COMMONWEALTH OF AUSTRALIA

DEPARTMENT OF
NATIONAL DEVELOPMENT

BUREAU OF MINERAL
RESOURCES, GEOLOGY
AND GEOPHYSICS

Record 1972/82

GEOCHEMICAL APPLICATIONS OF LINEAR PROGRAMMING

by

S. Henley[1]

[1] Bureau of Mineral Resources, Geology and Geophysics, Canberra, ACT, Australia

Record 1972/82

GEOCHEMICAL APPLICATIONS OF LINEAR PROGRAMMING

by

S. Henley[1]

[1]Bureau of Mineral Resources, Geology and Geophysics,
Canberra, ACT, Australia

CONTENTS

SUMMARY

Rock compositions may be expressed in terms of non-negative proportions of end-members by use of linear programming methods, in which one maximises the sum of these proportions; linear programming may in many cases be a preferable alternative to both Q-mode factor analysis (in which end-member proportions are not constrained to non-negative values) and traditional norm computation (where linear programming has the advantage of generality).

A general purpose computer program has been written to do this, with an option to identify, iteratively, the most divergent rock compositions from within the data matrix, and include them as additional end-members.

KEY WORDS: linear programming, geochemistry, norm computation, factor analysis.

INTRODUCTION

Linear programming is used to optimize linear functions where the variables are subject to a number of linear constraints (inequalities). In general, one seeks to maximize z, given by the relation:

$$z = c_1x_1 + c_2x_2 + c_3x_3 + ... + c_nx_n,$$

subject to the constraints $x_i \geqslant 0$ and

$$a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n \leqslant b_1$$

$$a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n \leqslant b_2$$

.....

$$a_{m1}x_m + a_{m2}x_2 + ... + a_{mn}x_n \leqslant b_m \quad ,$$

where a_{ij}, b_i, and c_j are given constants (Harbaugh & Bonham-Carter, 1970).

Frequently in geochemical and petrological studies, we wish to characterize a rock as a mixture of 'end-member' compositions, which may be idealized mineral compositions, as in norm computation, or extreme rock compositions or hypothetical rock compositions as in Q-mode factor analysis (Imbrie, 1963). In each case, a number of constraints are implied in the model. If one considers a chemical analysis of m elements, of which the ith element gives a percentage value p_i, and one wishes to split the observed composition into proportions x_j of n end-members, each of composition given by a vector of percentage values a_{ij}, the situation will fit the general case by substitution p for b, and setting $c_j = 1$. Each constraint,

$$a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n \leqslant p_i,$$

has the simple meaning that the sum of contributions to each variable by each end-member cannot exceed the observed value of that variable: i.e., end-member scores must be non-negative, and the 'error' term, given by

$$e_i = p_i - \sum_{j=1}^{n} a_{ij}x_j$$

must also be non-negative - This contrasts with normal multivariate techniques, such as least-squares optimization, where the error term is allowed to take negative values. There is one additional constraint, since

$$1x_1 + 1x_2 + \ldots + 1x_n \leqslant 1.0,$$

the sum of proportions of end-members cannot exceed 1. Normally, with linearly independent end-members, this constraint is superfluous and may be ignored, but in badly formulated problems containing redundant variables, and in norm calculations involving extensive mineral lists, it would be advisable to include this constraint in the formulation of the problem.

SIMPLEX SOLUTION OF LINEAR PROGRAMMING PROBLEMS

This method is described in practical detail by Harbaugh & Bonham-Carter (1970) and, as the explanations involved are rather complicated the reader is commended to their excellent treatment, and also to the account by Vajda (1961). Harbaugh & Bonham-Carter present a FORTRAN computer program for solution of linear programming problems, and this has been modified (Appendix 1) for application to problems of rock chemistry.

TWO-COMPONENT EXAMPLE

Consider that one has obtained a number of chemical analyses within the K-Al-Si-O system and wishes to express the data in terms of quartz and kaliophyllite. Two end-member vectors could be set up as follows:

	Q	Kp
Si	33.33	14.29
Al	0	14.29
K	0	14.29
O	66.67	57.14

A sample with a composition corresponding to orthoclase could then be allotted a position between the two, and expressed in terms of proportions of Q and Kp. Since negative proportions are disallowed, one may write the following inequalities:

$$33.33p_q + 14.29p_{kp} \leqslant 23.08 \quad (Si)$$
$$0p_q + 14.29p_{kp} \leqslant 7.69 \quad (Al)$$
$$0p_q + 14.29p_{kp} \leqslant 7.69 \quad (K)$$
$$66.67p_q + 57.14p_{kp} \leqslant 61.65 \quad (O)$$

The sum of p_q and p_{kp} cannot exceed 1, and thus one may write a further inequality:

$$1p_q + 1p_{kp} \leqslant 1$$

By application of the simplex method to maximize the objective function,

$$z = p_q + p_{kp},$$

one immediately finds that $z_{max} = 1$, $p_q = 0.40$ and $p_{kp} = 0.60$. Since only two end-members are involved, this solution can be represented graphically in two-dimensional space (Fig. 1), which may clarify a little the principles of linear programming.

