ON THE RESOLUTION OF COMPOSITIONAL DATASETS INTO CONVEX COMBINATIONS OF EXTREME VECTORS

Ross Martyn Renner

Submitted for the degree of

Doctor of Philosophy

in Mathematics

at the

Victoria University of Wellington

December 1989

Endmember

(a) One of the two or more simple compounds of which an isomorphous (solid-solution) series is composed. For example, the endmembers of the plagioclase feldspar series are albite ($NaAlSi_3O_8$) and anorthite ($CaAl_2Si_2O_8$). Syn. *minal.*

(b) One of the two extremes of a series for example, types of sedimentary rock or fossils.

Source: *Glossary of Geology* (1980). R.I. Bates and J.A. Jackson (Editors). The American Geological Institute, Falls Church, Virginia, 751 pages.

ABSTRACT

Large compositional datasets of the kind assembled in the geosciences are often of remarkably low approximate rank. That is, within a tolerable error, data points representing the rows of such an array can approximately be located in a relatively small dimensional subspace of the row space.

A physical mixing process which would account for this phenomenon implies that each observation vector of an array can be estimated by a convex combination of a small number of fixed source or 'endmember' vectors. In practice, neither the compositions of the endmembers nor the coefficients of the convex combinations are known. Traditional methods for attempting to estimate some or all of these quantities have included Q-mode 'factor' analysis and linear programming. In general, neither method is successful.

Some of the more important mathematical properties of a convex representation of compositional data are examined in this thesis as well as the background to the development of algorithms for assessing the number of endmembers statistically, locating endmembers and partitioning geological samples into specified endmembers.

Keywords and Phrases: Compositional data, convex sets, endmembers, partitioning by least squares, iteration, logratios.

Acknowledgements

I wish to thank:

Professor D. Vere-Jones for his encouragement, innumerable helpful suggestions and overall supervision;

Dr G. P. Glasby for his encouragement, geochemical advice, regular referrals to the current geochemical literature and especially for access to the following geochemical databases, (i) the New Zealand Oceanographic Institute Southwest Pacific Sediment collection including the Deep Sea Drilling Project and Ross Sea collections, (ii) the Operation Raleigh survey of Lake Te Anau, (iii) sediments from Porirua Harbour, (iv) sediments from Manakau and Waitemata Harbours, (v) river-derived sediments from the New Zealand Continental Shelf, (vi) manganese micronodules and sediments collected by the West German research vessel R.V. *Sonne* from the equatorial and Southwest Pacific;

Dr F.T. Manheim for access to the World Ocean Ferromanganese Crust database of the United States Geological Survey;

Dr S.J. Haslett for sharing his geometrical insights with me;

Shelley Carlyle for supervising the production of this thesis;

Roko Vularewa for making it possible.

CONTENTS

NOTATION

Matrices and vectors are denoted by bold-faced letters preceded or followed by their orders in parentheses (except when there is absolutely no possible doubt), for example (n×p) **X**. The i-th row and j-th column vectors of a matrix such as (n×p) **X**, when they require identification, are denoted by (1×p) $\mathbf{x}_i$ and (n×1) $\mathbf{X}_j$ respectively and the intersection of the these vectors is the element x_{ij}.

The R-mode origins of Q-mode 'factor' analysis, are examined in the first chapter. In order that there be no ambiguities in the accounts of the two modes, symbols with a particular R-mode interpretation always appear with a subscript such as $\mathbf{X}_R$, $\mathbf{\Sigma}_R$. It is not necessary to take subscripts to second levels. So, for example, the covariance matrix of the random vector $\mathbf{x}_R$ is $\mathbf{\Sigma}_\mathbf{x}$ in which 'R' has been dropped. An occasional subscript is necessary to distinguish a Q-mode construct such as $\mathbf{R}_Q$. One further distinction between the arrays used in the discussions of R-mode and Q-mode procedures is that the order of a multivariate sample of n observations ($n \geq 1$) on each of p variables will always be (p×n) in the R-mode case and (n×p) in the Q-mode. Thus (p×n) $\mathbf{X}_R$ and (n×p) **X** are data matrices in the R-mode and Q-mode contexts respectively. Geometrically, the rows and columns of either array represent n points in p-space and p points in n-space, or vice versa. The R-mode account focusses on the relative positions of p points in n-space while the Q-mode account focusses on n points in p-space. With the convention described above for defining the orders of the arrays, geometrically analogous relationships in the two modes involve algebraically analogous pairs of matrix equations.

It is a convention to distinguish between random variables and the values they take by upper and lower case letters respectively. The reservation of upper and lower case letters to denote matrices and vectors as already described, prevents this distinction being employed between random variables, vectors or matrices on the one hand, and their realizations on the other.

The following list sets out in approximately alphabetical order those symbols that are used consistently with one meaning.

$[\ ]^T$	the matrix transpose of the array enclosed by [ ]
A	a constant
a (1×k)	an arbitrary row vector
$\alpha, \beta, \gamma, \delta$	integers for subscripted variables or points
$B_1, B_2, \ldots, B_k$	the k vertices of a convex polytope whose position vectors are endmembers
B (k×p)	a matrix whose rows are endmembers which are the basis for k-space S. **B** is the estimate for β
$\mathbf{B}^c$ (k×p)	the matrix product **BC** where **C** is (p×p) diagonal
$\mathbf{B}^s$ (k×q)	a matrix whose rows are the endmembers of a subcomposition ($q < p$)
$\mathbf{B}_j$ (k×1)	the j-th column of **B**
$\mathbf{B}_R$ (m×n)	the matrix of mean-corrected factor scores in the R-mode factor model
$\mathbf{B}_R{}^v$ (m×n)	the matrix of mean-corrected rotated factor scores in the R-mode factor model
$\mathbf{b}_i$ (1×p)	the i-th row (endmember) of **B**
β (κ×p)	a matrix of compositions of true or theoretical endmembers
β^p (κ×p)	a matrix of compositions of perturbed true or theoretical endmembers
C	a convex cone vertex O, whose generators are endmembers
C (p×p)	a diagonal matrix for postmultiplicative column transformations

ξ	$\mathrm{Inf}(\upsilon_j,\nu_j)$, the smaller of υ_j, ν_j
$\mathbf{D}_R$ (p×n)	the matrix of mean-corrected standardized scores in an R-mode factor model
$\mathbf{D}_{Ri}$ (1×n)	the i-th row of $\mathbf{D}_R$ (p×n)
d_{Rij}	the (i,j)th element of $\mathbf{D}_R$ (p×n)
Δ_{Σ} (p×p)	the diagonal matrix of diagonal elements of $\mathbf{\Sigma}_R$
Δ_S (p×p)	the diagonal matrix of diagonal elements of $\mathbf{S}_R$
$\Delta\mathbf{B}$ (k×p)	the matrix of incremental adjustments to matrix **B** of endmembers
E[.]	the expectation operator
E (n×p)	the matrix of residuals in a convex representation $\mathbf{X} = \mathbf{LB} + \mathbf{E}$
$\mathbf{E}^c$ (n×p)	the matrix product **EC** in the transformation $\mathbf{XC} = \mathbf{LBC} + \mathbf{EC}$
E* (n×p)	the matrix of residuals in the equation $\mathbf{X} = \mathbf{X}^* + \mathbf{E}^*$
$\mathbf{e}_i$ (1×p)	the i-th row of **E**
e (1×p)	a vector of residuals
e_{ij}	the (i,j)th component of **E**
$\mathbf{E}_R$ (p×n)	the matrix of residuals in the estimated R-mode factor equation $\mathbf{W}_R = \mathbf{L}_R\mathbf{F}_R + \mathbf{E}_R$
$\boldsymbol{\varepsilon}$ (n×p)	the error matrix in the convex model $\mathbf{X} = \mathbf{\Lambda\beta} + \boldsymbol{\varepsilon}$
ε_R (p×n)	the matrix of specific errors in an R-mode factor analysis sampling model

$\boldsymbol{\varepsilon}_R$ (p×1)	the vector of specific factors in an R-mode factor analysis model
F (n×p)	a matrix of n error vectors created by removing negative components from **L**
f (1×p)	the error vector $\mathbf{x}' - \mathbf{x}^0 = (\boldsymbol{l} - \boldsymbol{l}^0)\mathbf{B}$
$\mathbf{F}_R$ (m×n)	the matrix of factor scores in R-mode factor analysis sampling model
$\mathbf{F}_R^v$ (m×n)	the matrix of (varimax) rotated factor scores (R-mode factor model)
$\mathbf{f}_R$ (m×1)	the vector of common factors in an R-mode factor analysis model
$\mathbf{f}_R^v$ (m×1)	the vector of (varimax) rotated factors (R-mode factor analysis)
$\mathbf{f}_R^1$ ([m+p]×1)	a vector of mutually orthogonal standardized random variables
$\boldsymbol{\Phi}_R$ (p×p)	the diagonal matrix of specific variances in an R-mode factor model
ϕ_{Rii}	the i-th diagonal element of $\boldsymbol{\Phi}_R$ (p×p)
$\boldsymbol{\Phi}_R'$ (p×p)	the covariance matrix of the errors for a partial principal components solution
G (k×n)	the error coefficients matrix. $\Delta\mathbf{B} = \mathbf{GF}$
H	the convex hull of the points $B_1, B_2,...,B_k$
$\boldsymbol{\eta}_R$ (p×n)	the mean-corrected specific factors of an R-mode factor model
I	the unit matrix of any order
i,j,m	integers (m is reserved in Chapter 1 for the number of factors)
k	an integer, the dimension of estimate space S, also the estimated number of endmembers

κ	an integer, the dimension of the true mixture space $\mathcal{S}$, also the true number of endmembers
$\mathbf{L}$ (n×k)	a loading matrix of estimated mixture coefficients, the components of each row are the coefficients of a convex combination
$\boldsymbol{l}_i$ (1×k)	the i-th row of $\mathbf{L}$
$\boldsymbol{l}$ (1×k)	a vector of mixture coefficients, the coefficients of a convex combination
$\boldsymbol{l}^0$ (1×k)	the corrected solution for $\boldsymbol{l}$ in which negative components have been set to zero
$\boldsymbol{l}^*$ (1×k)	the least squares solution for $\boldsymbol{l}$ to the overdetermined system $\mathbf{x} = \boldsymbol{l}\mathbf{B}$
$\boldsymbol{l}''$ (1×k)	the linear programming solution to the overdetermined system $\mathbf{x} = \boldsymbol{l}\mathbf{B}$
l_{ij}	the (i,j)th component of matrix $\mathbf{L}$
$\mathbf{L}_R$ (p×m)	an estimated loading matrix in R-mode factor analysis
Λ (n×k)	the true or theoretical matrix of the contributions of k endmembers to each of n geological samples. Each row contains the coefficients of a convex combination
λ_i	the i-th row of Λ
λ	a true or theoretical mixture vector, the coefficients of a convex combination.
λ_{ij}	the (i,j)th component of Λ
Λ_R (p×m)	the factor pattern or loading matrix of an R-mode factor analysis model
Λ_R^v (p×m)	the rotated loading matrix of an R-mode factor analysis model

λ_{Rij}	the (i,j)th element of Λ_R (p×m)
$\mathbf{M}$ (n×(p-1))	the matrix of logratios of exact true or theoretical mixtures
$\mathbf{M}_R$ (m×m)	an orthogonal matrix, $\mathbf{M}_R\mathbf{M}_R^T = \mathbf{M}_R^T\mathbf{M}_R = \mathbf{I}$
m	an integer, the number of factors in an R-mode factor model (Chapter 1 only)
μ (1×p)	an exact true or theoretical mixture $\boldsymbol{\lambda\beta}$
n	the sample size, number of objects or geological samples or specimens
O	the origin of Euclidean p-space
O_R	the origin of Euclidean n-space in an R-mode factor analysis sampling model
P	the hyperplane through the points $B_1,B_2,...,B_k$
p	the number of variables associated with a single object or geological sample
P_R	a hypersphere in n-space whose centre is O_R and radius $\sqrt{n-1}$
q, s	integers, usually less than p
$\mathbf{R}$ (n×n)	a diagonal matrix, associated with row transformations
$\mathbf{R}_1$ (k×k)	a diagonal matrix
$\mathbf{R}_2$ (n×n)	a diagonal matrix
$\mathbf{R}_Q$ (n×n)	the similarity matrix of a Q-mode analysis
$\mathbf{R}_R$ (p×p)	the correlation matrix of an R-mode analysis

r_{Rij}	the (i,j)th element (correlation) of $\mathbf{R}_R$
$\mathbf{r}^*$	the position vector of R*, the orthogonal projection of point R into space S
r^2	the coefficient of determination between the observed and estimated values of a variable
S	the k-dimensional estimate space formed by the intersection of the positive orthant of Euclidean p-space with the subspace spanned by k estimated endmembers
$\mathcal{S}$	the κ-dimensional mixture space formed by the intersection of the positive orthant of Euclidean p-space with the subspace spanned by κ true or theoretical endmembers
S_R	the m-dimensional factor space spanned by estimated factor-vectors
$\mathcal{S}_R$	the m-dimensional factor space spanned by theoretical factor vectors
$\mathbf{S}$ (k×k)	$\text{diag}(s_1,s_2,...,s_k)$
s_i	the i-th row total of $[\mathbf{B}_1,\mathbf{B}_2,...,\mathbf{B}_q]$ and the i-th diagonal component of $\mathbf{S}$
$\mathbf{\Sigma}_R$ (p×p)	correlation matrix of the joint distribution of $\mathbf{z}_R$
t	the sum of the components of $[x'_1,x'_2,...,x'_q]$ where $q < p$
$\mathcal{T}$	the standardized sum of the standardized residual logratios
$\mathbf{U}$ (n×p)	the matrix of unitized column eigenvectors
$\mathbf{u}_j$ (n×1)	the j-th column of $\mathbf{U}$ (n×p)
υ_j, v_j	non-negative errors in the linear programming method

$\mathbf{V}$ (p×p)	the matrix of unitized column eigenvectors
$\mathbf{v}_j$ (p×1)	the j-th column of $\mathbf{V}$ (p×p)
$\mathbf{v}$ (p×1)	a unit vector
$\mathbf{v}_R$ (p×1)	a unit vector
$\mathbf{W}$ (n×p)	a matrix whose rows are unit vectors parallel to the rows of (n×p) $\mathbf{X}$
$\mathbf{W}_R$ (p×n)	the matrix of row standardized scores of multivariate random sample
$\mathfrak{X}$ (n×p)	a matrix of raw geological data in weight, volume or other units
$\mathbf{X}$ (n×p)	a matrix of compositional data, containing the concentrations of p minerals in each of n geological samples
$\mathbf{X}_R$ (p×n)	a multivariate random sample or its realization (R-mode)
$X_1, X_2, \ldots X_n$	datapoints whose position vectors are $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$
$\mathbf{x}_i$ (1×p)	the i-th row of $\mathbf{X}$
$\mathbf{x}$ (1×p)	a vector of the composition of a single geological sample
x_{ij}	the (i,j)th element of $\mathbf{X}$
$\mathbf{X}'$ (n×p)	equal to the matrix product $\mathbf{LB}$, an estimate of the matrix product $\mathbf{\Lambda\beta}$, also an estimate of $\mathbf{X}$ (n×p) when it is given
$X'_1, X'_2, \ldots, X'_n$	estimated positions in k-space S of the datapoints $X_1, X_2, \ldots X_n$
$\mathbf{x}'_i$ (1×p)	the i-th row of $\mathbf{X}'$, equal to $\boldsymbol{l}_i\mathbf{B}$
$\mathbf{x}'$ (1×p)	an exact mixture, equal to $\boldsymbol{l}\mathbf{B}$ where the components of $\boldsymbol{l}$ are the coefficients of a convex combination

x'_{ij}	the (i,j)th element of $\mathbf{X}'$
$\mathbf{X}_0$ (n×p)	a matrix of true or theoretical exact mixtures, equal to $\Lambda\beta$
$\mathbf{x}'^{s}$ (1×p)	an exact mixture for a subcomposition
$\mathbf{X}^*$ (n×p)	the matrix of orthogonal projections of the rows of $\mathbf{X}$ (n×p) into the space spanned by the eigenvectors $\mathbf{v}_1,\mathbf{v}_2,...\mathbf{v}_k$
X^*	the orthogonal projection of the point X onto k-space S
$\mathbf{x}^*$ (1×p)	the position vector of X* and least squares estimate of $\mathbf{x}$
$\mathbf{X}^c$ (n×p)	the matrix product $\mathbf{XC}$ where $\mathbf{C}$ (p×p) is diagonal
$\mathbf{x}''$ (1×p)	the linear programming estimate of $\mathbf{x}$
$\mathbf{Y}$ (n×(p-1))	the matrix of logratio data corresponding to $\mathbf{X}$ (n×p)
Ψ (p×p)	a diagonal matrix of eigenvalues
ψ_j	the j-th eigenvalue of a positive definite symmetric matrix
$\mathbf{Z}$ (n×(p-1))	the residual matrix for the logratio model
$\mathbf{z}$ (1×(p-1))	the residual vector for the logratio model
$\mathbf{Z}_R$ (p×n)	a random sample from a joint distribution of standardized random variables
$\mathbf{z}_R$ (p×1)	a vector of standardized random variables, each component having distribution mean 0 and variance 1

TERMINOLOGY

Column transformation: post-multiplication of (n×p) **X** by non-singular (p×p) diagonal matrix **C**

Composition: any (1×p) vector $\mathbf{x} = (x_1, x_2, \ldots, x_p)$, uniquely defined on a geological sample or specimen whose components are all non-negative and sum to 1. The components of **x** are often interpreted as percentages or ppm and called concentrations

Concentration: a component of a composition or part composition (see Composition)

Element: either a component of a matrix or a chemical element

Factor space: R-mode, an m dimensional space spanned by a set of m orthogonal (1×n) factor vectors.

Factor vector: R-mode, a vector (1×n) of scores of an individual factor.

Mixture: a convex combination of distinct compositions. See also Mixture coefficient

Mixture coefficient: If $\sum_{j=1}^{k} l_j = 1$ and $l_j \geq 0$ all j, then l_j is a mixture coefficient and $\sum_{j=1}^{k} l_j \mathbf{b}_j$ is a mixture, the vector $\boldsymbol{l} = (l_1, l_2, \ldots, l_k)$ is also a composition.

n-ball: $\left\{(x_1, x_2, \ldots, x_n): \; x_1^2 + x_2^2 + \cdots + x_n^2 \leq a^2\right\}$

Object: a sampling unit, a geological sample, a specimen

Object space: the measurements on a single variable taken for each of n objects define a unique point in n dimensional object-space.

Object vector: a vector of measurements on the p variables associated with a single object (also an observation vector)

Part composition: a sub-collection of the components of a composition (see Composition)

Partial composition: given a composition $\mathbf{x}$ ($1\times p$), any other composition formed from a subcollection of q components of $\mathbf{x}$, $1 < q < p$, together with a $(q+1)$th component equal to the sum of the remaining $(p-q)$ components of $\mathbf{x}$ (see Composition)

Q-mode: given an array of the values of p variables for each of n objects, an analysis of the relationships between the p-component object vectors, usually based on an $(n\times n)$ similarity matrix

R-mode: given an array of the values of p variables for each of n objects, an analysis of the relationships between the n-component variable-vectors, usually based on the $(p\times p)$ correlation matrix

Sample: either a statistical entity (see Johnson and Wichern (1988, Chapter 3)) or a geological specimen

Sample vector: as for variable-vector

Subcomposition: given a composition $\mathbf{x}$ ($1\times p$), any other composition $\mathbf{x}^s$ ($1\times q$), $1 < q < p$, formed by scaling a subcollection of q of the components of $\mathbf{x}$ to sum to 1. The scale factor being the reciprocal of the sum of the q components of $\mathbf{x}$ (see Composition)

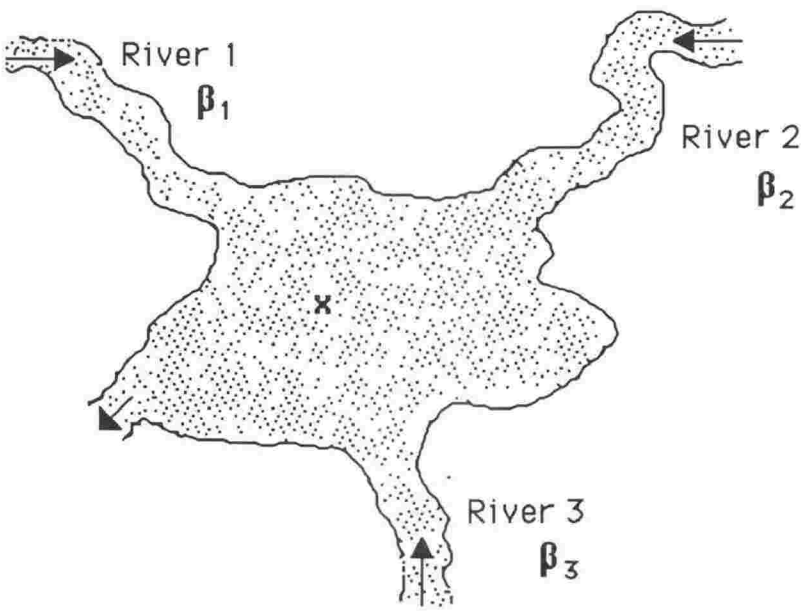
Variable-space: the measurements on the p variables associated with a single object, define a unique point in p-dimensional variable-space

Variable-vector: a vector of the n values of a single variable observed for each of n objects (also the realization of a random sample)

INTRODUCTION

This thesis marks the completion of an initial investigation into the problem of resolving each of the observation vectors of a compositional dataset into mixtures of a small set of fixed vectors known as *endmembers*, whose compositions may be identified with particular source materials.

