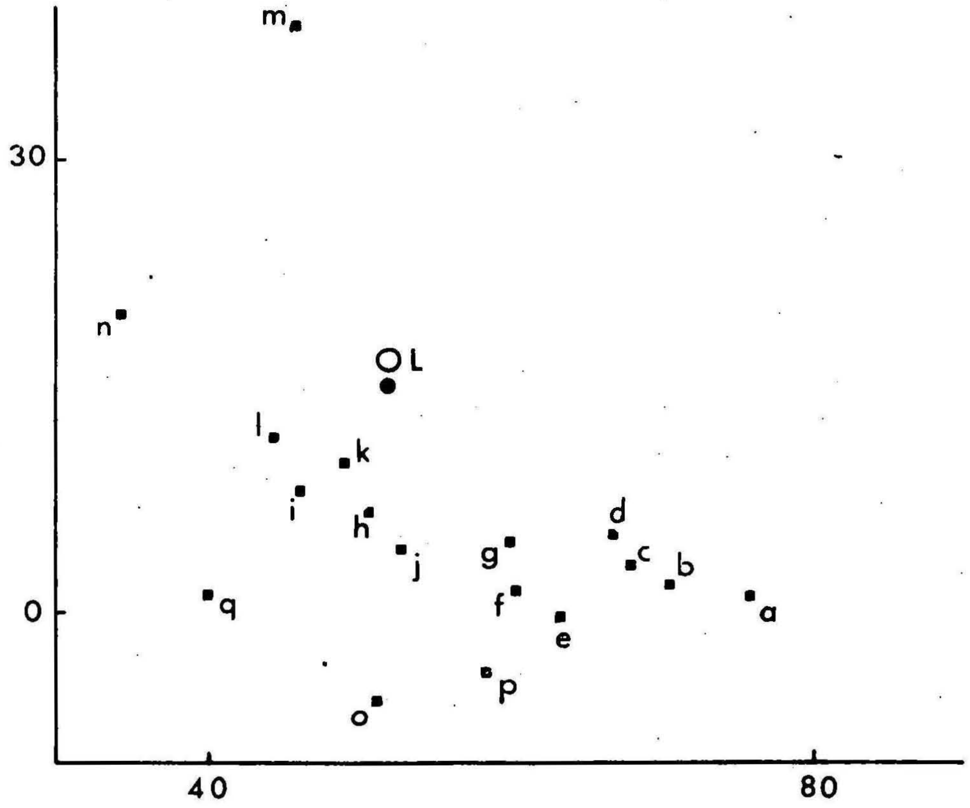


Fig 4

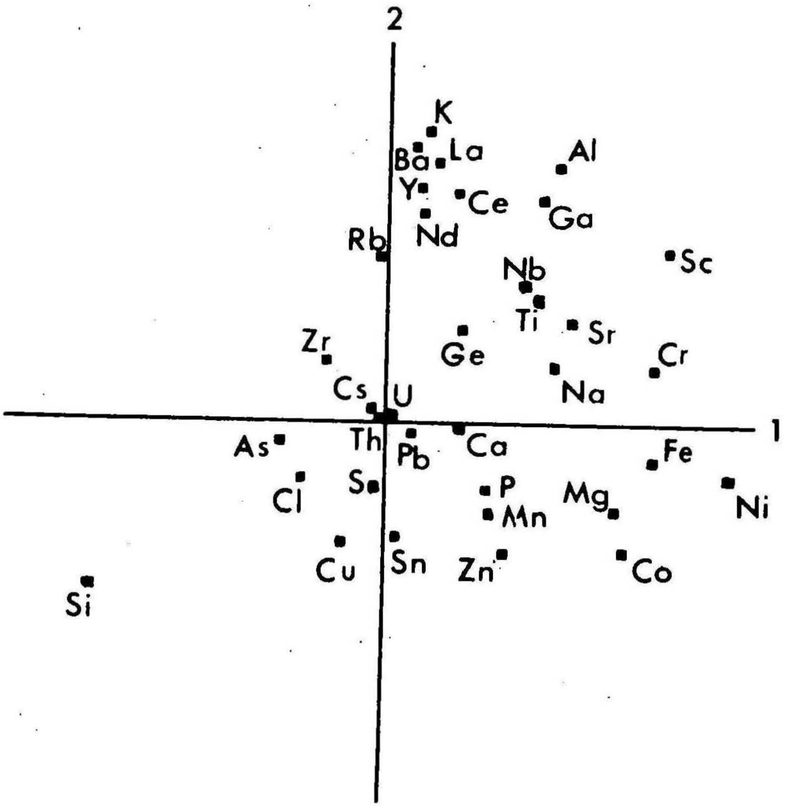
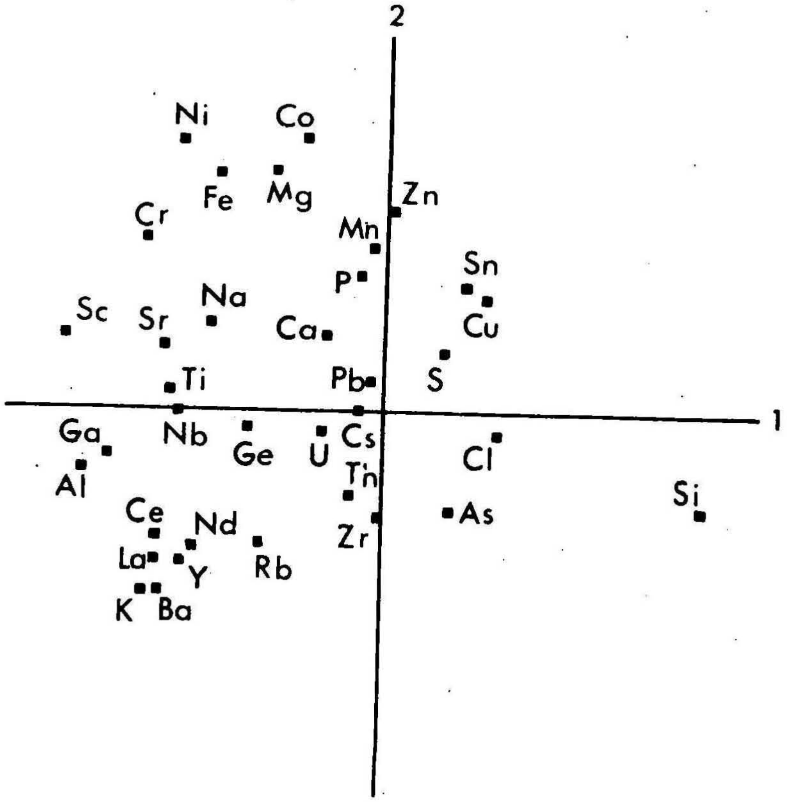

2
1
K
Ba
La
Al
Y
Ce
Ga
Nd
Rb
Sc
Nb
Ti
Sr
Zr
Ge
Cr
Cs
U
Na
As
Th
Pb
Ca
Fe
Cl
S
P
Mg
Ni
Mn
Cu
Sn
Zn
Co
Si

Fig 5



2
Ni
Co
Fe
Mg
Zn
Cr
Mn
Sn
P
Na
Sc
Sr
Ca
Cu
Ti
Pb
S
1
Ga
Nb
Ge
U
Cs
Cl
Al
Th
Si
Ce
Nd
As
Zr
La
Y
Rb
K
Ba

PENDIX C - LISTING OF PROGRAM NLM

```
      PROGRAM NLM

      NON-LINEAR MAPPING BASED ON THE ALGORITHM BY SAMMON, P,401 IEEE TRANS-
      -ACTIONS ON COMPUTING, 1969

      PROGRAMMED BY S.HENLEY,  1972

      REDUCES DIMENSIONALITY OF A DATA MATRIX BY NONLINEAR COMPRESSION OF DATA
      FROM M (LE, 50) DIMENSIONS INTO K (EQ.2) DIMENSIONS, WITH MAXIMAL
      PRESERVATION OF EUCLIDEAN INTERPOINT DISTANCES

      DISTANCE MATRICES STORED IN ARRAY DS(N,N)
      MATRIX D* IN LOWER HALF DS(I,J) WITH I,LT,J
      MATRIX D IN UPPER HALF WITH I,GT,J



      DATA CARDS - - -

          1    COLS 1-2       LOGICAL UNIT NO, OF DATA INPUT
                    3-4       NO. OF VARIABLES
                    11-20     'MAGIC FACTOR' (NORMALLY 0,3 OR 0,4)
                    21-22     NO, OF POPULATIONS IN DATA MATRIX SUPPLIED
                    23-24     NO. OF ITERATIONS TO BE CARRIED OUT
          2    COLS 1-80      INPUT FORMAT ENCLOSED IN BRACKETS, MUST INCLUDE 2 ALPHA
                              IDENTIFICATION FIELDS AT THE START OF EACH RECORD
          3    DATA CARDS IF ON UNIT 60

          4 END-OF-FILE CARD PUNCHED IEOF

          5    IF NEW MAGIC FACTOR DESIRED, ADDITIONAL CARDS (ANY NUMBER )
          CAN BE ADDED AT THE END OF THE DECK

      DIMENSION X(100,50),Y(100,2),YN(100,2),LETTER(10),FMT(10),VAR(50),