NORM COMPUTATION

The essence of norm calculation is the transformation of a chemical analysis vector by a mineral composition matrix to a vector of mineral proportions. Expressed in these terms, it becomes somewhat analogous to factor analysis, since in both methods one is attempting to transform a set of observations into a set of more readily interpretable variables. Computer programs for norm computation do not normally reflect this theoretical simplicity, however, being generally rather long and untidy, due to the many 'special cases' which must be allowed for (e.g., the problem of partitioning iron and magnesium among the femic minerals). Most of the constraints in norm computation are linear, and thus one may use linear programming methods of solution. In simple cases, where partitioning problems do not arise and mineral compositions are linearly independent, the general model will invariably give correct results, but attempts to apply it to more complex problems may lead either to incorrect results, or to cyclic solutions (Beale, 1954), in which there is no decision among several incorrect results.

When pairs of solid solution series are involved, it is possible to set up additional constraints concerning them. To consider the case of coexisting orthopyroxene and olivine, let the molar percentages of $MgSiO_3$ and $FeSiO_3$ be x_{en} and x_{fs} respectively, and the percentages of Mg_2SiO_4 and Fe_2SiO_4 be x_{fo} and x_{fa}. Also let the ratios of Mg to Fe in the two minerals be equal. Then

$$x_{en}/x_{fs} = x_{fc}/f_{fa} = r,$$

where r is the ratio Mg/Fw.

This equality may be expressed as two constraints:

$$x_{en} - rx_{fs} = 0$$

$$x_{fo} - rx_{fa} = 0$$

If olivine and pyroxene are the only normative minerals containing iron and magnesium, r may be computed beforehand and these constraints applied directly (by use of the 'Big M' method - Harbaugh & Bonham-Carter, 1970). However, the presence of any minerals in which iron and magnesium are treated separately (e.g. pyrite, dolomite) invalidates this approach, as the proportions of residual Fe and Mg cannot be determined beforehand. It is possible to remove r from the constraints by a rearrangement of the equations, but the resulting constraint,