Figure 1. Illustration of a Perfect Mixing Process Involving Three Source Endmembers.



To illustrate, consider the lake (Figure 1) which is fed by three rivers. River 1 carries sediment of fixed *composition* β_1 into the lake. Similarly, rivers 2 and 3 deposit sediments of fixed compositions β_2 and β_3 respectively. The three p-dimensional vectors β_1,β_2,β_3 are endmember (source) compositions, each containing measurements on the same p *elements*. Various dynamical processes move these source materials around the floor of the lake. If a *sample* of sediment is taken off the lake bottom, then

in a perfect (error-free) model, its composition $\mathbf{x}$ will be a mixture of the compositions $\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\boldsymbol{\beta}_3$. Algebraically, $\mathbf{x}$ will be a convex combination of $\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\boldsymbol{\beta}_3$. That is,

$$\mathbf{x} = \lambda_1\boldsymbol{\beta}_1 + \lambda_2\boldsymbol{\beta}_2 + \lambda_3\boldsymbol{\beta}_3$$

where $\lambda_1 + \lambda_2 + \lambda_3 = 1$ and $\lambda_1, \lambda_2, \lambda_3$ are non-negative.

Denoting the *mixture coefficients* $[\lambda_1, \lambda_2, \lambda_3]$ by $\boldsymbol{\lambda}$ and treating $\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\boldsymbol{\beta}_3$ as row vectors, then row vector $\mathbf{x}$ (1×p) is the matrix product of $\boldsymbol{\lambda}$ (1×3) by matrix $\boldsymbol{\beta}$ (3×p) whose rows are $\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\boldsymbol{\beta}_3$ in that order. This perfect mixture may be written,

$$\mathbf{x} = \boldsymbol{\lambda}\boldsymbol{\beta}$$

If a number n samples are taken from different locations on the lake floor, then their n composition vectors will constitute a matrix $\mathbf{X}$ (n×p) which will be the matrix product of $\boldsymbol{\Lambda}$ (n×3), a matrix of mixture coefficients, with $\boldsymbol{\beta}$ (3×p), the matrix of endmember vectors. Hence,

$$\mathbf{X} = \boldsymbol{\Lambda}\boldsymbol{\beta}$$

The rank of $\mathbf{X}$ (in this perfect model) will be exactly 3. Therefore, its n row vectors will occupy a 3-dimensional subspace $\mathcal{S}$ of p-space. Further, because each of the corresponding n sets of mixture coefficients sum to 1, these row vectors define the positions with respect to the origin of n points inside a plane triangle whose vertices (extreme points) are defined by $\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\boldsymbol{\beta}_3$.

In practice given an array of compositional data $\mathbf{X}$ (n×p), the number κ of endmembers and their compositions $\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\ldots,\boldsymbol{\beta}_\kappa$ are unknown. There will also be error-causing random contamination, which may be represented by the error matrix $\boldsymbol{\varepsilon}$ (n×p). So the theoretical model for the true decomposition of $\mathbf{X}$ is given by,

$$\mathbf{X} = \Lambda\beta + \varepsilon$$

and the complete 'linear unmixing' problem in a real situation is first to attempt to identify the space $\mathcal{S}$ spanned by the unknown $\beta_1,\beta_2,...,\beta_\kappa$ (which also implies an estimate of the integer κ), then to estimate both the matrix of endmember compositions, and the matrix of mixture coefficients. (It should be noted that departures from the matrix $\Lambda\beta$ of perfect mixtures may include non-random components. For example, the $CaCO_3$ contribution from a biogenic source will appear to vanish in marine sediments which are taken below the carbonate compensation depth ($\approx$ 4500m) where $CaCO_3$ is mostly dissolved).

Historically, the problem was not formulated in this manner. Indeed, the literature to date has really only described algorithms for constructing approximate decompositions of particular transformations of a compositional data matrix. Suppose for example $\mathbf{W}$ (n×p) is the matrix whose rows are the unit vectors in the directions of the corresponding rows of the matrix $\mathbf{X}$ (n×p) of observed compositional data. The earliest approach to the unmixing problem was a procedure for eventually expressing each row of $\mathbf{W}$ as an approximate linear combination of k 'extreme' rows of $\mathbf{W}$, where k was the analyst's choice of its approximate rank. The method employed adaptations of factor analytic algorithms which were applied to the (n×n) similarity matrix $\mathbf{R}_Q = \mathbf{W}\mathbf{W}^T$. This strategy could not work in general because a real compositional dataset rarely contains a set of 'extreme' observations, of which linear (or convex) combinations would account for each of the remaining rows of $\mathbf{W}$ (or $\mathbf{X}$) while simultaneously obeying the non-negativity constraints necessary to account for real mixtures.

The initial algorithms were extensively modified over time, both to exploit the 'constant-sum' property of compositions, and to attempt to grapple with the difficulty of absent extremes in the observed data. But what remained unchanged throughout these modifications and has endured until the present is the basic perception of the problem as

an application of factor analysis. The only challenge to the factor-analytic approach has come from the advocates of the linear programming method, but since this requires extremes to be specified *a priori*, it is seen by some to lack objectivity. In fact, the linear programming method has other weaknesses which are discussed in Chapters 2 and 3.

The adaptation of classical factor analysis brought many of the difficulties of that dubious practice to the analysis of mixtures. In the conventional analysis of a sample correlation matrix $\mathbf{R}_R$, the confusion of a perceived low approximate rank k for $\mathbf{R}_R$ with the influence of an underlying k-factor model, is almost universal. Johnson and Wichern (1988), for example, in their 5 step 'strategy for factor analysis', recommended first a 'principal component factor analysis' (standardized principal components) with which to compare the maximium likelihood factor analysis solution which was to follow. However, the existence or otherwise of an underlying factor model can not be established from the approximate rank k, low or not (see Chapter 1). In the absence of an adequate testing paradigm such as that afforded by the maximum likelihood method, the choice of the number of 'factors' (if any) is always a difficulty. Another difficulty of course pertains to the rotation of a set of initial factors into an interpretable configuration, and then there follows the problem of an appropriate oblique rotation of this. There are no formal solutions to these problems in the analysis of mixtures. Nevertheless, to a growing school of thought, the (n×n) similarity matrix $\mathbf{R}_Q$ had replaced the (p×p) correlation matrix $\mathbf{R}_R$ as an array of reliable and exploitable associations. Thus, without formulating a distinct mixing model with precisely defined properties for each of the matrices $\mathbf{\Lambda}$, $\mathbf{\beta}$, $\mathbf{\varepsilon}$, which would in turn permit the derivation of a theoretical structure for the similarity matrix $\mathbf{R}_Q$, the interchangeability of $\mathbf{R}_Q$ with the correlation matrix $\mathbf{R}_R$ was assumed, and so factor analytic concepts, terminology and algorithms were adapted to the mixing problem.

Given its importance in the development of the analysis of mixtures, the first chapter of this work is devoted entirely to a review of principal factor analysis. All those

aspects that were borrowed for mixture analysis are examined in the chapter and, although not always relevant to that purpose, some issues which challenge existing orthodoxies in the application of factor analysis are also discussed.

Following that, it is the ultimate endeavour of this work to identify a systematic approach to the analysis of mixtures which will unite in a single system all the procedures which may be employed to construct and evaluate endmember estimates, mixture coefficients and their residuals, for any given array of compositional data. There are 5 chapters whose contents are as follows:

Chapter 1, as explained above, contains a review of classical factor analysis.

Chapter 2 is on the historical background to the analysis of mixtures. It contains a survey of the most important developmental literature.

Chapter 3 covers first, those fundamental properties of mixtures that are necessary for complete analyses. Following that, an iterative method is developed for partitioning a compositional dataset by least squares into mixtures of assumed extremes, then adjusting just those 'extremes' which are identified by their regression coefficients as not being extreme enough. Descriptions of the computer algorithms employed at each stage of a complete analysis are included in the last section. This chapter is the detailed discussion of the theoretical portion of the paper by Renner *et al.* (1990) and the Technical Report by Renner (1988).

Chapter 4 describes applications of the procedures discussed in Chapter 3. The sections on the analyses of the ferromanganese nodules from the Manganese Nodule Program (United States National Science Foundation) and the Mid-Pacific cobalt-rich manganese crusts (United States Geological Survey), set out in detail the Applications portion of the paper by Renner *et al.* (1990). The section on the analysis of the bediasite

source materials is the basis for the published Comment by Renner (1989), and the analysis of the sediments from Lake Te Anau expands on the author's contribution to a submission to *The New Zealand Journal of Marine and Fresh Water Research.*

Chapter 5 examines approaches to two problems. The first is the purely technical matter of exploiting the information in a specimen which has missing values. For this, the well-researched data base of geochemical analyses of the Nazca Plate surface sediments provided a particular case for a trial study. The evaluation by Dymond (1981) of 5 specified sources for these data has become widely cited in the literature. The large number of missing values for zinc (50 out of 425 samples) had not obstructed the use of normative analyses (see Chapter 2) on which the evaluation by Dymond (*ibid*) was based. So the first section of this chapter describes an attempt to extract all the information available in the dataset in order to conduct a confirmatory analysis by the distinctly different procedures advocated earlier in this work. The second problem concerns the testing of an essentially multiplicative model for the errors in the mixing model. Again, the Nazca Plate surface sediments proved to be a suitable data base for experimentation as did the Mid-Pacific cobalt-rich manganese crusts (U.S. Geological Survey) already analyzed in Chapter 3. The second section of this Chapter largely summarizes the content of an address to the 18th Geochautauqua, Delaware, October 1989 by Renner, which has been submitted to *Mathematical Geology*.

CHAPTER 1

A REVIEW OF CLASSICAL FACTOR ANALYSIS

SUMMARY

The orthogonal linear factor model for standard (zero mean unit variance) random variables with arbitrary joint distribution is defined, and the well-known relationships between the distribution correlations, the factor loadings and the specific variances are quoted. The properties of a multivariate random sample drawn from such a distribution are examined, leading to the derivations of the principal factor and principal components solutions.

It is shown that, in general, a principal components solution cannot be rearranged into a factor analytic solution.

Although disjoint clusters of mean-corrected variable-vectors, recognizeable by their high correlations within clusters and negligible correlations between clusters, are commonly associated with factors, it is shown that their existence does not constitute a sufficient condition for an underlying factor model.

1.1 INTRODUCTION

The main purpose of this chapter is to review the essential aspects of classical factor analysis. This must be done in order to clarify the development of the original approach to the analysis of mixtures which will be described in Chapter 2. However, the development of factor analysis itself follows in part the establishment of a sequence of algorithms which were guaranteed always to work in practice, and have been followed uncritically by scores of specialists from fields as diverse as clinical psychiatry to meteorology. Accordingly, the rudiments of the subject are reviewed in some detail, and where established conventions are not supported by theory, these may be expanded on whether or not that has any relevance to mixture analysis.

The chapter is divided into a further five sections as follows:

Section 1.2 describes the well-documented properties of the distribution parameters of the orthogonal linear factor model which are true for all distributions with second order moments. The assumption that the *manifest* variables (Everitt (1984)) follow a multivariate normal distribution is not made in this or any of the following sections because it is not appropriate to the geochemical applications described in Chapter 2. Maximum likelihood estimation and the likelihood ratio criterion due to Lawley, for testing hypotheses relating to the estimates (Lawley and Maxwell (1971)) are therefore not discussed.

Section 1.3 examines the implications of the model for particular *sample vectors* and sample statistics associated with a multivariate random sample.

Section 1.4 contains the derivation of the principal factor solution when the distribution correlation matrix and specific variances are known.

Section 1.5 examines the sample (standardized) principal components solution. Although it is not a factor solution and can not be rearranged into a factor solution, it is widely interpreted as such. Further, the methods for obtaining it were eventually adapted to the analysis of mixtures.

Finally, Section 1.6 on orthogonal rotations examines 'simple structure' which, it is shown, is a phenomenum that is unrelated to the presence of a factor model.

1.1.1 R-mode and Q-mode analyses

Throughout the geochemical literature, any study based on a (p×p) correlation matrix $\mathbf{R}_R$ between p variables is described as an R-mode analysis while a study based on an (n×n) similarity matrix $\mathbf{R}_Q$ between n objects is called a Q-mode analysis. In particular, the application of factor analytic procedures to the matrices $\mathbf{R}_R$ or $\mathbf{R}_Q$ are known as R-mode or Q-mode factor analyses respectively. This terminology has been adopted where appropriate in Chapters 1 and 2 of this work.

In order to reserve notation specifically for the minor study of the factor analysis of the correlation matrix (between variables) which is set out in this chapter, most quantities in that context will appear with the subscript 'R'. Hence the use of (p×p) $\mathbf{R}_R$ above. Subscripts are not taken to second levels so that the covariance matrix of the random vector $\mathbf{f}_R$ will be denoted by $\mathbf{\Sigma}_\mathbf{f}$. Subscript 'Q' will be used rarely except to remove any ambiguity as in (n×n) $\mathbf{R}_Q$ above.

1.2 THE ORTHOGONAL LINEAR FACTOR MODEL

Almost exclusively, practical applications of R-mode factor analysis concentrate on 'factoring' an observed correlation matrix $\mathbf{R}_R$ (p×p). The elements of this matrix are the cosines of the angles between all pairs of *mean-corrected variable-vectors* of a multivariate sample. In the Q-mode analysis of mixtures, the elements of an observed similarity matrix $\mathbf{R}_Q$ (n×n) are the cosines of the angles between all pairs of *object vectors* of a compositional dataset. The 'factoring' of $\mathbf{R}_Q$ has consequently been perceived as an exercise in essentially the same algebra as that for the 'factoring' of $\mathbf{R}_R$.

Adopting $\mathbf{R}_R$ as the estimate of the correlation matrix of the joint distribution of the manifest variables $\mathbf{x}_R$ (Everitt (1984)), requires that the distribution has second order moments and implies that the actual variables being studied each have zero mean and unit variance. Accordingly, let (p×1) $\mathbf{z}_R$ be a vector of standardized components of the random vector $\mathbf{x}_R$. Further, let (m×1) $\mathbf{f}_R$, where m < p, be a vector of uncorrelated (mutually orthogonal) random variables (*common factors*) which are also standardized, let $\mathbf{\Lambda}_R$ (p×m) be a matrix of correlations (*factor loadings*) of rank m, and $\boldsymbol{\varepsilon}_R$ (p×1) be a vector of uncorrelated errors (*specific factors*) whose means are necessarily zero. Then an *orthogonal linear factor analysis model* (after Harman (1967)) is given by,

$$\mathbf{z}_R = \mathbf{\Lambda}_R \mathbf{f}_R + \boldsymbol{\varepsilon}_R \qquad (1.1)$$

where $\mathbf{x}_R$, $\mathbf{z}_R$, $\mathbf{f}_R$ and $\boldsymbol{\varepsilon}_R$ are of course all defined on the same sampling unit. (Harman (1967, p.16) assumes that sample correlations are the true population correlations for most of his exposition).

The model requires $\mathbf{f}_R$ and $\boldsymbol{\varepsilon}_R$ to be uncorrelated, so the (p×p) covariance matrix $\mathbf{\Sigma}_R = E[\mathbf{z}_R \mathbf{z}_R^T]$ of the joint distribution of $\mathbf{z}_R$ is, by equation (1.1),

$$\mathbf{\Sigma}_R = \mathbf{\Lambda}_R \mathbf{\Lambda}_R^T + \mathbf{\Phi}_R \qquad (1.2)$$

where $\mathbf{\Phi}_R = E[\boldsymbol{\varepsilon}_R \boldsymbol{\varepsilon}_R^T]$ is the diagonal matrix of *specific variances*.

Since $\mathbf{z}_R$ is the vector of standardized components of $\mathbf{x}_R$, the covariance matrix $\mathbf{\Sigma}_R$ is also the correlation matrix of the joint distributions of $\mathbf{x}_R$ and $\mathbf{z}_R$ respectively.

Assuming that model equation (1.1) is true and that the distribution correlation matrix $\mathbf{\Sigma}_R$ is known, the basic problem of factor analysis then is to determine the solutions if any, either for $\mathbf{\Lambda}_R$ (which would imply both m and $\mathbf{\Phi}_R$), or for $\mathbf{\Phi}_R$ (which would imply $\mathbf{\Lambda}_R \mathbf{M}_R$ where $\mathbf{M}_R$ is an arbitrary orthogonal (m×m) matrix), that will satisfy equation (1.2).

It will always be assumed that $\mathbf{\Sigma}_R$ is of full rank p. Although $\mathbf{\Phi}_R$ can not be the zero matrix, equation (1.2) is often described as the (matrix) factorization of $\mathbf{\Sigma}_R$.

An alternative way of expressing the model (1.1) is,

$$\mathbf{z}_R = \left[\mathbf{\Lambda}_R, \mathbf{\Phi}_R^{1/2}\right] \mathbf{f}_R^1 \qquad (1.3)$$

In this form, the p components of $\mathbf{z}_R$ are linear combinations of [m+p] mutually orthogonal standardized random variables which are the components of $\mathbf{f}_R^1$. The matrix of these combinations partitioned as in equation (1.3) displays the rectangular array of loadings and the diagonal array of specific standard deviations, which is the necessary matrix formulation for the model.

The diagonal elements of $\mathbf{\Sigma}_R$ are the unit variances of the p components of $\mathbf{z}_R$, so from equation (1.2), for i = 1,2,...,p,

$$1 = \sum_{\alpha=1}^{m} \lambda_{R i \alpha}^2 + \phi_{R i i} \qquad (1.4)$$

The sum of squared loadings on the right of equation (1.4) is the i-th *communality*, which Everitt (1984) describes as that part of the variance of z_{Ri} which is shared with the other variables via the the common factors. The second term, ϕ_{Rii}, is the i-th specific variance. All terms on the right of equation (1.4) are non-negative, and their sum is 1, which appears on the left. The magnitude of the i-th communality is therefore the proportion of the variance of the i-th variable which is accounted for by the common factors.

The correlations between the components of $\mathbf{z}_R$ are the off-diagonal elements of $\mathbf{\Sigma}_R$ given by

$$\sigma_{Rij} = \sum_{\alpha=1}^{m} \lambda_{R i \alpha} \lambda_{R j \alpha} \qquad (1.5)$$

from equation (1.2).

It is result (1.5) that many authors cite to emphasize the distinction between principal components and linear factor analysis. Following Harman (1967, pp. 14-15), the principal components of $\mathbf{z}_R$ are described as accounting for the maximum variation in the distribution (or the data) because of the well-known optimal properties of their variances (Seber (1984)). Factor analysis on the other hand is described as accounting for the covariances (correlations in this case) in view of equation (1.5). This distinction is specious. A complete set of principal components is determined by a non-singular transformation of $\mathbf{z}_R$ which, in a matrix product with its own transpose, determines $\mathbf{\Sigma}_R$ exactly (see equation (1.32)). The real distinction between principal components analysis and factor analysis is that the former should seek to recover equation (1.34) (a little further on) while the latter should seek to recover equation (1.3). (It is shown in Section 1.4 that in general, a principal components solution can not be rearranged into a factor solution).

The interpretation of the results of an orthogonal R-mode factor analysis assumed to be based on standardized manifest variables, is determined by an important property of the elements of the loading matrix Λ_R. Since the components of $\mathbf{z}_R$ and $\mathbf{f}_R$ are standardized, their intercorrelation matrix is,

$$\begin{aligned} \text{Corr}[\mathbf{z}_R, \mathbf{f}_R] &= \text{Cov}[\mathbf{z}_R, \mathbf{f}_R] \\ &= E[\mathbf{z}_R \mathbf{f}_R^T] \\ &= \Lambda_R \end{aligned} \tag{1.6}$$

by equation (1.1).