     .DS(100,100),ID(100,2),NUM(10,2),NTITLE(3),ER(200),EX(200)
      COMMON/222/ERR
      COMMON/10/DMAX
      COMMON/111/DS,X,Y,YN
      INTEGER FMT
      DATA (LETTER=1RA,1RB,1RC,1RD,1RD,1RE,1RF,1RG,1RH,1RI,1RJ)
      CALL Q8QINTOF
      N=0
      READ 10,LUN,L,FM,NPOP,NIT
      IF (NPOP.EQ.0) NPOP=1
   10 FORMAT (2I2,6X,F10,I2,I2)
      READ 20,FMT
   20 FORMAT (10A8)
      DO 30 I=1,NPOP
      NN=1
      IF(I,GT,1) NN = NUM(I-1,1)+1
      CALL READX(X,NN,L,ID,LUN,FMT,NEW)
      NUM(I,1)=NEW
      NUM(I,2)=LETTER(I)
   30 N=NEW
      DO 5 I=1,N
      DO 5 J=1,N
    5 DS(I,J)=0,0
      CALL VARN(X,N,L,VAR)
      VMAX=VMAX2=0.0
      DO 40 I=1,L
      IF (VMAX.LT.VAR(I)) 31,35
   31 VMAX=VAR(I)
```

```
      NVA=I
      GO TO 40
   35 IF (VMAX2.LT.VAR(I)) 36,40
   36 VMAX2=VAR(I)
   37 NVB=I
   40 CONTINUE
      PRINT 41,NVA,NVB
   41 FORMAT (*0VARIABLES WITH GREATEST VARIANCE,*,2I7,* , ARE USED AS *
     .,*INITIAL ESTIMATE*)
      PRINT 997,FM
   42 DO 50 I=1,N
      Y(I,1)=X(I,NVA)
   50 Y(I,2)=X(I,NVB)
      M=0
      CALL DSTAR (X,N,L,DS,M)
      M=1
      CALL DSTAR (Y,N,2,DS,M)
 1000 CALL ERROR (DS,N,M,C)
      ER(M)=ERR $ EX(M)=FLOATF(M)
      IF (M.EQ.NIT) GO TO 990
      CALL YNEW (DS,N,Y,2,C,YN,FM)
      DO 60 I=1,N
      Y(I,1)=YN(I,1)
   60 Y(I,2)=YN(I,2)
      CALL DSTAR(Y,N,2,DS,M)
      M=M+1
      GO TO 1000
  990 J=1
      NTITLE(1)=8HNEW DIST
      NTITLE(2)=8HANCE MAT
      NTITLE(3)=8HRIX
      CALL TMPRINT(DS,N,M,NTITLE,3)
      CALL QUIKPLOT(EX,ER,M,1,12H*ITERATIONS*,7H*ERROR*)
      PRINT 991
  991 FORMAT(1H1)
      DO 994 I=1,N
      IF (I.LE.NUM(J,1)) GO TO 992
      J=J+1
  992 PRINT 993, ID(I,1),ID(I,2),Y(I,1),Y(I,2),NUM(J,2)