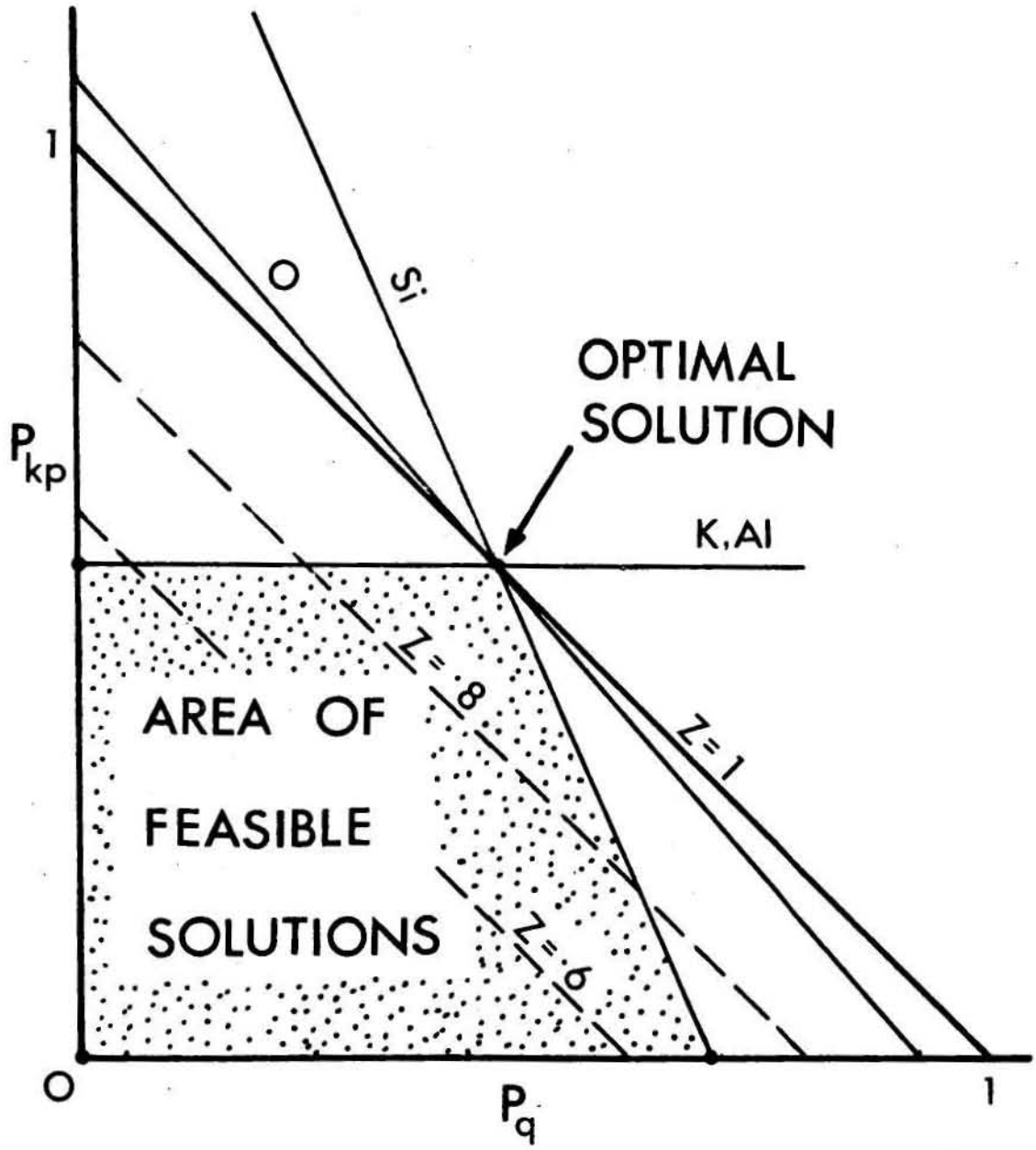
$$x_{en}x_{fa} = x_{fs}x_{fo},$$

is nonlinear, and thus linear programming methods can no longer be used.

If such a case arises, it might be possible to remove the constituents containing Fe and Mg alone or in fixed ratios first, then compute r and carry out the main part of the norm computation. If this is not possible, one could resort to more sophisticated programming methods (e.g., dynamic programming), or return to the specialised forms of norm computation. Despite this disadvantage of such a generalised method, its great advantage over the traditional types of norm computation lies in the variety of mineral end-members which one may incorporate into the problem without alteration of the computer program.

FIGURE 1. Representation of orthoclase as molar proportions of quartz (q) and kaliophyllite (kp). The sum of p_q and p_{kp} cannot exceed 1; thus the objective function $z = p_q + p_{kp} \leqslant 1$ is one constraint. Additional constraints are imposed by the relative proportions of the analysed elements in the minerals. e.g. the sum of $p_q Si_q$ and $p_{kp} Si_{kp}$ cannot exceed $Si_{or} = 23.08$.

These constraints, together with the requirement that all values be non-negative, define a field (stippled) of feasible solutions. The optimal solution is that in which $z = p_q + p_{kp}$ is maximised.



To accompany Record 1972/82

M(G) 215

LINEAR PROGRAMMING AND FACTOR ANALYSIS

The factor analysis model (Harman, 1967) may be defined as:

$$z_{ji} = \sum_{p=1}^{m} a_{jp} F_{pi} + d_j U_{ji},$$

where z is the value of variable j for individual or sample i, and each of the m terms $a_{jp} F_{pi}$ represents the contribution of the corresponding factor to the value of z_{ji}, while $d_j U_{ji}$ is the 'residual error' in the theoretical representation of z_{ji}. The F's and U's are usually taken to be linearly independent, and are assumed to have zero means and unit variances; thus in any individual sample, the term containing U could take a negative value, so that no physical significiance can be ascribed to the solution unless (as is often the case) one assumes that U_{ji} approximates to zero.

By re-definition of all the components of a, F, and U as non-negative (a reasonable definition in view of the fact that we are dealing with a set of non-negative observations), one may re-write the model as an inequality:

$$z_{ji} \geqslant \sum_{p=1}^{m} a_{jp} F_{pi} ,$$

leading to a set of constraints in a model in which one seeks to meximize $\sum_{p=1}^{m} F_{pi}$.

PSEUDO FACTOR ANALYSIS

Q-mode factor analysis, as carried out by Imbrie (1963), involves the selection of the most divergent rock compositions from a data matrix, followed by computation of the relative proportions (scores) of these hypothetical end-members in each sample.