Consequently R-mode factor analysts scan the rows of the computed *factor pattern* (estimated loading) matrix in order to classify variables with the factor with which they are most highly correlated. In applications therefore, exploratory R-mode factor analysis is a clustering technique applied to the variables. Once a cluster of variables is identified, then the factor with which they are associated is in its turn identified with some perceived attribute that the variables must have in common (although it is shown in Section 1.6 that none of this sufficient evidence for the existence of an underlying factor model).

Remark

If the factors are correlated (oblique) with correlation matrix Σ_f then $\text{Corr}[\mathbf{z}_R, \mathbf{f}_R] = \Lambda_R \Sigma_f$ the *factor structure* matrix. Seber (1984, p.213) showed that multiplying the vector of factors by $\Sigma_f^{-1/2}$ transforms an oblique into an orthogonal model (whose loading matrix is obviously $\Lambda_R \Sigma_f^{1/2}$). So if an oblique factor structure matrix displays unambiguously disjoint groups of variables, then the presence of these groups may not be so evident among the loadings of the factor pattern matrix of the corresponding orthogonal representation.

Suppose that $\mathbf{\Phi}_R$ is unique. If m = 1 then $\mathbf{\Lambda}_R$ is a (p×1) column, and by equation (1.4) there are two possible forms for $\mathbf{\Lambda}_R$ which differ only by a reversal of the signs of all its elements (by equation (1.5)). This is the one dimensional elementary case of the well-known property of orthogonal transformations in Euclidean m-space of the vector of common factors $\mathbf{f}_R$ namely, that they all result in valid factorizations of $\mathbf{\Sigma}_R$. If matrix $\mathbf{M}_R$ (m×m) is orthogonal where $m \geq 2$, then $\mathbf{M}_R\mathbf{M}_R^T = \mathbf{M}_R^T\mathbf{M}_R = \mathbf{I}$. Let $\mathbf{f}_R^v = \mathbf{M}_R\mathbf{f}_R$, then $E[\mathbf{f}_R^v] = \mathbf{0}$ and $Covar[\mathbf{f}_R^v, \mathbf{f}_R^v] = \mathbf{M}_R\mathbf{I}\mathbf{M}_R^T = \mathbf{I}$. Hence the components of $\mathbf{f}_R^v$ have zero means, unit variances and are orthogonal in exactly the same way as $\mathbf{f}_R$. Substituting $\mathbf{f}_R = \mathbf{M}_R^T\mathbf{f}_R^v$ into equation (1.1) and setting $\mathbf{\Lambda}_R^v = \mathbf{\Lambda}_R\mathbf{M}_R^T$, equation (1.1) becomes,

$$\mathbf{z}_R = \mathbf{\Lambda}_R^v\mathbf{f}_R^v + \boldsymbol{\varepsilon}_R \tag{1.7}$$

which is identical in form to equation (1.1). Further,

$$\begin{aligned} \mathbf{\Lambda}_R^v(\mathbf{\Lambda}_R^v)^T &= \mathbf{\Lambda}_R\mathbf{M}_R^T\mathbf{M}_R\mathbf{\Lambda}_R^T \\ &= \mathbf{\Lambda}_R\mathbf{\Lambda}_R^T \\ &= \mathbf{\Sigma}_R - \mathbf{\Phi}_R \end{aligned} \tag{1.8}$$

Hence any set of orthogonal (uncorrelated) factors in Eucidean m-space will satisfy the model given the appropriate mapping $\mathbf{M}_R$. Similarly, it will be shown later that indefinitely many solutions to a factor analytic problem can be constructed from a particular solution.

It should be pointed out that the p (1×m) rows of $\mathbf{\Lambda}_R$ define p points in m-space, (m×1) $\mathbf{f}_R$ defines one point in m-space, and the p components of the product $\mathbf{\Lambda}_R\mathbf{f}_R$ are the scalar (inner) products between the position vectors of the first p points and the

(m×1) vector $\mathbf{f}_R$. The p rows of $\mathbf{\Lambda}_R\mathbf{M}_R^T$ are of course the p (m×1) columns of $\mathbf{M}_R\mathbf{\Lambda}_R^T$ each of which has undergone the same rigid body rotation as (m×1) $\mathbf{M}_R\mathbf{f}_R$ (therefore the p scalar products described above are invariant under this transformation). Hence it is just as valid to refer to the rotated loading matrix $\mathbf{\Lambda}_R^v$ as it is to the rotated factors $\mathbf{f}_R^v$.

The expression on the right of equation (1.8) is called the *reduced correlation matrix* (Harman (1967)). It is a correlation matrix with the ones in the diagonal replaced by the communalities. It is the basic hypothesis of R-mode factor analysis that there is a reduced distribution correlation matrix of exact rank m < p. If there is no such matrix then there is no underlying factor model. (This latter observation is also true of oblique models).

1.3 THE FACTOR SAMPLING MODEL

Associated with a set of n sampling units are the multivariate random samples $\mathbf{X}_R$ (p×n), $\mathbf{Z}_R$ (p×n), $\mathbf{F}_R$ (m×n) and $\mathbf{\varepsilon}_R$ (p×n). The columns of each of these matrices are assumed to form four distinct though related collections of n independent identically distributed vector random variables. The j-th columns of $\mathbf{Z}_R$, $\mathbf{F}_R$ and $\mathbf{\varepsilon}_R$, denoted by $\mathbf{z}_{Rj}$, $\mathbf{f}_{Rj}$ and $\mathbf{\varepsilon}_{Rj}$ respectively, are related by equation (1.1), j = 1,2,...n. The factor sampling model is therefore,

$$\mathbf{Z}_R = \mathbf{\Lambda}_R\mathbf{F}_R + \mathbf{\varepsilon}_R \tag{1.9}$$

Since $E[\mathbf{Z}_R\mathbf{Z}_R^T] = n\mathbf{\Sigma}_R$, $E[\mathbf{F}_R\mathbf{F}_R^T] = n\mathbf{I}$ and $E[\mathbf{\varepsilon}_R\mathbf{\varepsilon}_R^T] = n\mathbf{\Phi}$, equations (1.1) to (1.7), of course, also follow directly from equation (1.9).

In general, the mean vector $\boldsymbol{\mu}_{\mathbf{X}}$ and covariance matrix $\boldsymbol{\Sigma}_{\mathbf{X}}$ of the distribution of the manifest variables $\mathbf{x}_R$, are unknown. It is not possible therefore to construct matrix $\mathbf{Z}_R$ from $\mathbf{X}_R$ since

$$\mathbf{Z}_R = \boldsymbol{\Delta}_{\boldsymbol{\Sigma}}^{-1/2}\left[\mathbf{X}_R - \boldsymbol{\mu}_{\mathbf{X}}\mathbf{1}^T\right] \tag{1.10}$$

where $\boldsymbol{\Delta}_{\boldsymbol{\Sigma}}$ is the (p×p) diagonal matrix of diagonal elements of $\boldsymbol{\Sigma}_{\mathbf{X}}$ and (n×1) $\mathbf{1} = [1,1,...,1]^T$ (n×1). However, an estimate for $\mathbf{Z}_R$ is the matrix $\mathbf{W}_R$ (p×n) of variables standardized with respect to the sample means and standard deviations of $\mathbf{X}_R$. That is,

$$\mathbf{W}_R = \boldsymbol{\Delta}_S^{-1/2}\left[\mathbf{X}_R - \mathbf{m}_{\mathbf{X}}\mathbf{1}^T\right] \tag{1.11}$$

where $\boldsymbol{\Delta}_S$ is the (p×p) diagonal matrix of sample variances and $\mathbf{m}_{\mathbf{X}} = \mathbf{X}_R\mathbf{1}[1/n]$ is the sample mean vector.

(Primarily, the sample mean vector $\mathbf{m}_{\mathbf{X}}$ and the sample covariance matrix $\mathbf{S}_{\mathbf{X}} = \left[\mathbf{X}_R - \mathbf{m}_{\mathbf{X}}\mathbf{1}^T\right]\left[\mathbf{X}_R - \mathbf{m}_{\mathbf{X}}\mathbf{1}^T\right]^T \frac{1}{n-1}$ are unbiassed estimators respectively of $\boldsymbol{\mu}_{\mathbf{X}}$ and $\boldsymbol{\Sigma}_{\mathbf{X}}$).

Postmultiplying throughout equation (1.10) by (n×n) matrix $\mathbf{1}\mathbf{1}^T[1/n]$ and subtracting corresponding sides of the result from equation (1.10) creates the *mean-corrected* form $\mathbf{D}_R$ of matrix $\mathbf{Z}_R$. That is,

$$\mathbf{D}_R = \mathbf{Z}_R - \mathbf{Z}_R\mathbf{1}\mathbf{1}^T[1/n]$$

or, as in equation (1.12) below,

$$\mathbf{D}_R = \Delta_{\Sigma}^{-1/2}\left[\mathbf{X}_R - \mathbf{m}_X\mathbf{1}^T\right] \tag{1.12}$$

Denoting the i-th sample means of $\mathbf{Z}_R$ and $\mathbf{X}_R$ by $\bar{z}_{Ri}$ and $\bar{x}_{Ri}$, equations (1.11) and (1.12) yield,

$$d_{Rij} = (z_{Rij} - \bar{z}_{Ri})$$

$$= (x_{Rij} - \bar{x}_{Ri})/\sqrt{\sigma_{Xii}} \tag{1.13}$$

$$= w_{Rij}\sqrt{s_{Xii}/\sigma_{Xii}} \tag{1.14}$$

It is apparent from equations (1.11) to (1.14) that corresponding rows of $\mathbf{W}_R$ and the mean corrected matrices $\mathbf{D}_R = \left[\mathbf{Z}_R - \mathbf{m}_Z\mathbf{1}^T\right]$ and $\mathbf{D}_X = \left[\mathbf{X}_R - \mathbf{m}_X\mathbf{1}^T\right]$ are parallel vectors in Euclidean n-space. (Note: $\mathbf{m}_Z = \mathbf{Z}_R\mathbf{1}[1/n]$ in equation (1.12)).

Denoting the α-th and β-th rows of $\mathbf{W}_R$ by $(1\times n)$ $\mathbf{W}_{R\alpha}$ and $\mathbf{W}_{R\beta}$, the (α,β)th element of the sample correlation matrix $\mathbf{R}_R$ is,

$$r_{R\alpha\beta} = \frac{\mathbf{W}_{R\alpha}\mathbf{W}_{R\beta}^T}{\sqrt{\left(\mathbf{W}_{R\alpha}\mathbf{W}_{R\alpha}^T\right)\left(\mathbf{W}_{R\beta}\mathbf{W}_{R\beta}^T\right)}} \tag{1.15}$$

Or, more concisely,

$$\mathbf{R}_R = \mathbf{W}_R\mathbf{W}_R^T(1/[n-1]) \tag{1.16}$$

Alternatively, the entire right hand side of (1.15) is the scalar product of the two unit vectors in Euclidean n-space whose directions are from the origin O_R to the points $W_{R\alpha}(w_{R\alpha1},w_{R\alpha2},\ldots,w_{R\alpha n})$ and $W_{R\beta}(w_{R\beta1},w_{R\beta2},\ldots,w_{R\beta n})$ respectively.

The α-th and β-th rows of $\mathbf{W}_R$ are the (1×n) position vectors $\mathbf{W}_{R\alpha}$, $\mathbf{W}_{R\beta}$ with respect to the origin O_R of the points $W_{R\alpha}(w_{R\alpha 1}, w_{R\alpha 2},\ldots,w_{R\alpha n})$ and $W_{R\beta}(w_{R\beta 1}, w_{R\beta 2},\ldots,w_{R\beta n})$. As was noted above, these vectors are parallel to similarly defined vectors represented by the corresponding rows of the mean-corrected matrices $\mathbf{D}_R$ and $\mathbf{D}_X$. In particular therefore, $O_R D_{R\alpha} W_{R\alpha}$ and $O_R D_{R\beta} W_{R\beta}$ are straight lines. Let $\theta_{R\alpha\beta}$ be the angle $W_{R\alpha} O_R W_{R\beta}$, then by equation (1.15),

$$r_{R\alpha\beta} = \cos(\theta_{R\alpha\beta}) \tag{1.17}$$

So far, the only condition imposed on the distribution of $\mathbf{x}_R$ has been that all the second order moments exist. Now, it will also be required that $\mathbf{S}_X$ is a consistent estimator for $\mathbf{\Sigma}_X$, so that $r_{R\alpha\beta}$ is a consistent estimator for $\sigma_{R\alpha\beta}$, by equations (1.10) to (1.15), and $\theta_{R\alpha\beta}$ is a consistent estimator for $\arccos(\sigma_{R\alpha\beta})$.

The first requirement of a large sample assumption which will prevail for the remainder of this chapter is that $r_{R\alpha\beta}$ is close to $\sigma_{R\alpha\beta}$ for each $\alpha,\beta = 1,2,\ldots p$.

So an analysis of the estimated distribution correlation structure becomes an analysis of the relative angular positions of the p mean-corrected (1×n) variable-vectors, $\mathbf{W}_{R1}, \mathbf{W}_{R2},\ldots,\mathbf{W}_{Rp}$, in n-space. Equivalently, since the points $W_{R1}, W_{R2},\ldots,W_{Rp}$ lie on a hypersphere whose centre is O_R and radius is $\sqrt{n-1}$, the angular positions and hence the estimated correlation structure are also determined by the relative positions of $W_{R1}, W_{R2},\ldots,W_{Rp}$ on the hypersphere P_R. The angle $\theta_{R\alpha\beta}$ and its cosine $\cos(\theta_{R\alpha\beta})$ are respectively measures of *dissimilarity* and *similarity* between the two (1×n) vectors $\mathbf{W}_{R\alpha}$ and $\mathbf{W}_{R\beta}$. Unlike functions of the distance between the points $W_{R\alpha}$ and $W_{R\beta}$, the dissimilarity and similarity measures $\theta_{R\alpha\beta}$ and $\cos(\theta_{R\alpha\beta})$ are independent of the magnitudes of the two vectors. This property has been perceived to be especially appropriate in the study of the relationships between compositional vectors. It is the ratios of the various components of composition vectors that distinguish compositions,

not their absolute values. All parallel vectors have identical compositions.

To complete this geometrical interpretation, it is useful to examine the relationships between collections of points in n-space whose locations are determined by the factor model (1.9). First, it is necessary to mean-correct the data as before.

Postmultiplying throughout equation (1.9) by (n×n) matrix $\mathbf{1}\mathbf{1}^T[1/n]$ and subtracting corresponding terms of that result from equation (1.9) creates the mean-corrected arrays $\mathbf{D}_R$, $\mathbf{B}_R$ and η_R, where $\mathbf{D}_R$ is given by equation (1.12), $\mathbf{B}_R = \mathbf{F}_R(\mathbf{I} - \mathbf{1}\mathbf{1}^T[1/n])$ and $\eta_R = \boldsymbol{\varepsilon}_R(\mathbf{I} - \mathbf{1}\mathbf{1}^T[1/n])$. The end result is the mean-corrected factor equation below,

$$\mathbf{D}_R = \boldsymbol{\Lambda}_R\mathbf{B}_R + \eta_R \tag{1.18}$$

The sample covariance matrix associated with $\mathbf{Z}_R$ is (p×p) $\mathbf{S}_R = \mathbf{D}_R\mathbf{D}_R^T[1/(n-1)]$. Substituting for $\mathbf{D}_R$ from equation (1.18),

$$\begin{aligned}\mathbf{D}_R\mathbf{D}_R^T\frac{1}{n-1} &= (\boldsymbol{\Lambda}_R\mathbf{B}_R + \eta_R)(\boldsymbol{\Lambda}_R\mathbf{B}_R + \eta_R)^T\frac{1}{n-1} \\ &= (\boldsymbol{\Lambda}_R\mathbf{B}_R\mathbf{B}_R^T\boldsymbol{\Lambda}_R^T + \boldsymbol{\Lambda}_R\mathbf{B}_R\eta_R^T + \eta_R\mathbf{B}_R^T\boldsymbol{\Lambda}_R^T + \eta_R\eta_R^T)\frac{1}{n-1}\end{aligned} \tag{1.19}$$

Now $\mathbf{D}_R\mathbf{D}_R^T\frac{1}{n-1}$, $\mathbf{B}_R\mathbf{B}_R^T\frac{1}{n-1}$, and $\eta_R\eta_R^T\frac{1}{n-1}$ are unbiassed estimators for $\boldsymbol{\Sigma}_R$, $\boldsymbol{\Sigma}_\mathbf{f}$ and $\boldsymbol{\Phi}_R$ respectively, and $\mathbf{B}_R\eta_R^T\frac{1}{n-1}$ and $\eta_R\mathbf{B}_R^T\frac{1}{n-1}$ are unbiassed estimators of the covariances between $\mathbf{f}_R$ and $\boldsymbol{\varepsilon}_R$. Therefore, since the components of $\mathbf{f}_R$ are standardized and uncorrelated, and the random vectors $\mathbf{f}_R$, $\boldsymbol{\varepsilon}_R$ are required to be uncorrelated,

$$E\left[\mathbf{B}_R\mathbf{B}_R^T\frac{1}{n-1}\right] = \mathbf{I} \ \text{(m×m)}, \text{ and } E\left[\mathbf{B}_R\eta_R^T\frac{1}{n-1}\right] = \mathbf{0} \ \text{(m×p)} \tag{1.20}$$

Applying the expectation operator to either side of equation (1.19) clearly recovers equation (1.2). From the first result of line (1.20), the expectation of the scalar (inner)

product of the i-th and j-th (1×n) row vectors of $\mathbf{B}_R[1/\sqrt{n-1}\,]$, is the (i,j)th element of $\mathbf{I}$ (m×m). Hence the expected configuration for the row vectors of $\mathbf{B}_R[1/\sqrt{n-1}\,]$ is an orthogonal set of m unit vectors in Euclidean n-space. Similarly, from the second result of line (1.20), the expected orientation of the same m row vectors is orthogonal to the p (1×n) row vectors of $\boldsymbol{\eta}_R$. The expected configuration for the rows of $\boldsymbol{\eta}_R$ is also a mutually orthogonal set since $E[\boldsymbol{\eta}_R\boldsymbol{\eta}_R^T] = (n-1)\boldsymbol{\Phi}$, another diagonal matrix. For such configurations to be possible (though not necessarily realized), it is required that $n \geq p + m$. In fact, in order to make full use of the assumption that all sample covariances are consistent estimators of corresponding distribution covariances, it will be necessary to assume that n is large enough for the expected geometrical configurations described above to be approximately true of the positions of the sample variable-vectors. In that case, from equation (1.19),

$$\mathbf{D}_R\mathbf{D}_R^T\frac{1}{n-1} \approx \boldsymbol{\Lambda}_R\boldsymbol{\Lambda}_R^T + \boldsymbol{\Phi}_R \tag{1.21}$$

It follows also from the large sample assumption when it is applied to equations (1.14) and (1.20), that the (1×n) variable-vectors $\mathbf{D}_{R1},\mathbf{D}_{R2},\ldots,\mathbf{D}_{Rp}$ and the (1×n) *factor vectors* $\mathbf{B}_{R1},\mathbf{B}_{R2},\ldots\mathbf{B}_{Rm}$ are the position vectors of points on or near the surface of the hypersphere P_R. So writing out the i-th variable-vector of $\mathbf{D}_R$ by equation (1.18),

$$\mathbf{D}_{Ri} = \sum_{j=1}^{m}\lambda_{Rij}\mathbf{B}_{Rj} + \boldsymbol{\eta}_{Ri} \tag{1.22}$$

Dividing both sides of this by $\sqrt{n-1}$ creates unit vectors approximately, in the directions of $\mathbf{D}_{Ri}$ and $\mathbf{B}_{R1},\mathbf{B}_{R2},\ldots,\mathbf{B}_{Rm}$. It is then evident that $\lambda_{Ri1},\lambda_{Ri2},\ldots,\lambda_{Rim}$ are the direction cosines (approximately), of the i-th variable-vector with respect to an axis system defined by the (approximately orthogonal) factor-vectors. This interpretation is consistent with result (1.6) in which $\boldsymbol{\Lambda}_R$ was identified as the matrix of correlations between $\mathbf{z}_R$ and $\mathbf{f}_R$. The (nearly) orthogonal set of factor vectors $\mathbf{B}_{R1},\mathbf{B}_{R2},\ldots,\mathbf{B}_{Rm}$ span

an m-dimensional subspace $\mathcal{S}_R$ of Euclidean n-space commonly called the *factor space* (which is at variance with the notions of variable and object spaces). The loadings $\lambda_{Ri1},\lambda_{Ri2},\ldots,\lambda_{Rim}$ can also be regarded as the regression coefficients for the position vector of the orthogonal projection of the point D_{Ri} into $\mathcal{S}_R$.