  993 FORMAT (10X,A8,A3,10X,2F12.4,10X,1R1)
  994 PUNCH 993, ID(I,1),ID(I,2),Y(I,1),Y(I,2),NUM(J,2)
      READ 10,IDUM,IDUM,FM
      IF (EOF,60) 999,998
  998 PRINT 997,FM
  997 FORMAT (*1MAGIC FACTOR IS *,F8.3)
      GO TO 42
  999 CALL EXIT
      END
      SUBROUTINE READX(X,NN,L,ID,LUN,FMT,NEW)
      DIMENSION X(100,50),FMT(10),ID(100,2)
      INTEGERFMT
      I=NN
      PRINT 60,FMT
   60 FORMAT(*0 INPUT FORMAT IS *,10A8)
   10 READ (LUN,FMT) ID(I,1),ID(I,2), (X(I,J),J=1,L)
      IF (EOF,LUN) 30,20
   20 PRINT 15,ID(I,1),ID(I,2), (X(I,J),J=1,L)
      I=I+1
   15 FORMAT(1H0,2R8,4X,10F10.4,9(/1H ,10F10.4))
      GO TO 10
   30 NEW=I-1
      PRINT 40,NEW
   40 FORMAT (*0READX COMPLETED, N=*,I4)
```

```
      RETURN
      END
      SUBROUTINE VARN(X,N,L,VAR)
      DIMENSION X(100,50),VAR(50)
      PRINT 1
    1 FORMAT (*0VARN ENTERED*)
      DO 30 I=1,L
      SUM=0,0
      DO 10 J=1,N
   10 SUM=SUM+X(J,I)
      AVE=SUM/FLOAT(N)
      SUM=0,0
      DO 15 J=1,N
      DIF = X(J,I)-AVE
   15 SUM=SUM+ DIF**2
      VAR(I)= SUM/FLOAT(N-1)
   30 PRINT 25,I,VAR(I),AVE
   25 FORMAT (10X,I10,2F20.4)
      RETURN
      END
      SUBROUTINE DSTAR(X,N,L,DS,M)
      DIMENSION X(100,L),DS(100,100),NTITLE(3)
      COMMON/10/DMAX
      IF (M,GT.0) GO TO 10
      DMAX=0.0
      IB=1
      IT=N-1
      M=0
      NTITLE(1)=8HINITIAL
      NTITLE(2)=8HDISTANCE
      NTITLE(3)=8H MATRIX
      GO TO 20
   10 IB=2
      IT=N
   20 DO 50 I=IB,IT
      IF (M,GT.0) GO TO 30
      JB=I+1
      JT=N
      GO TO 40
   30 JB=1
      JT=I-1
   40 DO 50 J=JB,JT
      SUM=0,0
      DO 45 K=1,L
   45 SUM=SUM+(X(I,K)-X(J,K))**2
      DS(I,J)=SQRT(SUM)
      IF (M,GT.0)  47,50