Definition of the most divergent compositions is an area in which linear programming appears at first signt to be of little relevance. However, one may adopt an iterative approach, by first selecting a single end-member, in which, for example, the variable of highest variance (e.g. silica) has

maximum (or minimum) value. Applying linear programming methods to a set of analyses, one may now select the sample with poorest fit to the chosen composition, to become the second end-member. And so the process might continue, obtaining as many end-members as desired, to a maximum of m (the number of variables).

A computer program has been developed to carry out this type of analysis, and a listing, together with a worked example, is included in Appendices B and C. Operating instructions are given in Appendix A.

LIMITATIONS

Apart from the limitations of the method with respect to norm calculation, discussed above, linear programming suffers perhaps by being principally a deterministic technique, though Vajda (1961) has discussed 'stochastic linear programming' in which one assumes that the constants (or 'technological coefficients' as they are termed by Vajda) are random variables.

The main difficulties in introducing random variables concern the estimation of type and parameters of the distribution function of the objective function optimum. The method used in the present paper has been based on Madansky's (1960) 'Wait-and-See' principle, by which one does not try to estimate the distribution parameters, but simply takes the available observation (or waits for an observation) and solves the non-stochastic linear programming problem resulting.

POSSIBLE APPLICATIONS

The use of linear programming techniques has the virtue of generality, compared with classical methods of norm computation: empirical mineral compositions may be slotted in as required, without the necessity of writing a special computer program for each.

Further uses might be seen in the realms of experimental geochemistry (construction of phase diagrams, etc.), and in studies of magmatic differentiation, metasomatism, sedimentology, and ore-deposit paragenesis.

An interesting application might be the calculation of ternary diagram co-ordinates. Apart from the conventional triangular diagram, one could also envisage a diagram with two components plus an 'unknown' or 'remainder' component. In Fig. 1, for example, one might plot the optimal points for a number of samples in the trangle defined by 1.0q, 1.0kp, and the origin, which represents simply zero quartz, plus zero kaliophyllite, plus 100% of something else. Such a plot could be useful in studies of contamination of igneous rocks, or in economic geochemical surveys (analogous to the communality maps recommended by Krumbein & Graybill, 1965).

ACKNOWLEDGMENTS

The ideas developed in this paper were inspired by Harbough & Bonham-Carter (1971). The CSIRO CDC 3600 computer was used for development of the program.

REFERENCES

BEALE, E.M.L., 1955 - Cycling in the Dual Simplex Algorithm: Nav. Res. Log. Qu., 2, 269-75.

HARBAUGH, J.W., and BONHAM-CARTER, G., 1971 - Computer Simulation in Geology. N.Y., Wiley-Interscience. 575 p.

HARMAN, H.H., 1967 - Modern Factor Analysis. Chicago. University of Chicago Press, 474 p. (2nd edition).

IMBRIE, J., 1963 - Factor and vector analysis programs for analysing geologic data. ONR Geog. Br. tech. Rep. 6, ONR Task 389-135.

KRUMBEIN, W.C., and GRAYBILL, F.A., 1965 - An Introduction to Statistical Models in Geology. N.Y., McGraw-Hill, 475 p.

MADANSKY, A., 1960 - Inequalities for Stochastic Linear Programming Problems. Management Science, 6, 197-204.

VAJDA, S., 1961 - Mathematical Programming. Reading (Mass.), Addison-Wesley, 310 p.

APPENDIX 1

PROGRAM ENDMEM : OPERATING INSTRUCTIONS, LISTING, AND SAMPLE OUTPUT.

1) Operating instructions

The data input format has been kept as simple as possible, but the user can easily include additional options if desired.

Card 1: Variable format card for input of analytical data. It consists of a FORTRAN format definition, in brackets, allowing an initial alphanumeric field for identification, e.g. (A8, 2X, 14F4.2).

Card 2: Columns 1-3 (right-justified) M, the number of variables.

4-6 (right-justified) N, the number of end-members supplied.

N may be punched as a negative number, in which case -N end-members are supplied, and additional end-members are extracted from the data matrix until every sample composition is accounted for to at least TOL percent (set at 95 at present) by linear combinations of the end-members.

Cards 3 to N + 2 (or -N + 2): End-member compositions in the format specified on card 1.

Cards N + 3 (or -N + 3) onward: Data cards in the format specified on card 1.