Equation (1.22) is the basis of the factor plots which were historically used to initiate plane rotations, and are optionally produced by most factor analytic software to permit visual appraisal of factor solutions. Suppose for example $\mathbf{B}_{R1}$, $\mathbf{B}_{R2}$ and $\boldsymbol{\eta}_{Ri}$ are assumed to be mutually orthogonal, then $(\lambda_{Ri1}, \lambda_{Ri2})$ are the coordinates of the (projection of the) i-th variable vector on the plane of $\mathbf{B}_{R1}$ and $\mathbf{B}_{R2}$. Hence the p ordered pairs in the first 2 columns of $\boldsymbol{\Lambda}_R$ can be plotted with respect to an orthogonal reference system assumed to represent $\mathbf{B}_{R1}$ and $\mathbf{B}_{R2}$, thus portraying the relative positions of the projections of $D_{R1},D_{R2},\ldots,D_{Rp}$ in the plane of the 1st and 2nd factors. Such plots can be constructed for m(m-1)/2 pairs of columns of $\boldsymbol{\Lambda}_R$ taken two at a time. In practice, factor analysts must work with the estimate $\mathbf{L}_R$ for $\boldsymbol{\Lambda}_R$ (see the estimated model, equation (1.38) in Section 1.4). but the underlying assumptions remain the same.

Finally, the large sample assumption must also guarantee an important relation between the rows of (m×n) arrays of factor scores that have been orthogonally rotated in Euclidean m-space. The result seems to be taken for granted in the literature (possibly due to Harman (1967)) namely,

If $\mathbf{f}_R^v = \mathbf{M}_R\mathbf{f}_R$ *where* $\mathbf{M}_R$ (m×m) *is orthogonal as in the derivation of equation* (1.7), *then the expected forms of the original and the transformed mean-corrected factor-vectors are both orthogonal systems of vectors.*

The transformed factor model can be derived from Equation (1.18) as follows,

$$\begin{aligned} \mathbf{D}_R &= \Lambda_R \mathbf{B}_R + \eta_R \\ &= \Lambda_R \mathbf{M}_R^T \mathbf{M}_R \mathbf{B}_R + \eta_R \\ &= \Lambda_R^v \mathbf{B}_R^v + \eta_R \end{aligned} \tag{1.23}$$

Denoting $\mathbf{M}_R\mathbf{B}_R$ by $\mathbf{B}_R^v$ in equation (1.23), and comparing with equations (1.7) and (1.18). But,

$$\mathbf{B}_R^v(\mathbf{B}_R^v)^T = \mathbf{M}_R\mathbf{B}_R\mathbf{B}_R^T\mathbf{M}_R^T \tag{1.24}$$

If $\mathbf{B}_R\mathbf{B}_R^T = (n\text{-}1)\mathbf{I}$ the expected form, then by equation (1.24) $\mathbf{B}_R^v(\mathbf{B}_R^v)^T = (n\text{-}1)\mathbf{I}$, since since $\mathbf{M}_R\mathbf{M}_R^T = \mathbf{I}$. Hence the expectation is that the rows of $\mathbf{B}_R^v$ will constitute an an orthogonal system of vectors which also define m points on the hypersphere P_R. That is, the transformation determined by $\mathbf{M}_R$ is equivalent to a rigid body rotation of the (1×n) factor-vectors $\mathbf{B}_{R1}, \mathbf{B}_{R2}, \ldots \mathbf{B}_{Rm}$. Furthermore, this rotation takes place in the factor space $\mathcal{S}_R$. The rows of $\mathbf{B}_R$ span $\mathcal{S}_R$ and each row of $\mathbf{B}_R^v = \mathbf{M}_R\mathbf{B}_R$ is a linear combination of the rows of $\mathbf{B}_R$, so every set of rotated factor vectors belongs to $\mathcal{S}_R$.

It is clear that these conclusions demand considerable precision of all the estimates as a consequence of the large sample assumption. When this precision is assured and $\mathbf{L}_R$ is an initial estimated loading matrix, then by the derivation of equation (1.23), orthogonal rotations of the m factor vectors may generate indefinitely many arrays of factor loadings leading to the possible discovery of interpretable loadings $\mathbf{L}_R^v$ concomitant with interpretable factors $\mathbf{f}_R^v$. These issues will be examined in Section 1.6.

1.4 THE PRINCIPAL FACTOR SOLUTION

Several procedures for estimating the parameters of the R-mode factor model have been developed, of which the maximum likelihood method (Lawley and Maxwell (1971)) and the principal factor solution are possibly the most important. The multivariate normal assumption which is the basis of the maximum likelihood method can not validly be extended to the Q-mode treatment of compositional data. In any event, the historical development of the Q-mode factor approach to the analysis of mixtures was conceived by its authors (Imbrie (1963), Imbrie and Van Andel (1964) and, Klovan and Imbrie (1971)), to be an application of the principal factor solution described by Harman (1960, 1967). That is, a principal components analysis of the reduced correlation matrix. Harman (*ibid*) did not 'formally present components analysis', in which the main diagonal of the correlation matrix is unaltered. Indeed for a description of that method, he referred the reader to Hotelling (1933) and Anderson (1958, 1963). Yet, the established procedure for the Q-mode factor analysis of geochemical data has never included a modification of the main diagonal of the similarity matrix. Seber (1984, p. 222) remarked that the general confusion between R-mode factor analysis and principal components analysis 'is not helped' by the use of principal factor analysis. Algebraically, the complete principal components solution is a special case of the principal factor solution, as will be demonstrated a little further on. So although the orthodox principal factor solution is not strictly used in the analysis of mixtures, the identification of components analysis with the method warrants the discussion of their relationship which follows. (It might be noted that R-mode principal factor analysis is still the preferred approach of some geochemists seeking to identify natural element associations (see for example Walter and Stoffers (1985); Nath, Rao and Becker (1989)).

Let the reduced correlation matrix (equation (1.8)) be denoted by $\boldsymbol{\Sigma}_R^{'}$. Then,

$$\boldsymbol{\Sigma}_R^{'} = \boldsymbol{\Sigma}_R - \boldsymbol{\Phi}_R$$

$$= \Lambda_R \Lambda_R^T \tag{1.25}$$

Principal factor analysis is described by Harman (1967) as a principal components analysis of Σ_R'. That basically implies the reduction to canonical form of the symmetric matrix Σ_R' (p×p), of rank $m < p$, and hence to a factorization such as equation (1.25).

On the right of equation (1.25), $\Lambda_R \Lambda_R^T$ happens to be equal to $E[\Lambda_R \mathbf{f}_R \mathbf{f}_R^T \Lambda_R^T]$ so by equation (1.1) it follows that Σ_R' is the covariance matrix of the difference of random vectors $(\mathbf{z}_R - \boldsymbol{\varepsilon}_R)$. Let $y_R = \mathbf{v}_R^T(\mathbf{z}_R - \boldsymbol{\varepsilon}_R)$ be any linear combination of this difference provided only that $\mathbf{v}_R$ (p×1) is a unit vector ($\mathbf{v}_R^T \mathbf{v}_R = 1$). If τ_j is the orthogonal projection of the j-th column (p×1) of Λ_R (p×m) onto the unit vector $\mathbf{v}_R$ then,

$$\mathbf{v}_R^T \Lambda_R = [\tau_1, \tau_2, \ldots, \tau_m]$$

and,

$$\begin{aligned} \mathbf{v}_R^T \Sigma_R' \mathbf{v}_R &= \mathbf{v}_R^T \Lambda_R \Lambda_R^T \mathbf{v}_R \\ &= \tau_1^2 + \tau_2^2 + \ldots + \tau_m^2 \qquad (1.26) \\ &\geq 0 \qquad (1.27) \end{aligned}$$

Equation (1.26) confirms that Σ_R' is positive semidefinite. It also displays the variance $\mathbf{v}_R^T \Sigma_R' \mathbf{v}_R = \sigma_y^2$ of y_R, as the sum of the squared projections onto $\mathbf{v}_R$ of the columns of Λ_R.

The method for finding the maximum value for σ_y^2 given that $\mathbf{v}_R$ is a unit vector yields a more general result (Seber (1984), Johnson and Wichern (1988)) namely, that the critical values of σ_y^2 are equal to the p eigenvalues $\psi_1 \geq \psi_2 \geq \ldots \geq \psi_p \geq 0$, of Σ_R' and occur when $\mathbf{v}_R$ is equal to the corresponding (p×1) eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_p$ (see also Footnote 1). Therefore the critical values of the sum of squares on the right of

Footnote 1: To maximize $\mathbf{v}^T \Sigma \mathbf{v}$ given $\mathbf{v}^T \mathbf{v} = \mathbf{v}^T \mathbf{I} \mathbf{v} = 1$, introduce the Lagrange multiplier ψ and maximize $\mathbf{v}^T(\Sigma - \psi \mathbf{I})\mathbf{v}$. Partially differentiating this with respect to the components of $\mathbf{v}$, then provided Σ is symmetric, $(\Sigma - \psi \mathbf{I})\mathbf{v} = \mathbf{0}$ is a necessary condition for turning values of $\mathbf{v}^T \Sigma \mathbf{v}$. That is, $\Sigma \mathbf{v} = \psi \mathbf{v}$.

equation (1.26) are equal to $\psi_1,\psi_2,...,\psi_p$. But there are only m (< p) column vectors in $\mathbf{\Lambda}_R$, and the rank of $\mathbf{\Lambda}_R$ is m by assumption, therefore in Euclidean p-space there are (p-m) mutually orthogonal vectors which are also orthogonal to the subspace spanned by the columns of $\mathbf{\Lambda}_R$. Since the orthogonal projections of all the columns of $\mathbf{\Lambda}_R$ onto any of these (p-m) vectors must be zero, then by the inequality at line (1.27), they constitute a subset $\mathbf{v}_{m+1},\mathbf{v}_{m+2},...,\mathbf{v}_p$ of the complete set of eigenvectors $\mathbf{v}_1,\mathbf{v}_2,...,\mathbf{v}_p$, for which the corresponding eigenvalues, $\psi_{m+1} = \psi_{m+2} = ... = \psi_p = 0$. Let (p×p) $\mathbf{V} = [\mathbf{v}_1,\mathbf{v}_2,...,\mathbf{v}_p]$ and (p×p) $\mathbf{\Psi} = \text{diag}(\psi_1,\psi_2,...,\psi_p)$ then the vector $\mathbf{y}_R$ of principal components of $(\mathbf{z}_R - \boldsymbol{\varepsilon}_R)$ is,

$$\mathbf{y}_R = \mathbf{V}^T(\mathbf{z}_R - \boldsymbol{\varepsilon}_R) ,$$

and by definition,

$$\mathbf{\Sigma}_R'\mathbf{V} = \mathbf{V}\mathbf{\Psi} .$$

Since $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ (p×p),

$$\mathbf{V}^T\mathbf{\Sigma}_R'\mathbf{V} = \mathbf{\Psi} \qquad (1.28)$$

The matrix product on the left of equation (1.28) is equal to the covariance matrix of the vector $\mathbf{y}_R$. This reduces to the diagonal matrix (on the left), confirming the well-known result that principal components are mutually uncorrelated.

Since $\mathbf{V}\mathbf{V}^T = \mathbf{I}$ (p×p),

$$\begin{aligned}\mathbf{\Sigma}_R' &= \mathbf{V}\mathbf{\Psi}\mathbf{V}^T \\ &= \left(\mathbf{V}\mathbf{\Psi}^{1/2}\right)\left(\mathbf{V}\mathbf{\Psi}^{1/2}\right)^T \end{aligned} \qquad (1.29)$$

But $\psi_{m+1} = \psi_{m+2} = ... = \psi_p = 0$, so equation (1.29) can be rewritten in the form,

$$\mathbf{\Sigma}_R' = \left(\mathbf{V}'\mathbf{\Psi}'^{1/2}\right)\left(\mathbf{V}'\mathbf{\Psi}'^{1/2}\right)^T \qquad (1.30)$$

where (p×m) $\mathbf{V}' = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_m]$ and (m×m) $\Psi' = \text{diag}(\psi_1, \psi_2, ..., \psi_m)$.
Choosing,

$$\Lambda_R = \mathbf{V}'\Psi'^{1/2} \quad (p \times m) \tag{1.31}$$

creates an exact solution to equation (1.25) consistent with the uncorrelated set of standardized principal factors $\mathbf{f}_R = \Psi'^{-1/2}\mathbf{y}'_R$, where (m×1) $\mathbf{y}'_R$ contains the first m components of (p×1) $\mathbf{y}_R$ in order.

Therefore, if the distribution correlation matrix $\mathbf{\Sigma}_R$ and the specific variances $\mathbf{\Phi}_R$ are known, then the communalities are specified and the principal factor solution will determine a number m of factors and an exact associated loading matrix. Orthogonal rotations may then be employed to search for an interpretable factor pattern.

1.4.1 Distribution Principal Components

In general, neither $\mathbf{\Sigma}_R$ nor $\mathbf{\Phi}_R$ are known, although $\mathbf{\Sigma}_R$ can of course be estimated. Before moving on to that case, there remains a theoretical problem created by discarding only the assumed knowledge of $\mathbf{\Phi}_R$. This problem is stated below.

If just the distribution correlation matrix $\mathbf{\Sigma}_R$ *is known, what integers* m *or diagonal matrices* $\mathbf{\Phi}_R$ *of specific variances, will secure an exact solution to equation* (1.25) ?

The adjective 'theoretical' as used to introduce this problem alludes to the assumption of a known distribution correlation matrix, which is rarely a reality. But since this assumption has been made it is an ineluctable truth that if there is no exact solution to equation (1.25) then then there is no underlying factor model. Theoretical or otherwise, the ensuing discussion does have implications for the processing of the

observed correlation matrix $\mathbf{R}_R$.

Remark Algebraically the problem seems rather cut and dried. For any assumed value of m, there are p[p+1]/2 bilinear equations in p[m+1] unknowns (see equations (1.4) and (1.5)). The actual solubility of these equations aside, the system will be overdetermined, determined or underdetermined according as $2m <=> p-1$. So for example there is always a simple single factor solution when $p = 3$ (for which the specific variances may nevertheless be negative (the *ultra-Heywood* case)). Most applications of factor analysis are undertaken with the intention of finding solutions in which the number of factors is considerably less than half the number of variables. If the distribution correlations were known, such applications would be constrained by overdetermined systems for which there can be in general no exact solutions. When the distribution correlations are unknown, estimates of the loading matrix based on factor analyses of the sample correlation matrix are most probably fallacious, unless supported by rigorous tests on the validity of the estimated parameters. This pessimistic conclusion seems to prevail even under conditions which are the most favourable possible for the factor analyst (see Section 1.6).

The problem always has at least one solution. Set $m = p$ and $\boldsymbol{\varepsilon}_R = \mathbf{0}$ in equation (1.1) and thereafter. Then $\boldsymbol{\Phi} = \mathbf{0}$, the communalities are all equal to 1, and $\boldsymbol{\Sigma}'_R = \boldsymbol{\Sigma}_R$. Following this special case through the earlier discussion of the principal factor solution and reinterpreting the notation as appropriate, the distribution correlation matrix is given by,

$$\boldsymbol{\Sigma}_R = \boldsymbol{\Lambda}_R \boldsymbol{\Lambda}_R^T \tag{1.32}$$

where the loading matrix is,

$$\boldsymbol{\Lambda}_R = \mathbf{V}\boldsymbol{\Psi}^{1/2} \quad (p \times p) \tag{1.33}$$

This, of course, is a complete (standardized) principal components solution. Its importance in applications of factor analysis arises when the (p-k) eigenvalues

$\psi_{k+1},\psi_{k+2},\ldots,\psi_p$ of $\boldsymbol{\Sigma}_R$, are negligible. Then the realizations of $\mathbf{z}_R$ must approximately occupy a space of k dimensions (see equation (1.34) below) and a plausible strategy for modelling such a situation is to make the assumptions which are embodied in equation (1.1), with the additional requirement that the all the communalities be 'close to' 1. The interpretation then is that the data will be k-dimensional with small errors. This is a heuristic approach to the problem posed above. It does not suggest that the principal components solution in any way determines the existence of any other factor solution. Indeed, partitioning the loading matrix of equation (1.33) to separate the k 'common factors' and the errors does not in general create a solution to equations (1.2) and (1.25). Basically, $\mathbf{z}_R$ is a linear combination of p independent random variables (principal components) and partitioning $\boldsymbol{\Lambda}_R$ as described will not produce a linear combination of [k+p] independent variables akin to the [m+p] variables of equation (1.3).

Writing (p×1) $\mathbf{f}_R = [f_1, f_2, \ldots, f_p]^T$, the vector of standardized principal components $\boldsymbol{\Psi}^{-1/2}\mathbf{y}_R$, and (p×p) $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_p]$, the matrix of eigenvectors of $\boldsymbol{\Sigma}_R$, then the vector of standardized responses becomes,

$$\begin{aligned} \mathbf{z}_R &= \mathbf{V}\boldsymbol{\Psi}^{1/2}\mathbf{f}_R \\ &= \sum_{i=1}^{k} \mathbf{v}_i \psi_i^{1/2} f_i + \sum_{j=k+1}^{p} \mathbf{v}_j \psi_j^{1/2} f_j \end{aligned} \qquad (1.34)$$

where $\psi_{k+1}, \psi_{k+2}, \ldots, \psi_p$ are assumed to be very small. Suppose that the first sum on on the right of equation (1.34) corresponds to $\boldsymbol{\Lambda}_R' \mathbf{f}_R'$ where,

$$(p\times k)\ \boldsymbol{\Lambda}_R' = [\, \mathbf{v}_1\psi_1^{1/2}, \mathbf{v}_2\psi_2^{1/2}, \ldots, \mathbf{v}_k\psi_k^{1/2} \,] \text{ and } (k\times 1)\ \mathbf{f}_R' = [\, f_1, f_2, \ldots, f_k \,]^T.$$

The second sum on the right of equation (1.34) corresponds to the error vector (p×1) $\boldsymbol{\varepsilon}_R'$. Hence, the derived 'factor' equation is,

$$\mathbf{z}_R = \boldsymbol{\Lambda}_R' \mathbf{f}_R' + \boldsymbol{\varepsilon}_R' \qquad (1.35)$$

To obtain the correlation matrix $\mathbf{\Sigma}_R = E[\mathbf{z}_R\mathbf{z}_R^T]$, a direct approach is to observe that,

$$\mathbf{\Sigma}_R = \mathbf{V}\mathbf{\Psi}\mathbf{V}^T$$

and expanding the matrix product,

$$\mathbf{\Sigma}_R = \sum_{i=1}^{k} \mathbf{v}_i\mathbf{v}_i^T\psi_i + \sum_{j=k+1}^{p} \mathbf{v}_j\mathbf{v}_j^T\psi_j \qquad (1.36)$$

Equation (1.36) is the spectral decomposition of $\mathbf{\Sigma}_R$ (see Seber (1984)). The first sum on the right is of k (p×p) matrices and equals $\left(\mathbf{V}'\mathbf{\Psi}'^{1/2}\right)\left(\mathbf{V}'\mathbf{\Psi}'^{1/2}\right)^T$ where (p×k) $\mathbf{V}' = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k]$, and (k×k) $\mathbf{\Psi}'^{1/2} = [\psi_1^{1/2}, \psi_2^{1/2}, \ldots, \psi_k^{1/2}]$. Thus $\mathbf{\Lambda}_R' = \mathbf{V}'\mathbf{\Psi}'^{1/2}$ (p×k), which resembles the loading matrix at equation (1.31). The second sum on the right of equation (1.36) is of [p-k] (p×p) matrices and equals the covariance matrix of the errors, denoted by $\mathbf{\Phi}_R'$. So making the necessary substitutions into equation (1.36), the expression for $\mathbf{\Sigma}_R$ which is analogous to equation (1.2) is,

$$\mathbf{\Sigma}_R = \mathbf{\Lambda}_R'\mathbf{\Lambda}_R'^T + \mathbf{\Phi}_R' \qquad (1.37)$$

In this model, since $f_1, f_2, \ldots, f_p$ are standardized principal components, the k 'factors' $f_1, f_2, \ldots, f_k$ are mutually uncorrelated, and they are uncorrelated with the errors which are linear combinations of $f_{k+1}, f_{k+2}, \ldots, f_p$. These are necessary properties of an orthogonal factor model. But unless the errors are mutually uncorrelated so that $\mathbf{\Phi}_R'$ is a diagonal matrix, then this is not a solution to equations (1.2) and (1.25), the off-diagonal elements of $\mathbf{\Sigma}_R$ will not be exactly equal to the off-diagonal elements of $\mathbf{\Lambda}_R'\mathbf{\Lambda}_R'^T$.