   47 CALL CHECK (DS,N,I,J,M,X)
   50 CONTINUE
      IF (M,EQ.0) CALL TMPRINT(DS,N,M,NTITLE,3)
      RETURN
      END
      SUBROUTINE TMPRINT(DS,N,M,NTITLE,K)
      DIMENSION DS(100,100),NTITLE(K)
      PRINT 5, NTITLE
    5 FORMAT (1H1,10A8)
      PRINT 45
      DO 50 I=2,N
      JT=I-1
      IF (M,GT.0) 10,20
   10 PRINT 40, (DS(I,J),J=1,JT)
      GO TO 30
   20 PRINT 40,(DS(J,I),J=1,JT)
```

```
30 IF (JT,GT,16) PRINT 45
40 FORMAT (1H ,16F8,2)
45 FORMAT(1H )
50 CONTINUE
   PRINT 45
   RETURN
   END
   SUBROUTINE ERROR (DS,N,M,C)
   DIMENSION DS(100,100)
   COMMON/222/E
   C=S=0,0
   DO 50 J=2,N
   JM=J-1
   DO 50 I=1,JM
   C=C+DS(I,J)
   Q=(DS(I,J)-DS(J,I))**2
50 S=S+Q/DS(I,J)
   E=S/C
   PRINT 60,M,E
60 FORMAT (* ERROR AFTER THE *,I4* TH ITERATION IS*,E12,2)
   RETURN
   END
   SUBROUTINE YNEW(DS,N,Y,K,C,YN,FM)
   DIMENSION DS(100,100),Y(100;2),YN(100,2)
   COMMON/10/DMAX
   INTEGER P,Q
   DO 50 P=1,N
   DO 50 Q=1,K
   SUMA=SUMB=0,0
   DO 40 J=1,N
   IF (J-P) 22,40,21
21 DSPJ=DS(P,J)
   DPJ =DS(J,P)
   GO TO 23
22 DSPJ=DS(J,P)
   DPJ=DS(P,J)
23 YPQ =Y(P,Q)
   YJQ =Y(J,Q)
   YDIF=YPQ-YJQ
   DDIF=DSPJ-DPJ
   DPROD=DSPJ*DPJ
   SUMA=SUMA+(DDIF/DPROD)*YDIF
   B= DDIF -(YDIF**2/DPJ)*(1.0+DDIF/DPJ)
   SUMB=SUMB+B/DPROD
40 CONTINUE
45  DELTA=-SUMA/ABS(SUMB)
   YN(P,Q)=Y(P,Q)-FM*DELTA
50 Y(P,Q)=YN(P,Q)
   RETURN
   END
   SUBROUTINE CHECK(DS,N,I,J,M,Y)
   DIMENSION DS(100,100),Y(100;2)
   IF (DS(I,J),GT.0,005) RETURN
   Y(I,1)=Y(I,1)+0,01
   DS(I,J)=SQRT((Y(I,1)-Y(J,1))**2 + (Y(I,2)-Y(J,2))**2)
   RETURN
   END
```

PPENDIX D - LISTING OF PROGRAM SQUASH

```
      PROGRAM SQUASH
C
C   R-MODE NLM SIMILAR TO THE ALGORITHM BY SAMMON, P.401 IEEE TRANS-