Last card: End-of-file marker; format of this card will vary with the computer system; the CDC3600 format is $^{8}_{7}$EOF.

```
      PROGRAM ENDMEM                                                    ENDM0001
      DIMENSION KEYS(50),SCORES(200,50),PROP(200),ID(20),EM(70,50),X(71 ENDM0002
     ,),EMM(70,50),XX(20,200),IDXX(200),FMT(10)                        ENDM0003
      COMMON/KQZCOM/KQZSBX,KQZB00,KQZB06                                ENDM0004
C                                                                       ENDM0005
C     WRITTEN BY S. HENLEY  BMR  1972                                   ENDM0006
C                                                                       ENDM0007
CONVERTED AUTOMATICALLY TO ANSI FORTRAN FROM CDC3600 FORTRAN            ENDM0008
C                                                                       ENDM0009
C     SEPARATES A ROCK ANALYSIS INTO A SET OF END-MEMBERS BY THE SIMPLEXENDM0010
C     METHOD OF LINEAR PROGRAMMING IN SUCH A WAY AS TO MAXIMISE THE SUM ENDM0011
C     OF PROPORTIONS OF THE END-MEMBERS.                                ENDM0012
C     OPTIONALLY, SUBSEQUENT END-MEMBERS (MOST DIVERGENT COMPOSITIONS   ENDM0013
C     WITHIN THE DATA SET) MAY BE IDENTIFIED AND INCLUDED IN FURTHER    ENDM0014
C     ANALYSIS.                                                         ENDM0015
C                                                                       ENDM0016
C                                                                       ENDM0017
20    FORMAT(10A8)                                                      ENDM0018
      KQZB00=32                                                         ENDM0019
      KQZB06=8                                                          ENDM0020
      READ(5,20) FMT                                                    ENDM0021
      READ(5,170) M,N                                                   ENDM0022
      NS=0                                                              ENDM0023
      MM=M+1                                                            ENDM0024
      NUMBER=0                                                          ENDM0025
      NUMCYC=0                                                          ENDM0026
      LL=0                                                              ENDM0027
      IF(N.GT.0) GOTO 160                                               ENDM0028
      N=-N                                                              ENDM0029
      LL=1                                                              ENDM0030
      TOL=95.0                                                          ENDM0031
      X(71)=0.0                                                         ENDM0032
160   CONTINUE                                                          ENDM0033
170   FORMAT(2I3)                                                       ENDM0034
      IF(1.GT.N) GOTO 189                                               ENDM0035
      DO 188 I=1,N                                                      ENDM0036
      EM(MM,I)=1.0                                                      ENDM0037
      READ(5,FMT) ID(I),(EM(J,I),J=1,M)                                 ENDM0038
188   CONTINUE                                                          ENDM0039
189   CONTINUE                                                          ENDM0040
210   CONTINUE                                                          ENDM0041
      X(MM)=1.0                                                         ENDM0042
      IF(LL.EQ.1.AND.NUMCYC.GT.0) GOTO 270                              ENDM0043
      READ(5,FMT) IDX,(X(I),I=1,M)                                      ENDM0044
      CALL KQZEOF(5)                                                    ENDM0045
      GOTO(490,300),KQZSBX                                              ENDM0046
270   NIM=NUMBER+1                                                      ENDM0047
      IF(1.GT.M) GOTO 289                                               ENDM0048
      DO 288 I=1,M                                                      ENDM0049
      X(I)=XX(I,NIM)                                                    ENDM0050
288   CONTINUE                                                          ENDM0051
289   CONTINUE                                                          ENDM0052
300   CONTINUE                                                          ENDM0053
      IF(NUMCYC.GE.1) GOTO 370                                          ENDM0054
      NIM=NUMBER+1                                                      ENDM0055
      IDXX(NIM)=IDX                                                     ENDM0056
      IF(1.GT.M) GOTO 359                                               ENDM0057
      DO 358 I=1,M                                                      ENDM0058
      XX(I,NIM)=X(I)                                                    ENDM0059
358   CONTINUE                                                          ENDM0060
359   CONTINUE                                                          ENDM0061
370   IF(1.GT.N) GOTO 379                                               ENDM0062
```

```
      DO 378 I=1,N                                                    ENDM0063
      IF(1,GT,MM) GOTO 389                                            ENDM0064
      DO 388 J=1,MM                                                   ENDM0065
      EMM(J,I)=EM(J,I)                                                ENDM0066
386   CONTINUE                                                        ENDM0067
389   CONTINUE                                                        ENDM0068