If $\mathbf{\Phi}_R'$ did happen to be a diagonal matrix then since its rank is [p-k], k of its diagonal entries would necessarily be zeros. (In other words, p mutually uncorrelated errors cannot be formed from linear combinations of only [p-k] random variables). Hence k of the p components of $\mathbf{z}_R$ would have no error term and $\mathbf{z}_R$ would simply be be a nonsingular linear combination of m common factors and [p-m] specific factors. This is

not a factor model (see equation (1.3)). In general therefore, the principal components solution cannot be rearranged to form an exact factor analytic solution. An exception is that where $\psi_{m+1} = \psi_{m+2} = ... = \psi_p = \psi$. Then, $\mathbf{\Sigma}_R = \mathbf{V\Psi V}^T = \mathbf{V}[\mathbf{\Psi}-\psi\mathbf{I}]\mathbf{V}^T + \psi\mathbf{I}$, in which case (p×m) $\mathbf{\Lambda}_R = \mathbf{V}[\mathbf{\Psi}-\psi\mathbf{I}]^{1/2}$, discarding zero columns, and $\mathbf{\Phi}_R = \psi\mathbf{I}$. But when $\psi_{k+1},\psi_{k+2},...,\psi_p$ are merely very small, then by the second term on the right of equation (1.36), it may be anticipated that the contributions from the elements $\mathbf{\Phi}'_R$ to the corresponding elements of $\mathbf{\Sigma}_R$ in equation (1.37) are small. (Recalling that the $\mathbf{v}_j$ are unit vectors whose components must lie in the interval [-1, 1]).

There are wide applications for low rank approximations to large datasets, which include small non-orthogonal errors that are nonetheless orthogonal to the space spanned by an approximate basis for the dataset. The standardized principal components solution given by equation (1.35) when $\mathbf{\Sigma}_R$ is known, is a linear model which relates the p manifest variables to k eigenvectors. It is not a factor model, even so, these eigenvectors can serve as the approximate basis or, indefinitely many solutions can be constructed by orthogonally rotating $\mathbf{f}'_R$ exactly as for the factor model (see equations (1.7) and (1.23)).

In the next section, it will be assumed that $\mathbf{\Sigma}_R$ is unknown but that a model resembling equation (1.35) does account for the observations of a large dataset. The problem then is to find that solution within an arbitrary rotation of the loading matrix.

1.5 THE STANDARDIZED PRINCIPAL COMPONENTS SOLUTION

The problem of identifying an R-mode factor model becomes a good deal more obscure in the case where there is no information on any of the distribution parameters, all of which must be estimated from a multivariate random sample. The principal factor solution requires initial estimates of the number of factors, and either the communalities

or the specific variances, in order to estimate loadings. A common initializing approximation to the communalities is the set of squared multiple correlation coefficients (Dwyer (1939)) when the sample correlation matrix is non-singular. The maximum likelihood method also requires first an estimate of the number of factors as well as an approximation to the specific variances in order to execute an iterative minimization procedure which may or may not converge, or may or may not need to be steered away from negative estimates for the specific variances (*Heywood cases*).

In this section a representation of a data matrix will be examined whose existence is never in doubt. That is, the sample principal components solution. Apart from discarding all information on the distribution parameters, the underlying assumptions and the consequent geometrical interrelationships between the sample vectors follow from Section (1.3).

Let data matrix $\mathbf{X}_R$ (p×n) now be a realization of the random sample $(\mathbf{x}_{R1},\mathbf{x}_{R2},...,\mathbf{x}_{Rn})$ from the distribution of random vector $\mathbf{x}_R$ (p×1). When each of the variable-vectors (rows) of $\mathbf{X}_R$ is standardized with respect to its sample mean and standard deviation the result is $\mathbf{W}_R$ (p×n) given by equation (1.11). An estimated form of the factor model which corresponds to equation (1.18) is given by,

$$\mathbf{W}_R = \mathbf{L}_R \mathbf{B}_R + \mathbf{E}_R \tag{1.38}$$

where integer m is now an estimate of the number of factors, matrix $\mathbf{L}_R$ (p×m) is an estimate of $\mathbf{\Lambda}_R$ given m, the rows of matrix $\mathbf{B}_R$ (m×n) are the concomitant factor-vectors each of which contains the n estimated factor scores for each of the m factors, and $\mathbf{E}_R$ (p×n) is a matrix of residuals.

Restating equation (1.16), the (p×p) observed correlation matrix is given by

$$\mathbf{R}_R = \mathbf{W}_R\mathbf{W}_R^T[1/(n-1)] \tag{1.39}$$

and is assumed to be of full rank. Principal factor algorithms must allow for prior estimates of the communalities to replace the units in the diagonal of $\mathbf{R}_R$ (as previously noted, the squared sample multiple correlations are a frequent choice for initializing these estimates because the squared distribution multiple correlations are lower bounds respectively for the communalities associated with each variable (Dwyer (1939)). Since however an orthodox principal factor analytic solution may not even have an interpretation in the analysis of mixtures, the remainder of this section will concentrate on that solution for which the diagonal elements of $\mathbf{R}_R$ are unaltered that is, the principal components solution.

A rewarding method which produces all the loadings, 'factor' scores and residuals for a standardized principal components approximation is the singular value decomposition of $\mathbf{W}_R$. The following derivation of this method highlights its geometrical importance.

The (p×1) object vectors $\mathbf{w}_{R1},\mathbf{w}_{R2},\ldots,\mathbf{w}_{Rn}$ which are the columns of (p×n) $\mathbf{W}_R$, are also the position vectors with respect to the origin O of n points in Euclidean p-space (O also happens to be the centroid of these n points by equation (1.11). Let $\mathbf{v}_R$ (p×1) be any unit vector ($\mathbf{v}_R\mathbf{v}_R^T = 1$) through O and let υ_j be the orthogonal projecton of the vector $\mathbf{w}_{Rj}$ onto $\mathbf{v}_R$, produced if necessary. Then,

$$\mathbf{v}_R^T\mathbf{W}_R = [\upsilon_1,\upsilon_2,\ldots,\upsilon_n]$$

so,

$$\mathbf{v}_R^T\mathbf{W}_R\mathbf{W}_R^T\mathbf{v}_R = \upsilon_1^2 + \upsilon_2^2 + \cdots + \upsilon_n^2 \tag{1.40}$$

The turning values for the expression on the right of equation (1.40) are the p eigenvalues $\psi_1 \geq \psi_2 \geq \cdots \geq \psi_p > 0$, of the symmetric matrix $\mathbf{W}_R\mathbf{W}_R^T = [n-1]\mathbf{R}_R$,

and occur when $\mathbf{v}_R$ is a corresponding eigenvector. Redefining the (p×p) matrix $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_p]$ to be the matrix of unitized eigenvectors of $[n-1]\mathbf{R}_R$ and similarly $\mathbf{\Psi} = \text{diag}(\psi_1,\psi_2,\ldots,\psi_p)$, then by definition,

$$\mathbf{W}_R\mathbf{W}_R^T\mathbf{V} = \mathbf{V}\mathbf{\Psi}$$

Premultiplying both sides of this equation by (n×p) $\mathbf{W}_R^T$,

$$\mathbf{W}_R^T\mathbf{W}_R\mathbf{W}_R^T\mathbf{V} = \mathbf{W}_R^T\mathbf{V}\mathbf{\Psi}$$

and it is evident that (n×p) $\mathbf{W}_R^T\mathbf{V}$ is a matrix of p column eigenvectors of (n×n) $\mathbf{W}_R^T\mathbf{W}_R$, with the same (p×p) diagonal matrix of eigenvalues $\mathbf{\Psi}$ as for the symmetric (p×p) matrix $\mathbf{W}_R\mathbf{W}_R^T$. Setting (n×p) $\mathbf{U} = \mathbf{W}_R^T\mathbf{V}\mathbf{\Psi}^{-1/2}$ creates a matrix of the unitized eigenvectors of the symmetric (n×n) matrix $\mathbf{W}_R^T\mathbf{W}_R$.

(since $\mathbf{U}^T\mathbf{U} = \mathbf{\Psi}^{-1/2}\mathbf{V}^T\mathbf{W}_R\mathbf{W}_R^T\mathbf{V}\mathbf{\Psi}^{-1/2} = \mathbf{\Psi}^{-1/2}\mathbf{V}^T\mathbf{V}\mathbf{\Psi}\mathbf{\Psi}^{-1/2} = \mathbf{I}$ (p×p)).

Making $\mathbf{W}_R$ the subject of the expression for $\mathbf{U}$ above yields the 'singular value decomposition' for $\mathbf{W}$ (see Seber (1984)).

$$\mathbf{W}_R = \mathbf{V}\mathbf{\Psi}^{1/2}\mathbf{U}^T \tag{1.41}$$

Suppose as before that $\psi_{k+1}, \psi_{k+2},\ldots, \psi_p$ are very small. Partitioning equation (1.41) into two sums according to the magnitudes of the eigenvalues then,

$$\mathbf{W}_R = \sum_{i=1}^{k} \mathbf{v}_i\mathbf{u}_i^T\psi_i^{1/2} + \sum_{j=k+1}^{p} \mathbf{v}_j\mathbf{u}_j^T\psi_j^{1/2} \tag{1.42}$$

In the first sum on the right hand side of this equation set (p×k) $\mathbf{V}'\mathbf{\Psi}'^{1/2}[n-1]^{-1/2} = \mathbf{L}_R'$, where (p×k) $\mathbf{V}' = [\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_k]$ and (k×k) $\mathbf{\Psi}' = \text{diag}(\psi_1,\psi_2,\ldots,\psi_k)$. Also, noting that the vectors $\mathbf{u}_i$ are transposed, set (k×n) $\mathbf{B}_R' = [\mathbf{u}_1,\mathbf{u}_2,\ldots,\mathbf{u}_k]^T[n-1]^{1/2}$. The second sum on the right of the equation will be the matrix of residuals, $\mathbf{E}_R'$. Thus equation (1.42)

with $\mathbf{L}_R'$, $\mathbf{B}_R'$ and $\mathbf{E}_R'$ as defined becomes,

$$\mathbf{W}_R = \mathbf{L}_R'\mathbf{B}_R' + \mathbf{E}_R' \tag{1.43}$$

which has the same form as equation (1.38), the estimated factor model. This solution has other properties in common with a factor solution. The 'factor' vectors (rows of $\mathbf{B}_R'$) are standardized. Since $\mathbf{V}\Psi^{1/2}$ is non-singular then from equation (1.41), each column of $\mathbf{U}$ must be mean-corrected (sample mean equal to zero) as must the 'factor' vectors. Each column of $\mathbf{U}$ is a unit vector hence the squared magnitude of each 'factor' vector is [n-1] and so the variance of its components is 1. The 'factor' vectors are necessarily mutually orthogonal and in turn, orthogonal to the error vectors (rows of $\mathbf{E}_R'$). These are not just consequences of the large sample assumption, but follow from the orthogonality of the eigenvectors $\mathbf{u}_1,\mathbf{u}_2,...,\mathbf{u}_k$ established above. Therefore by equation (1.39) the sample equivalent to equation (1.37) is,

$$\mathbf{R}_R = \mathbf{L}_R'\mathbf{L}_R'^T + \mathbf{F}_R' \tag{1.44}$$

where $\mathbf{F}_R' = \mathbf{E}_R'\mathbf{E}_R'^T[1/(n-1)]$ (not $\mathbf{F}_R'$). Finally, the (1×n) vectors $\mathbf{W}_{R1},\mathbf{W}_{R2},...,\mathbf{W}_{Rp}$ (variable-vectors) and $\mathbf{B}_{R1}', \mathbf{B}_{R2}',..., \mathbf{B}_{Rk}'$ ('factor' vectors) are the position vectors of points on the hypersphere P_R. The basic difference between this solution and equation (1.38) for the factor model is that the error vectors (rows of $\mathbf{E}_R'$) can not be mutually orthogonal under the full-rank assumptions made at the outset.

In view of the strict orthogonality of these 'factor' vectors, premultiplying $\mathbf{B}_R'$ by the (k×k) orthogonal matrix $\mathbf{M}_R$ creates another strictly orthogonal set $\mathbf{B}_R'^v$ (see the discussion following equation (1.24). So by rotating the the former set of vectors, equation (1.43) becomes,

$$\mathbf{W}_R = \mathbf{L}_R'\mathbf{M}_R^T\mathbf{M}_R\mathbf{B}_R' + \mathbf{E}_R'$$

Thus an alternative solution may be written,

$$\mathbf{W}_R = \mathbf{L}_R^{'v}\mathbf{B}_R^{'v} + \mathbf{E}_R' \tag{1.45}$$

in the same way as the rotated orthodox factor solution. This transformation is used in the Q-mode factor analysis of mixtures to attempt to create nonnegative loadings which are a necessary condition for a mixture representation. Unlike the R-mode case, the actual values that would then be taken by the rotated factor vectors are regarded as an integral part of the resulting solution.

The i-th row of equation (1.43) can be written,

$$\mathbf{W}_{Ri} = \sum_{j=1}^{k} l_{Rij}'\mathbf{B}_{Rj}' + \mathbf{E}_{Ri}' \tag{1.46}$$

(which resembles equation (1.22)) and it follows from the geometrical interrelationships described above that $l_{Ri1}, l_{Ri2}, \ldots, l_{Rik}$ are the direction cosines (correlations) of the vector $\mathbf{W}_{Ri}$ in the directions of the respective 'factor' vectors. The 'communalities' given by,

$$h_{Ri}^{'2} = \sum_{j=1}^{k} l_{Rij}^{'2} \tag{1.47}$$

are an initial measure of the goodness of fit for each variable. Geometrically, the space spanned by the 'factor' vectors intersects the *n-ball* P_R in a k-ball. If $\mathbf{W}_{Ri}$ is the position vector of a point on the surface of that k-ball then the sum of squares on the right of equation (1.47) must be 1. In general of course,

$$h_{Ri}^{'2} \leq 1 \tag{1.48}$$

If the first sum on the right of equation (1.42) is a good approximation to $\mathbf{W}_R$ then writing (n×k) $\mathbf{U}' = [\mathbf{u}_1,\mathbf{u}_2,...,\mathbf{u}_k]$, an estimate for the former matrix is,

$$\begin{aligned} \mathbf{W}_R^{'} &= \mathbf{L}_R^{'}\mathbf{B}_R^{'} \\ &= \mathbf{V}'\mathbf{\Psi}'^{1/2}\mathbf{U}'^{T} \end{aligned} \tag{1.49}$$

This estimate is held invariant under the orthogonal rotation of the 'factor' vectors which generated the alternative solution (1.45). Further, since each of its rows is a linear combination of the 'factor' vectors, each row is orthogonal to the vectors of residuals. Therefore the point defined by the i-th row vector of $\mathbf{W}_R^{'}$ is the orthogonal projection into the subspace spanned by the 'factor' vectors, of the point defined by the i-th row of $\mathbf{W}_R$. Examining the symmetry of the lower term on the right of equation (1.49) it follows that the n points in Euclidean p-space defined by the columns of $\mathbf{W}_R^{'}$ are the orthogonal projections of the corresponding columns of $\mathbf{W}_R$ into the subspace spanned by the k columns of $\mathbf{V}'$.

The approximation to a rectangular data matrix (defined somewhat differently to $\mathbf{W}_R$) by a matrix of exact rank k as in equation (1.49) will be used in Chapter 2, together with the geometrical inter-relationships just described. This section concludes with some observations on the singular value decomposition defined by equation (1.41) which will be found to be useful.

The right hand side of equation (1.40) is the sum of squared deviations from O (the centroid in p-space) of the orthogonal projections of the object vectors onto $\mathbf{v}_R$ through O. It is non-negative of course, but suppose only $r < p$ eigenvalues of the symmetric matrix $\mathbf{W}_R\mathbf{W}_R^{T}$ are non-zero. Then there are [p-r] mutually orthogonal eigenvectors for which the right of equation (1.40) is zero. That can only happen when for every orthogonal projecton $\upsilon_j = 0$, $j = 1,2,...n$. That is, each of these

eigenvectors is orthogonal to every column vector of $\mathbf{W}_R$. Accordingly these columns lie in a subspace of r dimensions spanned by the first r eigenvectors, and the column rank of $\mathbf{W}_R$ is r.

The derivation of the singular value decomposition when rank($\mathbf{W}_R$) = r < p follows that set out above except that $\mathbf{\Psi}^{-1/2}$ in the definition for $\mathbf{U}$ must be defined to be that (p×p) diagonal matrix which has i-th diagonal element equal to $\psi_i^{-1/2}$ if $\psi_i > 0$, and zero otherwise. Although with this adaptation [p-r] columns of $\mathbf{U}$ (n×p) will be zero vectors, (p×p) $\mathbf{V}$ must be p mutually orthogonal column eigenvectors. Then the expression for $\mathbf{U}$ can be rearranged into equation (1.41) as before, and the right hand side of that equation reduces to $[\mathbf{v}_1,\mathbf{v}_2,...,\mathbf{v}_r]\text{diag}(\sqrt{\psi}_1,\sqrt{\psi}_2,...,\sqrt{\psi}_r)[\mathbf{u}_1,\mathbf{u}_2,...,\mathbf{u}_r]^T$.

Equation (1.41) possesses a certain symmetry. Taking the matrix transpose of both sides of the equation leads to a rearranged expression which permits the same interpretations to be made of its elements as were made of the original. Hence the column eigenvectors of (n×r) $\mathbf{U}$ are in the orthogonal directions of the critical values of the spread about O_R in n-space (which is however not the centroid). The sums of the squared projections of the rows of $\mathbf{W}_R$ onto these (n×1) eigenvectors are equal to the corresponding eigenvalues. Also the (p×1) columns of $\mathbf{W}_R$ have been resolved in the directions of the (p×1) orthogonal unit column vectors of $\mathbf{V}$, their components being in corresponding columns of $\mathbf{\Psi}^{1/2}\mathbf{U}^T$. Similarly the (1×n) rows of $\mathbf{W}_R$ have been resolved in the directions of the (1×n) orthogonal unit row vectors of $\mathbf{U}^T$, their components being in corresponding rows of $\mathbf{V}\mathbf{\Psi}^{1/2}$. When any of the eigenvalues are very small, all the components of the row or column vectors of $\mathbf{W}_R$ in the directions of the corresponding n or p-dimensional eigenvectors, are also very small. Thus the row or column vectors of $\mathbf{W}_R$ would be approximately orthogonal to such eigenvectors.

1.6 ORTHOGONAL ROTATIONS

In the three preceding sections, repeated reference has been made to the fact that the factorization of a correlation matrix as in equations (1.2), (1.20), (1.37) and (1.44) is not unique because postmultiplication of any loading matrix by any conformable orthogonal matrix would produce an equally valid alternative factorization. This has led to considerable investigation into analytical (objective) procedures for determining the terminal solution, and considerable controversy over the validity of any of it. Since the rotation methods that were developed for R-mode factor analysis have been adopted by the Q-mode factor analysts, it is necessary to return to the orthodox factor analysis solution.

Everitt (1984) remarked that rotation methods had acquired a certain 'notoriety'. 'Many statisticians have complained that investigators can choose to rotate factors in such a way as to get the answer they are looking for'. Everitt (*ibid*) went on to point out rightly, that the distribution of points (denoted in this work by $W_{R1}, W_{R2}, \ldots, W_{Rp}$) will remain invariant and anyway, a confirmatory analysis should always follow. There is also a constraint imposed by the existence of the factor space $\mathcal{S}_R$. Because all factor vectors must belong to $\mathcal{S}_R$, it is actually not possible to construct 'designed' loadings.

Historically, a rotation to a terminal solution was the resultant of a sequence of rotations. Factor plots were constructed in the planes of putative factor vectors taken two at a time. Orthogonal reference axes were drawn to represent the factor vectors, and the coordinates of the points representing the (projections of the) p variables were the corresponding p ordered pairs of estimated loadings on the chosen pair of factors. An orthogonal rotation in the plane would be identified, which ideally would result in most of the p points being near to either one of the two new axes or near the origin O_R with a few points removed from the origin but between the two axes. Within the constraints imposed by equation (1.4), the coordinates (loadings) of these points in the new system

would in the main be high on one axis and low on the other or both very small, with just a few having moderate loadings on both axes. The sequence of transformations of all possible pairs of factors was repeated until this 'simple structure' (Thurstone (1947)) for all the loadings was acheived (see Harman (1967), Lawley and Maxwell (1971), Morrison (1976), Everitt (1984) and Johnson and Wichern (1988))). Thus, in applications of exploratory factor analysis which started with a data matrix $\mathbf{X}_R$ (p×n) and no hypotheses about the underlying factor structure, the construction of any solution $\mathbf{L}_R$ for Λ_R by any means, has conventionally implied the identification of an m-dimensional factor space S_R spanned by an orthogonal reference system in the directions of m supposed factor vectors. The desired rotation of this reference system to a terminal solution would, if the configuration of the data permitted, bring each axis near to all the points of one cluster. It might be noted that there is nothing in these conventions that challenges (tests) the basic hypothesis that there is a reduced correlation matrix of exact rank $m < p$.