C   -ACTIONS ON COMPUTING, 1969
C
C   PROGRAMMED BY S.HENLEY, AUGUST 1972
C
C   REDUCES DIMENSIONALITY OF A DATA MATRIX BY NONLINEAR COMPRESSION OF DATA
C   WITH MAXIMAL PRESERVATION OF CORRELATION COEFFICIENTS
C
C
C   CORRELATION MATRICES STORED IN ARRAY DS(N,N)
C   MATRIX D* IN LOWER HALF DS(I,J) WITH I.LT.J
C   MATRIX D IN UPPER HALF WITH I.GT.J
C
C
C
C   DATA CARDS - - -
C
C      1    COLS 1-2      LOGICAL UNIT NO. OF DATA INPUT
C                3-4      NO. OF VARIABLES
C                11-20    'MAGIC FACTOR' (NORMALLY 0.3 OR 0.4)
C                23-24    NO. OF ITERATIONS TO BE CARRIED OUT
C      2    COLS 1-80     INPUT FORMAT ENCLOSED IN BRACKETS, MUST INCLUDE 2 ALPHA
C                         IDENTIFICATION FIELDS AT THE START OF EACH RECORD
C     3   DATA CARDS IF ON UNIT 60
C
C      4 END-OF-FILE CARD PUNCHED IEOF
C
C      5    IF NEW MAGIC FACTOR DESIRED, ADDITIONAL CARDS (ANY NUMBER )
C      CAN BE ADDED AT THE END OF THE DECK
C
      DIMENSION X(50,100),Y(50,2),YN(50,2),FMT(10),VAR(50),
     .DS(50,50),ID(100,2),ER(200),EX(200),NTITLE(2)
      COMMON/222/ERR
      INTEGER FMT
      CALL Q8QINTOF
      N=0
      READ 10,LUN,L,FM,NIT
   10 FORMAT (2I2,6X,F10,2X,I2)
      READ 20,FMT
   20 FORMAT (10A8)
      CALL SREADX(X,L,ID,LUN,FMT,N)
      DO 5 I=1,L
      DO 5 J=1,L
    5 DS(I,J)=0.0
      CALL SVARN(X,N,L,VAR)
      VMAX=VMAX2=0.0
      PRINT 997,FM
   42 DO 50 I=1,L
      Y(I,1)=FLOAT(I/2)*2.0/FLOAT(L)
   50 Y(I,2)=FLOAT((I-1)/2)*2.0/FLOAT(L)
      M=0
      CALL SDSTAR (X,N,L,DS,M)
      M=1
      CALL SDSTAR (Y,2,L,DS,M)
 1000 CALL SERROR (DS,L,M,C)
      ER(M)=ERR $ EX(M)=FLOATF(M)
      IF (M.EQ.NIT) GO TO 990