378   CONTINUE                                                        ENDM0069
379   CONTINUE                                                        ENDM0070
      CALL SIMLEX(X,EMM,MM,N,KEYS,OBJFUN)                             ENDM0071
      NUMBER=NUMBER+1                                                 ENDM0072
      PROP(NUMBER)=OBJFUN*100,0                                       ENDM0073
      IF(1,GT,N) GOTO 439                                             ENDM0074
      DO 438 I=1,N                                                    ENDM0075
      IF(KEYS(I),LT,1) GOTO 470                                       ENDM0076
      KQZ001=KEYS(I)                                                  ENDM0077
      SCORES(NUMBER,I)=X(KQZ001)                                      ENDM0078
470   CONTINUE                                                        ENDM0079
438   CONTINUE                                                        ENDM0080
439   CONTINUE                                                        ENDM0081
      IF(,NOT,(NUMBER,EQ,NS)) GOTO 210                                ENDM0082
490   CONTINUE                                                        ENDM0083
      NS=NUMBER                                                       ENDM0084
      WRITE(6,720)                                                    ENDM0085
      PMIN=100,1                                                      ENDM0086
      IF(1,GT,NUMBER) GOTO 539                                        ENDM0087
      DO 538 I=1,NUMBER                                               ENDM0088
      IF(PMIN,LT,PROP(I)) GOTO 580                                    ENDM0089
      PMIN=PROP(I)                                                    ENDM0090
      NMIN=I                                                          ENDM0091
580   CONTINUE                                                        ENDM0092
      WRITE(6,730) IDXX(I),PROP(I),(SCORES(I,J),J=1,N)                ENDM0093
538   CONTINUE                                                        ENDM0094
539   CONTINUE                                                        ENDM0095
      IF(LL,EQ,0,OR,PMIN,GT,TOL) GOTO 750                             ENDM0096
      NIM=NMIN                                                        ENDM0097
      WRITE(6,740) NIM,PMIN                                           ENDM0098
      N=N+1                                                           ENDM0099
      IF(1,GT,M) GOTO 669                                             ENDM0100
      DO 668 I=1,M                                                    ENDM0101
      EM(I,N)=XX(I,NIM)                                               ENDM0102
668   CONTINUE                                                        ENDM0103
669   CONTINUE                                                        ENDM0104
      EM(MM,N)=1.0                                                    ENDM0105
      NUMBER=0                                                        ENDM0106
      NUMCYC=NUMCYC+1                                                 ENDM0107
      GOTO 210                                                        ENDM0108
720   FORMAT(8H1SUMMARY,8H OF SIMP,8HLEX L,P,,8H CLASSIF,7HICATION//  ENDM0109
     ,8H0SAMPLE ,8H PERCENT,6H ACCTD,7X,8HPROPORTI,8HONS OF E,        ENDM0110
     ,8HND-MEMBE,2HRS)                                                ENDM0111
730   FORMAT(1H ,A8,F10,2,10X,10F10,4,(/29X,10F10,4))                 ENDM0112
740   FORMAT(8H0 SAMPLE,I4,8H  SELECT,8HED AS NE,8HXT END-M,5HEMBER/  ENDM0113
     ,8H  PERCEN,8HT ACCOUN,8HTED FOR ,8HBY EXIST,8HING END-,8HMEMBERS ENDM0114
     ,,2HIS,F10,2)                                                    ENDM0115
750   CALL EXIT                                                       ENDM0116
      END                                                             ENDM0117
      SUBROUTINE SIMLEX(RTSID,COEF,MM,NN,KEYS,OBJFUN)                 SIMP0001
      DIMENSION COEF(70,50),WEIGHT(70),IBASIC(50),RTSID(70),KEYS(50)  SIMP0002
      COMMON/KQZCOM/KQZSRX,KQZB00,KQZB06                              SIMP0003
C                                                                     SIMP0004
C,,,,,SIMPLEX METHOD OF LINEAR PROGRAMMING                            SIMP0005
C     PROGRAM FROM HARBAUGH AND BONHAM-CARTER , COMPUTER SIMULATION IN GSIMP0006
C     MODIFIED BY S. HENLEY FOR USE AS AN OPTIMISATION SUBROUTINE IN  SIMP0007
C     PROGRAM ENDMEM                                                  SIMP0008
```

```
C                                                                       SIMP0009
      NVAR=NN                                                           SIMP0010
      NEQU=MM                                                           SIMP0011
      IF(1,GT,NN) GOTO 49                                               SIMP0012
      DO 48 I=1,NN                                                      SIMP0013
      KEYS(I)=0                                                         SIMP0014
      WEIGHT(I)=-1,0                                                    SIMP0015
48    CONTINUE                                                          SIMP0016
49    CONTINUE                                                          SIMP0017
      OBJFUN=0,0                                                        SIMP0018
      IF(1,GT,MM) GOTO 89                                               SIMP0019
      DO 88 I=1,MM                                                      SIMP0020
      MI=I+NN                                                           SIMP0021
      IF(1,GT,MM) GOTO 109                                              SIMP0022
      DO 108 J=1,MM                                                     SIMP0023