From the estimated factor model (equation (1.38)), any initial solution for $\mathbf{B}_R$ determines an m-dimensional factor space S_R spanned by its (1×n) rows $\mathbf{B}_{Ri}$, i = 1,2,...m. This space is a subspace of the n-dimensional Euclidean object space. The orthogonal rotation $\mathbf{M}_R\mathbf{B}_R$ creates an alternative set of m orthogonal factor vectors (equation (1.23) *et seq.*) which, being linear combinations of the rows of $\mathbf{B}_R$, also belong to S_R. Thus the transformations $\mathbf{L}_R\mathbf{M}_R^T$ of any initial solution $\mathbf{L}_R$ for Λ_R are the loadings associated only with orthogonal systems belonging to some fixed m-dimensional space S_R. If S_R or, equivalently, the initial $\mathbf{L}_R$ are ill-chosen, then all other loading matrices formed by rotations will be equally spurious. (The Q-mode equivalent of this situation can lead to serious errors of interpretation (see Section 4.3). Let it be assumed that by some process a sound approximation to the true factor space has been identified (such identification may be implicit brought about by the construction of $\mathbf{L}_R$, or explicit due to the construction of $\mathbf{B}_R$), then it remains only to discover the 'correct' loading matrix.

In 1935, Thurstone proposed three conditions for 'simple structure'. Later, he presented five criteria which were an extension of the original three conditions (Thurstone (1947)). Morrison (1976) observed that, 'in essence these criteria say that under a simple structure the responses fall into generally mutually exclusive groups whose loadings are high on single factors, perhaps moderate to low on a few factors, and of negligible size on the remaining dimensions'. After 1935, many individuals made specific proposals in pursuit of objective analytical procedures for calculating a multiple factor simple structure solution. The 'normal varimax' criterion for rotation to a simple structure published by Kaiser (1958), together with a computer program he published in a later paper, would, apart from subsequent improvements to the algorithm, appear to be the best analytical procedure for achieving optimal simplicity of the column loadings.

Given the data matrix $\mathbf{W}_R$, the most favourable geometrical scenario can be built up as follows. Suppose each of m orthogonal factor vectors is *similar* to at least one variable-vector (row of $\mathbf{W}_R$) so that there are indeed m factors. In addition, each variable-vector is similar to one factor so that there are just m orthogonally located clusters of points on the hypersphere P_R defined by the p variable-vectors. Each row of the loading matrix should then have one large and [m-1] small correlations (Thurstone required at least one zero). Each column of the loading matrix should contain either large or small correlations as individual variable-vectors make either small or large angles respectively with each factor-vector. In practice, such a conjunction of such favourable circumstances is not commonplace. Nevertheless, if it occurs, then the variance (simplicity) of the squared loadings (cosines) in each column of the loading matrix described above should be a maximum above all other unitized linear combinations, and also therefore the sum of these variances should be a maximum. And that is the essence of Kaiser's procedure. Ultimately, the initial loading matrix is transformed so as to achieve simultaneously the greatest spread between 0 and 1 of the squared cosines in each of the columns. As much as is possible, the unsquared cosines are pushed towards +1, -1 or around 0. (In the 'normal varimax' criterion each row of the initial loading

matrix is first scaled so that the sum of squares of the row loadings is equal to 1. This implies that every factor vector defines a point on the surface of an m-ball whose centre is O_R and radius $\sqrt{n-1}$).

Thus classifying variables on the basis of their correlations with particular factors is geometrically equivalent to attempting to identify clusters of points on the surface of the hypersphere P_R that are located around or near to orthogonal axes through the origin O_R. Albeit these clusters are usually somewhat fuzzy, they may alternatively be disjoint but oblique. (Tryon and Bailey (1970, p. 118) described the application of cluster analysis to the correlation matrix as a discrete form of factor analysis). Neither the occurrence on the hypersphere P_R of disjoint clusters in particular nor their relative locations, are properties of any n-dimensional reference system of which the m factor vectors may be treated as a subset. And that is the usual justification for rotating these axes in search of a loading matrix with 'simple structure' (see Everitt (1984, p.25)).

1.6.1 A Note on Mutually Exclusive Groups

Although there is no mathematical requirement that the points $W_{R1}, W_{R2}, \ldots, W_{Rm}$ be clustered, examples in the literature inevitably reinforce the universal application of varimax rotation as a method for classifying disjoint groups of variables to individual factors (see for example Everitt (1984, pp. 22-30)).

But the occurrence of disjoint clusters on P_R is not a sufficient condition for the existence of an underlying factor model. This last assertion may be expressed more precisely:

The partitioning of the points defined by p mean-corrected variable-vectors into m disjoint clusters on the surface of the hypersphere P_R, *so that the pencil of*

variable-vectors through the origin O_R *to the points of each cluster is orthogonal to every other such pencil, is not a sufficient condition for an exact m-factor solution to equation* (1.2).

The large sample assumption (Section 1.3) guarantees that the relative positions of the mean-corrected variable vectors are close to their limiting positions. However, departures from this configuration due to sample variability are covered by a much stronger statement namely,

The existence of an m-block diagonal distribution correlation matrix is not a sufficient condition for an exact orthogonal m-factor model.

The proof follows by *reductio ad absurdum.*

Suppose that the distribution correlation matrix is of the block diagonal type, then clearly any reduced correlation matrix that is formed from it, as in equation (1.50) below, is also a block diagonal matrix.

$$\boldsymbol{\Sigma}_R^{'} = \boldsymbol{\Sigma}_R - \boldsymbol{\Phi}_R \tag{1.50}$$

the matrix $\boldsymbol{\Phi}_R$ (p×p) is of course diagonal but otherwise the values of its elements are irrelevant. Thus equation (1.50) may be written,

$$\boldsymbol{\Sigma}_R^{'} = \begin{bmatrix} \boldsymbol{\Sigma}_1^{'} & & & \\ & \boldsymbol{\Sigma}_2^{'} & & \\ & & \ddots & \\ & & & \boldsymbol{\Sigma}_m^{'} \end{bmatrix} \tag{1.51}$$

Let the reduced correlation matrices $\boldsymbol{\Sigma}_i^{'}$ in the diagonal of $\boldsymbol{\Sigma}_R^{'}$ be of order $(r_i \times r_i)$ i = 1,2,...,m. Assume that all correlations $\sigma_{R\alpha\beta}$ $\alpha \neq \beta$ are large (positive) if they belong to these submatrices, otherwise they are zero and thus conform to the block

diagonal array. Correlations are the limiting values of the cosines of the angles between all possible pairs of the mean-corrected variable-vectors. The r_i variables of the i-th group are uncorrelated with the variables of any other group (equation (1.51)), therefore their mean-corrected variable vectors will each tend to be orthogonal to those of any other group. Hence, all such vectors will define orthogonally located disjoint clusters of points on the surface of the hypersphere P_R. As Harman (1967, p.97) noted, 'a group of variables having high intercorrelations is encompassed by a "cone" (Harman's quotation marks) with a relatively small generating angle. If a vector or reference axis of the common factor space is chosen in the midst of this cone, all variables in the group will correlate high with it.' In the case of the block diagonal correlation matrix described above, any linear combination of the vectors of a group (the vector through the group centroid for example) will be orthogonal to all vectors belonging to every other group (including those through group centroids). So there are indefinitely many m-dimensional orthogonal systems of vectors which will serve as Harman's reference axes. Hence, this particular correlation structure appears to be explained by m orthogonal factors whose linear relationship to the manifest variables is defined by a loading matrix of perfect simplicity. It remains to show that this is not true in general.

Let it be supposed that the reduced correlation matrix (equation (1.51)) arises from m orthogonal factors. Therefore by equation (1.25),

$$\Sigma_R' = \Lambda_R \Lambda_R^T \tag{1.52}$$

Now Λ_R can be partitioned as in equation (1.53) below

$$\Lambda_R = \begin{bmatrix} \Lambda_1 \\ \Lambda_2 \\ \vdots \\ \Lambda_m \end{bmatrix} \tag{1.53}$$

such that $\mathbf{\Lambda}_i$ is of order ($r_i \times m$). Substituting from equation (1.53) into equation (1.52),

$$\mathbf{\Sigma}_R^{'} = \begin{bmatrix} \mathbf{\Lambda}_1\mathbf{\Lambda}_1^T & \mathbf{\Lambda}_1\mathbf{\Lambda}_2^T & \cdots\cdots & \mathbf{\Lambda}_1\mathbf{\Lambda}_m^T \\ \mathbf{\Lambda}_2\mathbf{\Lambda}_1^T & \mathbf{\Lambda}_2\mathbf{\Lambda}_2^T & \cdots\cdots & \mathbf{\Lambda}_2\mathbf{\Lambda}_m \\ \cdot & \cdot & \cdots\cdots & \cdot \\ \mathbf{\Lambda}_m\mathbf{\Lambda}_1^T & \mathbf{\Lambda}_m\mathbf{\Lambda}_2^T & \cdots\cdots & \mathbf{\Lambda}_m\mathbf{\Lambda}_m^T \end{bmatrix} \tag{1.54}$$

Each of the matrix products of this array reduce to,

$$\mathbf{\Lambda}_i\mathbf{\Lambda}_j^T = \begin{cases} \mathbf{\Sigma}_i^{'} \text{ if } i = j \\ \mathbf{0}\ (r_i \times r_j) \text{ if } i \neq j \end{cases} \tag{1.55}$$

The row vectors $\boldsymbol{\lambda}_{i\alpha}(1 \leq \alpha \leq r_i)$ from $\mathbf{\Lambda}_i$ and $\boldsymbol{\lambda}_{j\beta}\ (1 \leq \beta \leq r_j)$ from $\mathbf{\Lambda}_j$ are of order ($1 \times m$), and their scalar (inner) product $\boldsymbol{\lambda}_{i\alpha}\boldsymbol{\lambda}_{j\beta}^T$ is a correlation coefficient which, by result (1.55) is positive if i = j and zero otherwise. Hence for i = 1,2,...,m, $\boldsymbol{\lambda}_{i\alpha}$ can not be orthogonal to any row vector in $\mathbf{\Lambda}_i$ but must be orthogonal to every other ($1 \times m$) row vector of $\mathbf{\Lambda}_R$. Therefore the m ($1 \times m$) vectors $\boldsymbol{\lambda}_{1\alpha}, \boldsymbol{\lambda}_{2\beta},\ldots, \boldsymbol{\lambda}_{m\omega}$, chosen respectively from $\mathbf{\Lambda}_1, \mathbf{\Lambda}_2,\ldots, \mathbf{\Lambda}_m$, constitute an m-dimensional orthogonal set. Suppose $\boldsymbol{\lambda}_{1\gamma}$ $(1 \leq \gamma \leq r_1)$ from $\mathbf{\Lambda}_1$ is not parallel to $\boldsymbol{\lambda}_{1\alpha}$, which is also from $\mathbf{\Lambda}_1$. Then $\boldsymbol{\lambda}_{1\gamma}$ is a linear combination of $\boldsymbol{\lambda}_{1\alpha}, \boldsymbol{\lambda}_{2\beta},\ldots, \boldsymbol{\lambda}_{m\omega}$ since these vectors span m-space. But $\boldsymbol{\lambda}_{1\gamma}$ is orthogonal to each of $\boldsymbol{\lambda}_{2\beta},\ldots, \boldsymbol{\lambda}_{m\omega}$, therefore the coefficients of these vectors in such a linear combination are necessarily zero and so the supposition that $\boldsymbol{\lambda}_{1\gamma}$ is not parallel to $\boldsymbol{\lambda}_{1\alpha}$ is false. It follows by the same reasoning that for i = 1,2,...m,

$$\mathbf{\Lambda}_i = \begin{bmatrix} \lambda_{i1} \\ \lambda_{i2} \\ \cdot \\ \cdot \\ \lambda_{ir_i} \end{bmatrix} \mathbf{m}_i \tag{1.56}$$

where $\mathbf{m}_i$ ($1 \times m$) is a row of the ($m \times m$) orthogonal matrix $\mathbf{M}_R$ ($\mathbf{M}_R\mathbf{M}_R^T = \mathbf{M}_R^T\mathbf{M}_R = \mathbf{I}$).

Substituting for $\mathbf{\Lambda}_i$ from equation (1.56) into equation (1.55), it follows that the (α,β)th correlation in the i-th correlation submatrix is $\sigma_{i\alpha\beta} = \lambda_{i\alpha}\lambda_{i\beta}$ since $\mathbf{m}_i\mathbf{m}_i^T = 1$.

That is, the i-th correlation submatrix is the outer product of the $(r_i \times 1)$ vector of equation (1.56) with itself, for i = 1,2,...,m. This is a particularly severe constraint on the correlation submatrices which, for $r_i > 3$, is not in general true (even if $r_i = 3$, the result is not possible unless $\sigma_{i\alpha\beta}\sigma_{i\alpha\gamma} / \sigma_{i\beta\gamma} < 1$ for the 3 permutations of the off-diagonal elements, otherwise the specific variances are negative). Therefore, m mutually exclusive subcollections of correlated variables are not a sufficient condition for the existence of an orthogonal m-factor model, which completes the proof.

Postmultiplying the loading matrix $\mathbf{\Lambda}_R$ by $\mathbf{M}_R^T$ to create a new loading matrix $\mathbf{\Lambda}_R^v$ does not of course alter the correlations, but since

$$\mathbf{m}_i\mathbf{M}_R^T = [0,0,...,1,...0]$$

which is a $(1 \times m)$ unit vector with the 1 in the i-th position, the new loading matrix is given by,

$$\mathbf{\Lambda}_R^v = \begin{bmatrix} \boldsymbol{\lambda}_1 & & & \\ & \boldsymbol{\lambda}_2 & & \\ & & \ddots & \\ & & & \boldsymbol{\lambda}_m \end{bmatrix} \qquad (1.57)$$

The $(p \times m)$ matrix on the right of this equation may also be described as block diagonal. All entries other than the elements of $\boldsymbol{\lambda}_i$ are zero. Each $\boldsymbol{\lambda}_i$ is a column vector $(r_i \times 1)$ of the form enclosed in square brackets on the right of equation (1.56). Comparing result (1.57) with Thurstone's five criteria for simple structure (Thurstone (1947)), this loading matrix achieves perfection.

The block diagonal correlation matrix is the ideal outcome for the factor analyst. Variables can be grouped (clustered) unambiguously on inspection, the underlying factors are revealed, and their mutual orthogonality is assured. Geometrically, the mean-corrected variable vectors from a multivariate sample should form themseves into m of the "cones" of Harman (1967). Yet this, the most favourable disposition of the distribution parameters for the analyst does not in general arise from a factor model. Associating each of an orthogonal system of common factors with each of the orthogonal groups of variables would, in general, be a misinterpretation of the true state of nature.

What is true in general however, is that a block diagonal distribution correlation matrix is a consequence of a standardized principal component model (see equations (1.33) and (1.34)) in which the (p×p) loading matrix is also a block diagonal matrix with a matching block structure. The demonstration of this statement is quite straightforward. Let the distribution correlation matrix be given by,

$$\boldsymbol{\Sigma}_R = \begin{bmatrix} \boldsymbol{\Sigma}_1 & & & \\ & \boldsymbol{\Sigma}_2 & & \\ & & \ddots & \\ & & & \boldsymbol{\Sigma}_m \end{bmatrix} \tag{1.58}$$

where $\boldsymbol{\Sigma}_i$ is of order $(r_i \times r_i)$. An eigenvector $\mathbf{v}$ of $\boldsymbol{\Sigma}_R$ may be partitioned as below,

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_m \end{bmatrix} \tag{1.59}$$

so that $\mathbf{v}_i$ is $(r_i \times 1)$. If λ is the eigenvalue of $\boldsymbol{\Sigma}_R$ associated with $\mathbf{v}$ then,

$$\boldsymbol{\Sigma}_R \mathbf{v} = \lambda \mathbf{v} \tag{1.60}$$

and by equations (1.58) and (1.59),

$$\begin{aligned} \boldsymbol{\Sigma}_1 \mathbf{v}_1 &= \lambda \mathbf{v}_1 \\ \boldsymbol{\Sigma}_2 \mathbf{v}_2 &= \lambda \mathbf{v}_2 \\ &\cdots \quad \cdot \quad \cdots \\ \boldsymbol{\Sigma}_m \mathbf{v}_m &= \lambda \mathbf{v}_m \end{aligned} \tag{1.61}$$

Assume that none of the $\boldsymbol{\Sigma}_i$ have a common eigenvalue (which will be true in general) or zero eigenvalues (which is a consequence of $\boldsymbol{\Sigma}_R$ being full rank), then the p solutions for λ in equation (1.60) must be the set of p eigenvalues of the m submatrices $\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \ldots, \boldsymbol{\Sigma}_m$. Suppose $\boldsymbol{\Sigma}_i \mathbf{v}_{i\alpha} = \lambda_{i\alpha} \mathbf{v}_{1\alpha}$, $1 \le \alpha \le r_i$, then the (p×1) eigenvector of $\boldsymbol{\Sigma}_R$ associated with $\lambda_{i\alpha}$ will be the vector of equation (1.59) but with $\mathbf{v}_1 = \mathbf{0}$, $\mathbf{v}_2 = \mathbf{0}, \ldots, \mathbf{v}_i = \mathbf{v}_{i\alpha}, \ldots, \mathbf{v}_m = \mathbf{0}$. Assembling these p eigenvectors into one array, the orthogonal matrix $\mathbf{V}$ (p×p) of unitized column eigenvectors of $\boldsymbol{\Sigma}_R$ can therefore also be arranged in block diagonal form. Denoting the $(r_i \times r_i)$ matrix of column eigenvectors of $\boldsymbol{\Sigma}_i$ by $\mathbf{V}_i$, matrix $\mathbf{V}$ (p×p) may be written as,

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_1 & & & \\ & \mathbf{V}_2 & & \\ & & \ddots & \\ & & & \mathbf{V}_m \end{bmatrix} \tag{1.62}$$

Recalling equation (1.33), the (p×p) loading matrix for this case is,

$$\boldsymbol{\Lambda}_R = \mathbf{V} \boldsymbol{\Psi}^{1/2} \quad (p \times p) \tag{1.63}$$

Since (p×p) $\boldsymbol{\Psi}$ is the diagonal matrix of eigenvalues (appropriately positioned in the diagonal),

$$\boldsymbol{\Psi} = \operatorname{diag}\left(\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2, \ldots, \boldsymbol{\Psi}_m\right) \tag{1.64}$$

where $\mathbf{\Sigma}_i\mathbf{V}_i = \mathbf{V}_i\mathbf{\Psi}_i$, $i = 1,2,...,m$. Hence $\mathbf{\Lambda}_R$ is also a block diagonal matrix of correlations of the manifest variables $\mathbf{z}_R$ with the standardized principal components $\mathbf{f}_R$. The relation between $\mathbf{z}_R$ and $\mathbf{f}_R$, both appropriately partitioned, is given by,

$$\begin{bmatrix} z_1 \\ z_2 \\ \cdot \\ \cdot \\ z_m \end{bmatrix} = \begin{bmatrix} V_1\Psi_1^{1/2} & & & \\ & V_2\Psi_2^{1/2} & & \\ & & \cdot & \\ & & & V_m\Psi_m^{1/2} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \cdot \\ \cdot \\ f_m \end{bmatrix} \qquad (1.65)$$

The question now is, can a more parsimonious model be derived from this by discarding [p-m] standardized principal components? For m dimensions to dominate and provide a good overall account of $\mathbf{\Sigma}_R$ and each of its submatrices (equation (1.58)), it would be necessary for the largest eigenvalue of each of the submatrices $\mathbf{\Sigma}_1,\mathbf{\Sigma}_2,...,\mathbf{\Sigma}_m$ to belong to the set of m largest eigenvalues of $\mathbf{\Sigma}_R$. This is not altogether unreasonable given that the sum of the r_i eigenvalues of the i-th submatrix is r_i (with their mean value 1 exactly). Such a solution exists necessarily if for each submatrix of $\mathbf{\Sigma}_R$ one eigenvalue is greater than 1 while the remainder are less than 1. Then one column in each of the submatrices of equation (1.65) tends to contain dominant correlations (with one principal component), the column sums of squares are clearly equal to the eigenvalues. The error matrix for such a model would not be diagonal but block diagonal.

In practice, sample correlation matrices may hardly ever resemble block diagonal matrices, but they can frequently be arranged into diagonal blocks of mainly large correlations with 'off-diagonal' correlations of moderate to small absolute values. Otherwise factor analysis would be a curiosity. At the conclusion of a subchapter on hypothesis testing Lawley and Maxwell (1971, p.38) tender some rather strange advice: 'It should always be kept firmly in mind that, except in artificial sampling experiments, the basic factor model is, like other models, useful only as an approximation to reality, and it should not be taken too seriously' (*sic*).