      CALL SYNEW (DS,L,Y,2,C,YN,FM)
      DO 60 I=1,L
      Y(I,1)=YN(I,1)
   60 Y(I,2)=YN(I,2)
      CALL SDSTAR(Y,2,L,DS,M)
      M=M+1
      GO TO 1000
  990 CALL QUIKPLOT(EX,ER,M,1,12H*ITERATIONS*,7H*ERROR*)
      NTITLE(1)=8HNEW CORR
      NTITLE(2)=8HMX
      CALL TMPRINT (DS,L,M,NTITLE,2)
      PRINT 991
  991 FORMAT(1H1)
      DO 994 I=1,L
  992 PRINT 993,     Y(I,1),Y(I,2)
  993 FORMAT(10X,2F10.4)
  994 PUNCH 993,     Y(I,1),Y(I,2)
      READ 10,IDUM,IDUM,FM
      IF (EOF,60) 999,998
  998 PRINT 997,FM
  997 FORMAT (*1MAGIC FACTOR IS *,F8.3)
      GO TO 42
  999 CALL EXIT
      END
      SUBROUTINE TMPRINT(DS,N,M,NTITLE,K)
      DIMENSION DS(50,50),NTITLE(K)
      PRINT 5,NTITLE
    5 FORMAT(1H1,10A8)
      PRINT 45
      DO 50 I=2,N
      JT=I-1
      IF (M,GT.0) 10,20
   10 PRINT 40,(DS(I,J),J=1,JT)
      GO TO 30
   20 PRINT 40,(DS(J,I),J=1,JT)
   30 IF (JT.GT.16) PRINT 45
   40 FORMAT (1H ,16F8.2)
   45 FORMAT (1H )
   50 CONTINUE
      PRINT 45
      RETURN
      END
      SUBROUTINE SREADX(X,L,ID,LUN,FMT,NEW)
      DIMENSION X(50,100),FMT(10),ID(100,2)
      INTEGERFMT
      I=1
      PRINT 60,FMT
   60 FORMAT(*0 INPUT FORMAT IS *,10A8)
   10 READ (LUN,FMT) ID(I,1),ID(I,2), (X(J,I),J=1,L)
      IF (EOF,LUN) 30,20
   20 PRINT 15,ID(I,1),ID(I,2), (X(J,I),J=1,L)
      I=I+1
   15 FORMAT(1H0,2R8,4X,10F10.4,9(/21X,10F10.4))
      GO TO 10
   30 NEW=I-1
      PRINT 40,NEW
   40 FORMAT (*0READX COMPLETED, N=*,I4)
      RETURN
      END
      SUBROUTINE SVARN(X,N,L,VAR)
      DIMENSION X(50,100),VAR(50)
      DO 30 I=1,L
      SUM=0.0
      DO 10 J=1,N
   10 SUM=SUM+X(I,J)
      AVE=SUM/FLOAT(N)
      SUM=0.0
      DO 15 J=1,N
      DIF = X(I,J) = X(I,J)-AVE