      COEF(J,MI)=0,0                                                    SIMP0024
108   CONTINUE                                                          SIMP0025
109   CONTINUE                                                          SIMP0026
      COEF(I,MI)=1,0                                                    SIMP0027
      KEYS(MI)=I                                                        SIMP0028
      IBASIC(I)=MI                                                      SIMP0029
88    CONTINUE                                                          SIMP0030
89    CONTINUE                                                          SIMP0031
      NVAR=MI                                                           SIMP0032
      NTIM=25                                                           SIMP0044
      IF(1,GT,NTIM) GOTO 259                                            SIMP0045
      DO 258 NT=1,NTIM                                                  SIMP0046
      SMALL1=WEIGHT(1)                                                  SIMP0047
      JENTER=1                                                          SIMP0048
      IF(2,GT,NVAR) GOTO 289                                            SIMP0049
      DO 288 J=2,NVAR                                                   SIMP0050
      IF(WEIGHT(J),GE,SMALL1) GOTO 330                                  SIMP0051
      SMALL1=WEIGHT(J)                                                  SIMP0052
      JENTER=J                                                          SIMP0053
330   CONTINUE                                                          SIMP0054
288   CONTINUE                                                          SIMP0055
289   CONTINUE                                                          SIMP0056
      SMALL2=999999,9                                                   SIMP0057
      IF(1,GT,NEQU) GOTO 359                                            SIMP0058
      DO 358 I=1,NEQU                                                   SIMP0059
      IF(COEF(I,JENTER),LE,0,000) GOTO 430                              SIMP0060
      RATIO=RTSID(I)/COEF(I,JENTER)                                     SIMP0061
      IF(RATIO,GE,SMALL2) GOTO 430                                      SIMP0062
      SMALL2=RATIO                                                      SIMP0063
      ILEAVE=I                                                          SIMP0064
430   CONTINUE                                                          SIMP0065
358   CONTINUE                                                          SIMP0066
359   CONTINUE                                                          SIMP0067
      JLEAVE=IBASIC(ILEAVE)                                             SIMP0068
      IF(SMALL2,GT,11111.1,OR,SMALL1,GE,0,0) GOTO 670                   SIMP0069
      CON=COEF(ILEAVE,JENTER)                                           SIMP0070
      IF(1,GT,NVAR) GOTO 489                                            SIMP0071
      DO 488 J=1,NVAR                                                   SIMP0072
      COEF(ILEAVE,J)=COEF(ILEAVE,J)/CON                                 SIMP0073
488   CONTINUE                                                          SIMP0074
489   CONTINUE                                                          SIMP0075
      RTSID(ILEAVE)=RTSID(ILEAVE)/CON                                   SIMP0076
      IF(1,GT,NEQU) GOTO 519                                            SIMP0077
      DO 518 I=1,NEQU                                                   SIMP0078
      IF(I,EQ,ILEAVE) GOTO 580                                          SIMP0079
      CON=COEF(I,JENTER)                                                SIMP0080
      IF(1,GT,NVAR) GOTO 559                                            SIMP0081
      DO 558 J=1,NVAR                                                   SIMP0082
```

```
      COEF(I,J)=COEF(I,J)-CON*COEF(ILEAVE,J)                          SIMP0083
  558 CONTINUE                                                        SIMP0084
  559 CONTINUE                                                        SIMP0085
      RTSID(I)=RTSID(I)-CON*RTSID(ILEAVE)                             SIMP0086
  580 CONTINUE                                                        SIMP0087
  518 CONTINUE                                                        SIMP0088
  519 CONTINUE                                                        SIMP0089
      CON=WEIGHT(JENTER)                                              SIMP0090
      IF(1.GT.NVAR) GOTO 609                                          SIMP0091
      DO 608 J=1,NVAR                                                 SIMP0092
      WEIGHT(J)=WEIGHT(J)-CON*COEF(ILEAVE,J)                          SIMP0093
  608 CONTINUE                                                        SIMP0094
  609 CONTINUE                                                        SIMP0095
      OBJFUN=OBJFUN-CON*RTSID(ILEAVE)                                 SIMP0096
      KEYS(JENTER)=ILEAVE                                             SIMP0097
      IBASIC(ILEAVE)=JENTER                                           SIMP0098
  258 CONTINUE                                                        SIMP0099
  259 CONTINUE                                                        SIMP0100
      GOTO 690                                                        SIMP0101
  670 CONTINUE                                                        SIMP0102
  690 RETURN                                                          SIMP0103
      END                                                             SIMP0104
```