CHAPTER 2

THE HISTORICAL BACKGROUND TO THE ANALYSIS OF MIXTURES

SUMMARY

Large compositional datasets of the kind assembled in the geosciences are often of remarkably low approximate rank. That is, within a tolerable error, data points representing the rows of such an array can approximately be located in a relatively small dimensional subspace of the row space. A physical mixing process which would account for this phenomenon implies that each observation vector of the array can be estimated by a convex combination of a small number of fixed source or 'endmember' vectors. The compositions of such vectors are usually unknown, and must be estimated. Given the endmember compositions, either known or estimated, the matrix of proportional contributions of each endmember to each observation (approximate mixture) of the compositonal dataset, must also be estimated.

The analysis of mixtures unites in a single system all the procedures which may be employed to achieve and evaluate any of these estimates.

Historically, the construction of a mixing representation for a given array of compositional data has been regarded as an application of Q-mode factor analysis. The similarity matrix to be factored, which corresponds to the correlation matrix of an R-mode factor analysis, is the matrix of cosines of all possible angles between the position vectors of the datapoints. In a terminal solution, the number of endmembers equals the number of factors, the compositions of the endmembers are the factor scores of the rotated factors, and the mixture coefficients are available as the components of the rotated factor pattern (estimated loading) matrix. Most applications of Q-mode factor analysis however, fall somewhat short of terminal solutions.

More recently linear programming has been added to the methods available for linearly unmixing geological samples into specified endmembers, and least squares techniques have been proposed in order to adjust a set of specified endmembers when compliance with the linear programming constraints creates unacceptable errors between the estimated and the observed data.

These approaches to the analysis of mixtures are reviewed in this chapter.

2.1 BACKGROUND

Large compositional datasets of the kind assembled in the geosciences are often of remarkably low approximate rank. That is within a tolerable error, data points representing the rows of such an array can approximately be located in a relatively small dimensional subspace of the row space.

This phenomenon was recognized more than two decades ago. R-mode factor analyses that were employed to identify 'natural element groupings', would frequently yield sets of eigenvalues for the correlation matrices in which only small proportions of the eigenvalues were greater than one. In general, any result of this kind usually indicates that points representing the standardized data have non-negligible components along only a small number of orthogonal axes through the centroid of the data. Thus, such points approximately occupy a space of relatively small dimensions.

However, papers had begun to appear questioning the validity of any analysis based on the correlation matrix of 'constant-sum' data, that is data typically measured as percentages or in parts per million and possibly 'closed' by summing to a constant such as 100% or 1,000,000 ppm (see, for example, Chayes (1960)). It has repeatedly been reported since, that significant correlations can be induced between a pair of variables by manipulating the overall number of variables present in 'closed' sets (see Aitchison (1986)), that is by forming *subcompositions*. An elementary example (cited by Imbrie and Van Andel (1964) and Aitchison (1986)), is the trivial subcomposition of dimension 2 in which both variables sum to a constant, and for which the correlation coefficient between the pair is necessarily -1. So in general, R-mode analyses, based in particular on correlations between the variables, can not yield absolute and invariant attributes of the data such as 'natural element groupings'.

A physical explanation for the low approximate rank of a compositional dataset, is that the sample compositions derive from some natural mixing process. This is the historical geochemical 'mixing model'. Algebraically, it implies that each object vector (geological sample) is approximately a convex combination of a small number of fixed source, or endmember vectors which have some genuine physical existence. A particularly satisfactory feature of this explanation when it is valid, is that it does not depend on the modelling of relationships between the variables.

It will be shown in Chapter 3 that, under appropriate conditions, ratios formed from the components of the endmembers of a dataset of subcompositions, are equal to the ratios formed from the corresponding components of the endmembers of the full dataset. So, manipulating the number of variables in a subcomposition changes the values of the corresponding variables in each of its contributing endmembers by just the common scale factor which restores the sum-to-constant property. Therefore, in the interpretation of the data as the outcome of a mixing process, 'natural element groupings' are invariant in general, albeit as components of endmember compositions. This is intuitively reasonable.

For a given compositional dataset, a mixture analysis may briefly be described as the determination of a set of endmember compositions together with the contributions of those endmembers to each composition of the dataset.

When the attention of geochemists first focussed on mixing processes, it was realized that a mixture analysis was the approximate resolution of a set of object vectors into linear combinations of extreme (most dissimilar) endmember vectors. It was also realized that factor analysis was the approximate resolution of a set of mean-corrected variable vectors into linear combinations of orthogonal (most uncorrelated) factor vectors (see equation (1.38). Viewed in this way, the two procedures were originally seen to be essentially the same. The fundamental difference being that the data matrix processed by

one technique was the transpose of the data matrix processed by the other. Further, the distinction between the former and the latter approaches to the data had already been identified by psychologists as R- and Q-techniques.

Cattell (1952, pp. 90-91) used an array of the scores on 8 tests achieved by each of 7 people to present an illustrative definition of Q-mode factor analysis. His particular point being that 'the transposed or Q-technique' consisted of 'correlating' persons instead of variables (test scores). The correlation between two people indicated the extent to which they resembled each other. Just as it was true that many people were required for a reliable correlation between two tests, so it was also true that many tests were required for a reliable correlation between two people. In any event, the illustration evoked a possible (7×7) correlation matrix between people which could be factor analyzed, thus clustering the 'artistic' personalities (for example) and so forth.

Cattell (*ibid*) did not explicitly specify a Q-mode model. It is to be inferred that it was of the same form as equation (1.1). Therefore, the factor analytic algorithms that had been developed to construct R-mode solutions were clearly applicable to the 'transposed or Q-technique' merely by transposing the raw data matrix.

Here then was a precedent for the mixture analysts. Although the Q-mode correlations between objects were found to be unusable, there was an alternative similarity measure which was easy to interpret, and, all the necessary algebra was covered in the R-mode literature.

2.2 Q-MODE FACTOR ANALYSIS OF COMPOSITIONAL DATA

The establishment of Q-mode factor methods in the analysis of mixtures, originated with the interpretation made of the work of the factor-analytic school by the

earliest advocates of the technique. In two seminal papers, Imbrie (1963) and Imbrie and Van Andel (1964) acknowledged their debt in particular, to the publications of Thurstone (1947), Cattell (1952) and Harman (1960). The papers by Imbrie (1963), Imbrie and Van Andel (1964), Manson and Imbrie (1964) and Klovan and Imbrie (1971), together presented extensively worked examples, source code for computer programs, and some of the algebra for principal factor algorithms as described in detail by Harman (1960, 1967). It was later noted by Miesch (1976b) that the term factor analysis may have been 'unacceptable' to workers in multivariate statistics. Miesch pointed out that the diagonal values in the similarity matrices had been unity in all applications at that time, so the method might best have been referred to as 'components' analysis. Miesch nevertheless promoted factor analytic concepts, terminology and procedures. For example, the results of an application of 'the extended form of Q-mode factor analysis' to some petrologic-mixing problems, included 'the communalities' of vectors which represented compositions in the 'two-factor varimax space' (Miesch (1976b, p.30)).

Subsequently, the Q-mode factor model has become a well-established concept in geological research. Like R-mode factor analysis, certain conventions controlling the execution and presentation of the results of a Q-mode factor analysis are entrenched. Although of dubious value, such conventions include the reproduction of tables of eigenvalues, the percentages of the variabilities accounted for by each of the factors, barcharts to depict the 'compositions' of varimax-rotated factors (which always contain negative components) and sometimes even the 'communalities' associated with the samples (see for example, Leinen (1987), De Carlo, McMurtry and Kim (1987)).

In the view of many authors (including Harman (1967, p16)), the basic problem of R-mode factor analysis, given m the number of factors, is the estimation of the factor loading matrix. (Although it might be suggested that estimating m was the basic problem). In the analysis of mixtures however, given k the estimated number of

endmembers, a Q-mode terminal factor solution must yield endmembers with feasible (non-negative) compositions, together with a concomitant array of feasible (non-negative) *mixture coefficients*. The endmembers should have been reached by an oblique rotation of the axes of the varimax reference system into suitably extreme positions in the positive orthant of *variable-space*. But there are no guaranteed methods for accomplishing this, and so most analyses stop at the varimax rotation of the factors.

It will be argued further on that, although it is possible in some special cases to construct satisfactory terminal solutions using the Q-mode factor method (certainly with contrived data), in general it is not. The 'factoring' of the Q-mode similarity matrix is inefficient and unnecessary, and the application of varimax rotation to a subcollection of the principal axes of that matrix possibly obscures rather than locates extreme vectors.

The remainder of this section contains a description of the history of the Q-mode factor analysis of compositional data. An important aspect of that history are the attempts to solve two problems that were particularly associated with the method namely, the identification of feasible extremes and the enforcement of the non-negativity constraints. These two problems will appear again in Section 2.3 and Chapter 3.

2.2.1 Q-mode Factor Analysis

Note: Throughout the discussion of the Q-mode analyses that follow in this chapter, and in the development of the analysis of mixtures in subsequent chapters, the arrays of observations made on p variables for each of n objects will be denoted by (n×p) matrices $\mathfrak{X}, \mathbf{X}, \mathbf{W}$ *as needed. That is, when compared to the preceding treatment of R-mode factor analysis, n and p will always be interchanged.*

The first document to set out in some detail a rationale and an algorithm for the Q-mode 'factor analysis' of geological data was a 'computer program manual' prepared by Imbrie (1963) and released as a Technical Report by Northwestern University. The particular abstract problem addressed in the document was that of resolving each row or column vector of a dataset, into components in the directions of a small number of fixed, oblique row or oblique column vectors. Accordingly, both R and Q-mode analyses were presented. Indeed, the author stated at one point that that both analyses are identical 'mathematically' except in the choice of similarity matrix. For an R-mode analysis, the similarity matrix specified by Imbrie was the familiar product moment correlation matrix between the variables. That is, the matrix of cosines of angles between all possible pairs of mean-corrected (1×n) variable-vectors in n-space (equation (1.15)). For a Q-mode analysis, Imbrie defined the similarity matrix to be the cosines of angles between all possible pairs of (1×p) position vectors $\mathbf{x}_i$, $\mathbf{x}_j$ of the data points X_i, X_j (objects) in p-space (again equation (1.15) with $\mathbf{D}_{Ri}$, $\mathbf{D}_{Rj}$ replaced by $\mathbf{x}_i$, $\mathbf{x}_j$). He cited Imbrie and Purdy (1962), as have many subsequent authors, for the introduction of this cosine which they called the 'coefficient of proportional similarity' and denoted by $\cos\theta$. It has the obvious property of being independent of changes to the magnitudes of the object vectors. So that for a given set of variables, the similarity matrix $\mathbf{R}_Q$ between objects is invariant to such elementary row transformations of the (n×p) array $\mathfrak{D}$ of raw (weight, volume, count...) data, as the scale changes into compositions $\mathbf{X}$ or into unit vectors $\mathbf{W}$. It is however, altered by scale changes to the columns of $\mathfrak{D}$, $\mathbf{X}$ or $\mathbf{W}$, a transformation recommended by Imbrie when simultaneously analyzing major and trace elements. The symmetric matrix $\mathbf{R}_Q$ of similarity coefficients is widely referred to as the '$\cos\theta$ matrix'.

When either an R or a Q-mode analysis concluded with the (varimax) rotation of an orthogonal set of reference vectors, Imbrie incorrectly called it a 'factor' analysis (see below). To distinguish the next possible stage, in which the reference vectors may be transformed into an oblique reference system ('representing actual cases' in a Q-mode

context), Imbrie coined the unfortunate term 'vector' analysis (which in Mathematics unites an extensive branch of Algebraic Geometry with Potential Theory, Mechanics and Continuum Mechanics). Nevertheless, it appears that Imbrie envisaged the construction of a reference system consisting of relatively few oblique vectors, in the directions of which each vector representing either a variable-vector or a geological sample could be resolved. In the Q-mode case, a component in each of these reference directions would then represent in some way the contribution of that reference vector to the sample. In essence, that view was correct.

The first of Imbrie's two computer algorithms required the mode of the analysis (R or Q) as an input argument. Different subroutines were called to calculate the appropriate similarity matrix. Otherwise, the the program extracted eigenvectors and eigenvalues of the similarity matrix (with the diagonals intact) and, if required, carried out a rotation according to the varimax criterion of the specified number of 'factors' ('a complete factor analysis'). The second algorithm performed an oblique rotation of the varimax 'factors' output by the first, thus constructing an oblique loading matrix. Imbrie stated (*ibid*, p.14) that for the initial factor matrix, 'using algebraic procedures described in detail in Harman (1960),...,the principal components method is used'. That is certainly confirmed by his worked examples. The 2nd edition of Harman's 1960 text (Harman (1967)) states quite explicitly that principal components analysis is 'not presented' as noted in Section 1.2.3. It seems certain then that Imbrie regarded his procedure as principal factor analysis. Klovan and Imbrie (1971) later used the term in the description of their improvements on Imbrie's algorithm.

The original algorithms were elementary. For the Q-mode case, suppose **X** (n×p) is a matrix of n observations on the *concentrations* of p mineral constituents (*note again that this is the transpose of the conventional notation for R-mode analyses*). It is not necessary for each row vector to be a composition, that is, with unit sum (see Terminology). Transforming the rows of **X** into unit vectors creates the matrix **W**

(n×p). The (1×p) rows $\mathbf{w}_1,\mathbf{w}_2,...,\mathbf{w}_n$ of $\mathbf{W}$ are the position vectors of the points $W_1,W_2,...,W_n$ on the surface of a unit hypersphere, centre O in p-space, since $\mathbf{w}_i\mathbf{w}_i^T = 1$, i = 1,2,...,n. The (n×n) similarity matrix for the rows of $\mathbf{X}$ whether they are compositions or not is given by,

$$\mathbf{R}_Q = \mathbf{W}\mathbf{W}^T \tag{2.1}$$

Denoting the angle between the unit vectors $\mathbf{w}_i$ and $\mathbf{w}_j$ by θ_{ij}, the (i,j)th element of $\mathbf{R}_Q$ is the inner produuct $\mathbf{w}\mathbf{w}^T$ which is equal to $\cos(\theta_{ij})$. Provided n > p, which is usually assumed, the maximum possible rank for $\mathbf{R}_Q$ is p. It has at most p positive eigenvalues and the remainder are zero (*cf.* Section 1.2.3, equations (1.25) to (1.31)) . Let $r \leq p < n$ be the actual rank of (n×p) $\mathbf{X}$. Let (n×n) diagonal matrix $\mathbf{R}$ contain the magnitudes of the row vectors of $\mathbf{X}$ so that $r_{ii} = (\mathbf{x}_i\mathbf{x}_i^T)^{1/2}$. Then $\mathbf{X} = \mathbf{R}\mathbf{W}$ and since $\mathbf{R}$ is obviously of rank n, then rank($\mathbf{W}$) = rank($\mathbf{X}$) = r. Let Ψ (n×n) be the diagonal matrix of eigenvalues (in order of magnitude down the diagonal so that the lower [n-r] are zero), and let (n×n) $\mathbf{U}$ be the matrix of corresponding unitized column eigenvectors, then by definition,

$$\mathbf{R}_Q\mathbf{U} = \mathbf{U}\Psi$$

Postmultiplying both sides of this by $\mathbf{U}^T$,

$$\mathbf{R}_Q = \mathbf{U}\Psi\mathbf{U}^T$$

$$= \left(\mathbf{U}\Psi^{1/2}\right)\left(\mathbf{U}\Psi^{1/2}\right)^T \tag{2.2}$$

$$= \mathbf{L}_0\mathbf{L}_0^T \tag{2.3}$$

In equation (2.3), the initial loading matrix (n×r) $\mathbf{L}_0$ contains the first r columns of $\mathbf{U}\Psi^{1/2}$ which it replaces in equation (2.2), by discarding the [n-r] zero column vectors formed in the matrix product by the zeros in the diagonal of Ψ. Imbrie assumed that there must exist a unique (r×p) matrix $\mathbf{B}_0$ whose r rows are mutually orthogonal, such

that $\mathbf{W} = \mathbf{L}_0\mathbf{B}_0$. The result is true by the singular value decomposition theorem (transpose equation (1.41), and note the remarks at the end of Section 1.2.4). Basically, Imbrie followed the interpretation of the elements of $\mathbf{L}_0$ as direction cosines with respect to an orthogonal reference system, which was noted earlier in relation to equation (1.46) but with p and n interchanged. It was not necessary however, to construct $\mathbf{B}_0$. A rotation according to the varimax criterion was then undertaken on $\mathbf{L}_0$ to create (n×r) $\mathbf{L}^v$, and notionally (r×p) $\mathbf{B}^v$, so that in the abstract,

$$\mathbf{W} = \mathbf{L}^v\mathbf{B}^v \tag{2.4}$$

Imbrie remarked in his discussions of both R and Q-mode analyses that the varimax procedure should align the reference vectors (rows) of $\mathbf{B}^v$ as near as their orthogonality would permit, to the extremes of the 'vector configuration' (rows of $\mathbf{W}$ in this case). In the R-mode case, such variables are the most independent of the set, and in the Q-mode such samples are compositionally the most divergent. Thus the first estimate of the extremes could be identified by the highest absolute loadings in each of the columns of (n×r) $\mathbf{L}^v$. Suppose then that the (r×p) submatrix $\mathbf{W}_1$ contains the rows of $\mathbf{W}$ with the highest loadings in each of the r columns of $\mathbf{L}^v$. That is, of all the rows of $\mathbf{W}$, each row of $\mathbf{W}_1$ makes the smallest angle with one of the rows of $\mathbf{B}^v$. The rows of $\mathbf{L}^v$ which correspond to $\mathbf{W}_1$ constitute a nonsingular submatrix (r×r) $\mathbf{L}_1{}^v$ such that,

$$\mathbf{W}_1 = \mathbf{L}_1^v\mathbf{B}^v$$

hence,

$$\mathbf{B}^v = \left(\mathbf{L}_1^v\right)^{-1}\mathbf{W}_1$$

and so from equation (2.4),

$$\mathbf{W} = \left[\mathbf{L}^v\left(\mathbf{L}_1^v\right)^{-1}\right]\mathbf{W}_1$$

$$= \mathbf{L}_2\mathbf{W}_1 \tag{2.5}$$

From this equation, it is evident that the unit vectors representing the set of n geological samples have each been expressed as linear combinations of a fixed subset containing r

of their members. Although Imbrie did not make a clear statement of the equations to be solved or of any constraints that would apply to any solution such as equation (2.5), it is manifestly clear that the aim of his procedure was to identify that submatrix $\mathbf{W}_1$ of $\mathbf{W}$ such that all the elements of matrix $\mathbf{L}_2$ in equation (2.5) would be non-negative. That, after all, would be the unique solution in terms of extreme vectors for $\mathbf{W}$. It is equally clear that in general, $\mathbf{W}_1$ does not exist. For example, if the rank of $\mathbf{W}$ were exactly 3, there is no reason why there should be 3 vectors of $\mathbf{W}$ which would define the vertices of a spherical triangle whose boundaries would contain all the remaining points.

Imbrie did not set great store by the recovery of a mixture representation $\mathbf{X} = \mathbf{L}_3\mathbf{X}_1$ in which $\mathbf{W}$ and $\mathbf{W}_1$ are transformed back into $\mathbf{X}$ and $\mathbf{X}_1$ (the corresponding submatrix), and $\mathbf{L}_2$ is transformed into a matrix of mixture coefficients (see Terminolgy) $\mathbf{L}_3$. Indeed, his second program for the resolution of the rows of $\mathbf{W}$ into components $\mathbf{L}_2$ with respect to oblique reference vectors $\mathbf{W}_1$ stopped at that point. His reasons were that in most geological work it is the pattern of regional variations which is of interest rather than the exact numbers, and since the required transformation is linear, nothing of value is achieved by the calculation. The fact is, a sufficient condition for the recovery of the representation $\mathbf{X} = \mathbf{L}_3\mathbf{X}_1$ from $\mathbf{W} = \mathbf{L}_2\mathbf{W}_1$, is that each row vector of $\mathbf{X}$ is a composition (see equation (3.16)). If each row of $\mathbf{X}$ is not a composition, then the resolution of $\mathbf{X}$ into mixtures of the rows of the submatrix $\mathbf{X}_1$ may not be possible. He did include a brief sentence on the transformation of equation (2.5) into a linear relation between the original object vectors. A complete computation to achieve his intended solution is as follows. If the (r×r) diagonal matrix $\mathbf{R}_1$ is the submatrix of $\mathbf{R}$ (defined above) which corresponds to $\mathbf{X}_1$ then $\mathbf{X} = \mathbf{R}\mathbf{W}$ and $\mathbf{X}_1 = \mathbf{R}_1\mathbf{W}_1$. Equation (2.5) can be transformed into,

$$\begin{aligned} \mathbf{X} &= \left(\mathbf{R}\mathbf{L}_2\mathbf{R}_1^{-1}\right)\left(\mathbf{R}_1\mathbf{W}_1\right) \\ &= \mathbf{L}_3\mathbf{X}_1 \end{aligned} \tag{2.6}$$

From the point of view of computer algorithm preparation, there are two important observations to make about this equation. The first is that, if **X** is a matrix of compositions, then the simpler and equivalent operation for constructing $\mathbf{L}_3$ is to divide each row of $\mathbf{L}_2$ by the sum of its components. This amounts to an elementary row operation on both sides of equation (2.5) which is demonstrated in Chapter 3. The second is that if $\mathbf{W}_1$ has been constructed by some method so that it does consist of unit row vectors but it is not a submatrix of **W**, then $\mathbf{R}_1$ does not exist. Again, this is not a problem if **X** is a matrix of compositions, since any point on the unit hypersphere can be projected onto the hyperplane defined by the datapoints of **X**.