   15 SUM=SUM+ DIF**2
      VAR(I)= SUM/FLOAT(N-1)
      ST=SQRT(VAR(I))
      DO 35 J=1,N
   35 X(I,J)=X(I,J)/ST
   30 PRINT 25,I,VAR(I),AVE
   25 FORMAT (10X,I10,2F20.4)
      PRINT 40
   40 FORMAT (1H0)
      DO 50 J=1,N
   50 PRINT 60,J,(X(I,J),I=1,L)
   60 FORMAT (1H ,I10,10F10.4,9(/11X,10F10.4))
      RETURN
      END
      SUBROUTINE SDSTAR(X,N,L,DS,M)
      DIMENSION X(50,N),DS(50,50),NTITLE(2)
      IF (M.GT.0) GO TO 10
      IB=1
      IT=L-1
      M=0
      GO TO 20
   10 IB=2
      IT=L
   20 DO 50 I=IB,IT
      IF (M.GT.0) GO TO 30
      JB=I+1
      JT=L
      GO TO 40
   30 JB=1
      JT=I-1
   40 DO 50 J=JB,JT
      SUM=0.0
      DO 45 K=1,N
   45 SUM=SUM+X(I,K)*X(J,K)
      DS(I,J)=SUM/FLOAT(N-1)
   50 CONTINUE
      NTITLE(1)=8HINITIAL
      NTITLE(2)=8HCORR MX
      IF (M.EQ.0) CALL TMPRINT (DS,L,M,NTITLE,2)
      RETURN
      END
      SUBROUTINE SERROR (DS,N,M,C)
      DIMENSION DS(50,50)
      COMMON/222/E
      C=S=0.0
      Q=0.0
      DO 50 J=2,N
      JM=J-1
      DO 50 I=1,JM
   50 Q=Q+(DS(I,J)-DS(J,I))**2
      S=FLOAT(N)*FLOAT(N-1)/2.0
      E=Q/S
      PRINT 60,M,E
   60 FORMAT (* MEAN CELL ERROR AFTER THE *,I4* TH ITERATION IS*,E12.2)
      RETURN
      END
      SUBROUTINE SYNEW(DS,N,Y,K,C,YN,FM)
      DIMENSION DS(50,50),Y(50,2),YN(50,2)
      INTEGER P,Q
      DO 50 P=1,N
      DO 50 Q=1,K
      F2=F=0.0
      DO 40 I=1,N
      IF (I-P) 20,40,30
   20 DSIP=DS(I,P)
      DIP=DS(P,I)
      GO TO 31
   30 DSIP=DS(P,I)
      DIP=DS(I,P)
   31 SUM=0.0
      SUM=DIP-DSIP
      F=F+SUM*Y(I,Q)
      F2=F2+Y(I,Q)**2/FLOAT(K-1)
   40 CONTINUE
      DELTA=F/ABS(F2)
   50 YN(P,Q)=Y(P,Q)-FM*DELTA
      RETURN
      END
```