3) Output

Major element data on 24 altered clastic sediments from the Holywell Bay - Mitchell area of Cornwall were supplied. One sample, a highly siliceous sandstone, chosen for its dissimilarity with the other 23 rocks, was used as the first end-member. The second sample, 165, is the rock differing most from the first, being a slate containing much basic tuffaceous material. The third sample extracted by the program, is again an unusual rock, being a metasomatised slate with 12.8 % K_2O and 27% Al_2O_3. Original analyses may be seen in Henley, 1970 (PhD thesis, University of Nottingham) or will be sent, on request to the author.

SUMMARY OF SIMPLEX L.P. CLASSIFICATION

SAMPLE	PERCENT ACCTD	PROPORTIONS OF END-MEMBERS
101	65,34	0,6534
136	64,86	0,6486
137	56,76	0,5676
155	74,21	0,7421
156	85,09	0,8509
157	66,67	0,6667
158	70,84	0,7084
159	58,33	0,5833
160	66,67	0,6667
161	72,94	0,7294
162	71,57	0,7157
163	38,89	0,3889
164	70,07	0,7007
165	18,63	0,1863
181	69,38	0,6938
182	71,72	0,7172
183	74,81	0,7481
192	78,27	0,7827
193	78,40	0,7840
194	75,15	0,7515
197	76,58	0,7658
198	77,27	0,7727
199	66,67	0,6667

SAMPLE 14 SELECTED AS NEXT END-MEMBER
PERCENT ACCOUNTED FOR BY EXISTING END-MEMBERS IS 18,63

SUMMARY OF SIMPLEX L.P. CLASSIFICATION

SAMPLE	PERCENT ACCTD	PROPORTIONS OF END-MEMBERS	
101	65,36	0,6530	0,0006
136	64,86	0,6486	0,0000
137	56,76	0,5676	0,0000
155	75,68	0,7178	0,0390
156	87,60	0,8091	0,0669
157	66,67	0,6667	0,0000
158	71,82	0,6919	0,0264
159	58,33	0,5833	0,0000
160	66,67	0,6667	0,0000
161	83,60	0,5521	0,2839
162	77,93	0,6100	0,1693
163	38,89	0,3889	0,0000
164	80,62	0,5252	0,2811
165	100,00	0,0000	1,0000
181	70,92	0,6681	0,0410
182	80,61	0,5692	0,2369
183	85,01	0,5784	0,2718
192	85,32	0,6653	0,1879
193	78,75	0,7783	0,0091
194	80,88	0,6561	0,1527
197	83,69	0,6476	0,1893
198	79,21	0,7404	0,0518
199	66,67	0,6667	0,0000

SAMPLE 12 SELECTED AS NEXT END-MEMBER.
PERCENT ACCOUNTED FOR BY EXISTING END-MEMBERS IS 38,89

SUMMARY OF SIMPLEX L.P. CLASSIFICATION

SAMPLE	PERCENT ACCTD	PROPORTIONS OF END-MEMBERS		
101	77.03	0.4242	0.0005	0.3456
136	64.86	0.6486	0.0000	0.0000
137	56.76	0.5676	0.0000	0.0000
155	78.77	0.6188	0.0347	0.1536
156	90.25	0.7563	0.0593	0.0969
157	67.39	0.6522	0.0000	0.0217
158	76.80	0.5945	0.0263	0.1472
159	69.68	0.5112	0.0000	0.1856
160	72.00	0.5600	0.0000	0.1599
161	85.68	0.5110	0.2811	0.0647
162	83.58	0.4819	0.2112	0.1538
163	100.00	0.0000	0.0000	1.0000
164	85.64	0.4261	0.2744	0.1558
165	100.00	0.0000	1.0000	0.0000
181	78.87	0.5123	0.0410	0.2353
182	90.10	0.3833	0.2368	0.2809
183	89.23	0.4951	0.2662	0.1311
192	92.73	0.5190	0.1780	0.2303
193	85.86	0.6390	0.0091	0.2106
194	86.71	0.5411	0.1450	0.1809
197	93.52	0.4534	0.1763	0.3056
198	85.74	0.6124	0.0517	0.1933
199	73.84	0.5232	0.0000	0.2151

SAMPLE 3 SELECTED AS NEXT END-MEMBER
PERCENT ACCOUNTED FOR BY EXISTING END-MEMBERS IS 56.76