In Imbrie's contrived Q-mode example, he had constructed a (10×10) array in which the latter 7 object vectors were convex combinations (mixtures) of the first 3, hence $r = 3$ by design. Also, since the first 3 object vectors were created as compositions (each summing in fact to 100%), all object vectors were compositions (see Chapter 3). Consequently his computer programs progressed from equation (2.1) through to (2.5) interrupted only by the nomination of reference vectors 1,2,3 to execute an oblique rotation. The selection of these vectors was based on an inspection of the largest elements in the columns of the varimax rotated loading matrix (and foreknowledge). An important convention of R-mode factor analysis was adopted, to become a standard device of Q-mode 'factor' analysis also. From the well-known result that the sum of the eigenvalues of the (n×n) similarity matrix equals its trace n, the cumulative proportions of n (the 'sum of squares') for each of the eigenvalues were tabulated. Thus the apparent dimensionality k of the data could be assessed by the percentage of the total variability about the origin accounted for by the first k orthogonal 'factors'. For the contrived data, the sum of the first 3 eigenvalues was 10 which represented 100% of the possible total.

Imbrie's second illustration was based on a (31×6) array of real data which can only be described as very small. The exact rank r of the data matrix appeared to be 6 (on

inspection of the table of eigenvalues). However, Imbrie chose the approximate rank k to be 3 because several entries on the 3rd varimax 'factor' were nearly equal to those on the first two. He described entries on the 4th as trivial as would have been those on the 5th and 6th axes. That is, in equation (2.2), it may be supposed that the last 3 eigenvalues were negligible and so $\mathbf{L}_0$ was defined to contain just the first three columns of $\mathbf{U}\Psi^{1/2}$. Hence for Imbrie's purposes, equations (2.3) to (2.6) would have become approximations. Thus, Q-mode 'factor' analysis was to be employed to approximate a matrix of rank p by the linear combinations of a matrix of rank $k < p$.

Although the largest absolute loadings in the first 2 columns of the varimax rotated matrix (31×3) $\mathbf{L}^v$ did not identify true extremes in the (31×6) matrix $\mathbf{W}$, the second largest loadings purportedly did. Imbrie's criterion for an extreme vector was that no loading on it be greater than 1. He ignored another criterion namely, that no loading on any other reference vector be negative (which would also pose difficulties in interpretation). Eight of the loadings on two of the reference vectors for this illustration were negative, three of them quite large, indicating that the third vector was not extreme. The precise notion of a mixture and the implied constraints of the convex combination were not set out in the document. Towards the end, Imbrie remarked of the second illustration that 'all wells (vectors) can be considered as various mixtures of the reference wells (vectors)'. There remained one serious difficulty which Imbrie did not discuss, and that was the possibility that an array may have been of very low approximate rank k but did not contain a (k×p) submatrix $\mathbf{W}_1$ of extreme row vectors. Later this was to raise the problem of setting out objective procedures for constructing k (unobserved) extreme vectors which would then serve as endmembers. This problem is complicated by the fact that there may be no such vectors or indefinitely many of them. It is for this reason that many authors have been content to report the results of analyses which were concluded at the varimax rotated reference system.

Because Imbrie's program processed the matrix $\mathbf{R}_Q$ in the same way as a correlation matrix, it could accept a maximim of only 70 cases.

The document contained some oddities. Krumbein (1957) was credited with the original use of vector notation with compositional data. A statement which was reinforced a little later by by Imbrie and Van Andel (1964). Krumbein's paper dealt with a 2 component system (A,B) which could be transformed into a composition (P,Q) where P = A/(A+B) and Q = B/(A+B). (The quantities A and B happened to be the thicknesses of sand and shale at a control point). Quoting a result from Kempthorne (1952) that the arcsine square root transformation of P into angle ω stabilized the variance of a sample proportion based on a binomial random sample, Krumbein constructed a complex variable $(\sqrt{B} + i\sqrt{A})$ which defined a vector from the origin of length $(A+B)^{1/2}$ and in a direction ω from the real axis. This vector permitted development of single contour system maps which simultaneously conveyed thickness and composition, and facilitated statistical analysis of the map data. Krumbein promised further presentation and illustration.

Another curiousity was Imbrie's assertion that most Q-mode studies at that time used the product moment correlation coefficient between samples (not variables) as the measure of similarity. He included a table of these 'correlations' to demonstrate their absurdity, although his tabulation of a zero coefficient between a sample and itself was invalid.

The paper by Imbrie and Van Andel (1964) was an expanded descriptive version of Imbrie (1963). Parts of it were almost identical to the latter paper, with the same if not more detailed Q-mode illustration using the same (10×10) array of contrived data, and the same if not more fulsome citations (particularly to Thurstone (1947), Cattell (1952) and Harman (1960)). However, where Imbrie (1963) worked through two examples each of R and Q-mode 'factor' analysis, and covered the theoretical

background to each equally, Imbrie and Van Andel illustrated the absurdity of performing R-mode analyses on a 2-component composition, and repeated Chayes' (1960) warning about the use of the correlation coefficient in the analysis of compositional data in general. Accordingly their paper focussed on Q-mode 'vector analysis' only. In a number of developments, they used the term *endmember* to describe extreme vectors, and discussed the formation of *mixtures* of endmembers right from the start. They recommeded that each column of the final loading matrix be plotted on a *map* on which a loading was associated with its sample location, and so the geographic pattern would be indicated by contours showing the areal distribution of the proportional contribution of each endmember in all samples. If such a map pattern seemed to be random, they advised that the endmember should be disregarded. In the analysis of the (10×10) array of Imbrie's (1963) contrived data, they introduced the barchart (called a 'histogram') to portray the compositions of the 3 endmembers, and employed a ternary plot with the endmembers as vertices, to display the relative positions of the samples with respect to each other and the vertices, in terms of the contrived loadings.

Imbrie and Van Andel recommended but did not illustrate the advantages of changes of scale on the columns (variables) of (n×p) **X** to give major and trace elements equal weight in an analysis. It must be observed that postmultiplication by a non-singular (p×p) diagonal matrix will not effect the true rank $r \leq p$ of **X** but it may profoundly effect the approximate rank k.

Expanding the lexicon of unfortunate terminology, Imbrie and Van Andel called the matrix of loadings (components) on the principal reference axes through O, the 'principal components factor matrix'. The axes themselves they called 'factors' (after Imbrie (1963)) and worse, 'theoretical endmembers'. These vectors could never be endmembers of any sort. Their location and mutual orthogonality guarantee the presence of negative components (concentrations) which are geologically impossible to interpret. Further, since there is an obvious field of investigation into the presence of hypothetical

endmembers in a dataset, the term 'theoretical endmember' should be reserved for such endmembers in theory. The row sums of squares of the loading matrix were called 'communalities', naturally.

In addition to the analysis of the contrived data, Imbrie and Van Andel compared the results of hitherto conventional analyses with those of Q-mode 'vector analyses' of certain heavy mineral suites of the Gulf of California and of the Orinoco-guyana Shelf. Whereas the conventional analyses sought characteristic averages ('pigeonhole classification'), the identification of single extreme samples or endmembers permitted mixtures and gradational sequences to be 'unravelled'. These were demonstrated to good effect on contour maps.

The paper set out a table of the loadings on 6 endmembers, chosen from the data, for a subset of the Gulf of California samples. As was the case with Imbrie's (1960) selection of endmembers fom his data ('real cases') there were negative loadings some of them quite large and in all the columns in this case. No such table was reproduced for the Orinoco-Guayana Shelf data.

The paper by Imbrie and Van Andel (1964) is still frequently cited by authors reporting the results of the analyses of mixtures. It was a superior document to the 'computer program manual' of Imbrie (1963), being considerably more detailed, better illustrated and professionally presented. But principally it is still true that for certain large arrays of actual mixtures, the consequent low approximate rank of a dataset together with a useable loading matrix would readily be revealed by the method pioneered by Imbrie (1963) and described by Imbrie and Van Andel (1964).

Following Imbrie and Van Andel's (1964) paper, reports of the application of Q-mode 'factor' analysis to geological problems started to appear in research journals. A number of computer programs for its implementation also became available. Klovan

(1966) for example, reported on an analysis in which sediment samples were sieved into ten size-ranges which were each measured as percentages by weight of the whole sediment. This was a rather unusual dataset for an analyis of mixtures, given the arbitrariness of the grain-size classes. But it was Klovan's aim to identify the 'depositional environments', if any. His formation of the 'cosθ matrix' and extraction of the first 3 principal axes which were rotated according to the varimax criterion, led him to postulate surf energy, gravitational settling and current energy as the 3 energy types influencing grain-size distributions at a depositional site. Having computed the loadings on each of the three rotated reference vectors, he divided each squared loading by the corresponding 'communality' thus 'normalizing' the factor components which were then plotted on a ternary diagram. Klovan's transformation was equivalent to projecting his estimates onto the surface of the unit sphere. There is a one to one correspondence between the points of that sphere and the plane of the ternary plot whose vertices are the determined by the varimax reference vectors. Klovan asserted that the 'procedure is similar in intent to the oblique projection method of Imbrie and Van Andel (1964)'. That assertion is false.

The next major development was an improvement to the computer algorithm which exploited the algebraic relationships between $\mathbf{W}$, $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{\Psi}$. The basic method, originating with Imbrie (1963) and already described, was to compute directly the eigenvalues and eigenvectors of the (n×n) similarity matrix $\mathbf{W}\mathbf{W}^T$, where n denotes the number of objects. Even today, machine limitations place serious restrictions on the number of objects that can be processed by this method. Klovan and Imbrie (1971) published a computer algorithm that constructed (p×p) $\mathbf{W}^T\mathbf{W}$ where p denotes the number of variables, and hence obtained (p×p) $\mathbf{V}$ and $\mathbf{\Psi}$. The (n×p) matrix $\mathbf{L}_0$ followed by matrix multiplication (see equation (2.3)). Their algebraic summary started with the 'basic factor equation' of Harman (1967), it contained 17 numbered equations, one error and an incomplete derivation of the matrix equation for $\mathbf{L}_0$. The equality of the non-zero eigenvalues of $\mathbf{W}\mathbf{W}^T$ and $\mathbf{W}^T\mathbf{W}$ was not derived but quoted from 'matrix

theory' in mid argument. Just as in the earlier papers, the truth of the results on which the improved computer algorithm was based rested on the unstated singular value decomposition theorem. It will be helpful to demonstrate why this is so.

In the development of the account of the analysis of mixtures in Chapter 3 and of Q-mode methods generally, the integers n and p have been interchanged in most arrays. However, there are always exceptions, and these are prominent in the singular value decomposition of (n×p) matrix **W**. At the risk of being repetitive, this result is now revised with the appropriate changes in notation. As before, (p×p) $\mathbf{\Psi}$ is the diagonal matrix of the p non-zero eigenvalues of (n×n) $\mathbf{WW}^T$ and (n×p) **U** the matrix of corresponding unitized column eigenvectors, so by definition,

$$\mathbf{WW}^T\mathbf{U} = \mathbf{U\Psi}$$

Premultiply both sides of this by (p×n) $\mathbf{W}^T$ to obtain,

$$\mathbf{W}^T\mathbf{WW}^T\mathbf{U} = \mathbf{W}^T\mathbf{U\Psi}$$

and on inspection, $\mathbf{W}^T\mathbf{U}$ is a (p×p) matrix of column eigenvectors of (p×p) symmetric matrix $\mathbf{W}^T\mathbf{W}$. Hence,

$$\mathbf{V} = \mathbf{W}^T\mathbf{U\Psi}^{-1/2} \tag{2.7}$$

is the (p×p) matrix of unitized column eigenvectors of (p×p) $\mathbf{W}^T\mathbf{W}$ (it can readily be verified that $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ (p×p)). Rearranging equation (2.7) to make **W** the subject, the singular value decomposition of rectangular matrix (n×p) **W** is given by,

$$\mathbf{W} = \mathbf{U\Psi}^{1/2}\mathbf{V}^T \tag{2.8}$$

Postmultipying both sides of equation (2.8) by **V**,

$$\mathbf{WV} = \mathbf{U\Psi}^{1/2} \tag{2.9}$$

From equations (2.2) and (2.3),

$$\mathbf{L}_0 = \mathbf{WV} \tag{2.10}$$

and,

$$\mathbf{B}_0 = \mathbf{V}^T \tag{2.11}$$

With the appropriate attention to detail, these results are easily derived when rank($\mathbf{W}$) = rank($\mathbf{X}$) = r ≤ p, as was discussed in Section 1.2.4. With real geological data however, matrices $\mathbf{X}$ (n×p) and $\mathbf{W}$ (n×p) are usually of rank p, and it is their approximate rank k, equal to the number of non-negligible eigenvalues in the diagonal of Ψ which determines, by equations (2.9) and (2.10), the k columns retained in $\mathbf{L}_0$ and the k rows retained in $\mathbf{B}_0$. This latter choice is made by the analyst.

By constructing (p×p) $\mathbf{V}$ and Ψ from (p×p) $\mathbf{W}^T\mathbf{W}$, then (n×k) $\mathbf{L}_0$ and (k×p) $\mathbf{B}_0$ as in equations (2.9) and (2.10), the algorithm by Klovan and Imbrie (1971) would accept at most n = 1500 objects and p = 50 variables for running on a 'moderate-size' computer This was a vast improvement over the maximum of 70 objects for the algorithm written by Imbrie (1963). There were devices in the program for examining differing choices for k, ultimately the final choice was up to the user. The program stopped at the varimax rotated loading matrix, that is at the construction of $\mathbf{L}^v$ and $\mathbf{B}^v$.

There were no notable developments to the Q-mode 'factor' procedure until the publication of three interrelated papers by Miesch (1976a), Klovan and Miesch (1976) and Miesch (1976b). Miesch (1976a) was a 'companion paper' to Klovan and Miesch (1976), but it referred the reader to Miesch (1976b) for a more complete account of 'the extended form of the Q-mode method'. Klovan and Miesch (1976) contained the source code and program descriptions for a modification to the algorithm by Klovan and Imbrie (1971), and a new program to permit 'Q-mode model-building'. Miesch (1976b) covered the first two papers in greater depth but without reproducing the source code. It also provided four extensively worked illustrations based on published data. The three papers taken together focussed on the Q-mode 'factor' analysis of compositional data and presented 'an extension of the method of Q-mode factor (vector) analysis' of data matrices with constant row-sums.

Miesch (1976b) conceded at the outset that Q-mode 'factor' analysis was a misnomer. Nevertheless, he perservered and indeed reinforced the factor analytic interpretation of the mixing model, while simultaneously attempting to exploit the closure constraint (constant row-sum) of compositions in order to acheive solutions with feasible mixture coefficients and endmember compositions. Drawing attention once more to the unsolved difficulties associated with the R-mode analysis of compositional data in general, the authors of the three papers made much of the advantages of the Q-mode processing of data that summed to a constant. Principally these were claimed to be,

(a) The reproduction of 'unbiassed' approximations of the original data in the original units of measurement.

(b) The construction of 'factors' in terms of the original units of measurement.

(c) The construction of 'composition' loadings of the samples on the 'factors'.

(d) A validation procedure for the 'factor model' by means of 'factor variance' diagrams.

(e) The provision for the user of methods to propose hypothetical endmembers and to 'test' their validity.

There was no proof given that the approximations are 'unbiassed'. It is obvious that, if both the original and the approximate data matrices have the same constant row-sum A, then the matrix of residuals must have constant row-sums equal to zero. Perhaps that is what was intended.

A necessary condition for the implementation of the Klovan and Miesch procedure was that the data be of the constant row-sum type, in common units of measurement (proportions, percentages or ppm). It is easily shown (see equation (3.16)) that if, in the equation $\mathbf{X}' = \mathbf{LB}$, the $(n \times p)$ approximation $\mathbf{X}'$ for matrix $\mathbf{X}$ has row-sums all equal to A and the $(k \times p)$ matrix $\mathbf{B}$ of k extreme vectors also has row-sums A, then $(n \times k)$ $\mathbf{L}$ is necessarily a matrix of mixture coefficients. These are the so-called

'composition' loadings of the samples on the 'factors'.

Miesch (1976b) asserted 'that if the factor solution is to be used as a device for summarizing geochemical or petrologic data or for the purposes of sample classification, negative composition scores can be perfectly acceptible'. He maintained that the set of scores for each varimax axes was indicative of the general compositional nature of the 'theoretical' endmember. These remarks probably influenced numerous later authors who were satisfied to stop their analyses at the construction of varimax rotated axes and report the positive and negative 'composition' scores obtained on each axis.

The algebra for the 'cosθ' matrix was presented along with a detailed illustration in Miesch (1976b) of the optional invertible transformation,

$$x^1_{ij} = (x_{ij} - xmin_j)/(xmax_j - xmin_j) \qquad (2.12)$$

which is self-explanatory, and intended to eliminate the distinction in magnitudes between major and minor (trace) elements . The 3 options provided were (i) to make the transformation as set out in equation (2.12), (ii) to define $xmin_j$ to be zero, (iii) to define $xmin_j$ to be zero and $xmax_j$ to be one (the identity transformation). Since the original data summed to a constant the point $X_i(x_{i1},x_{i2},...,x_{ip})$ was transformed under equation (2.12) from one hyperplane into the point $X^1_i(x^1_{i1},x^1_{i2},...,x^1_{ip})$ on a second hyperplane. Transforming the coordinates of the latter into the components of a unit vector was equivalent to projecting the point from the second hyperplane into the point W_i on the surface of the unit hypersphere along a radius through the centre O. (All points must lie in the positive orthant). In such a procedure, there is a one-to-one correspondence between the three points X_i, X^1_i and W_i taken two at a time which accounts for the claimed advantages (a), (b), (c) and (e) above. An approximation to the point W_i for which the 'communality' will be less than 1, can be projected onto the hypersphere and then transformed into a point representing a composition in the original

units. Since $\mathbf{x}^1_i = (x^1_{i1}, x^1_{i2}, \ldots, x^1_{ip})$ is in general no longer a composition vector, the case made that the appropriate coefficient of similarity is the cosine of the angle between two composition vectors will not always apply to $\mathbf{x}^1_i$ and $\mathbf{x}^1_j$.

An extreme vector could be chosen by the user and transformed into a composition in exactly the same way. Since the algorithm was built on that of Klovan and Imbrie (1971), both (n×k) $\mathbf{L}_0$ and (k×p) $\mathbf{B}_0$ were available, and by retaining the full rank (p×p) matrix $\mathbf{V}$, it was possible to form either (1×p) $\mathbf{w}_{new} = \boldsymbol{l}_{new}\mathbf{B}_0$ when (1×k) $\boldsymbol{l}_{new}$ was specified, or $\boldsymbol{l}_{new} = \mathbf{w}_{new}\mathbf{V}$ when $\mathbf{w}_{new}$ was specified, discarding the last [p-k] components from the latter matrix product. So the problem of identifying a geologically interpretible set of endmembers when none were present in the data was confronted by allowing users to experiment with their own choices.

The three papers are probably most important for their approach to assessing the minimum number of endmembers required for a viable representation for (n×p) $\mathbf{W}$ of the form $\mathbf{W}' = \mathbf{L}_0\mathbf{B}_0$. Miesch (1976b) commented that this could be done before the compositions of the endmembers were actually known. The demonstration of the truth of that comment is elementary. Suppose the approximate rank of (n×p) $\mathbf{W}$ is assumed to be k. From equations (2.10) and (2.11), redefine (n×k) $\mathbf{L}_0$ to contain the first k columns of $\mathbf{WV}$, and (k×p) $\mathbf{B}_0$ to contain the first k rows of $\mathbf{V}^T$. Then, the (n×p) matrix approximation $\mathbf{W}'$ for $\mathbf{W}$ is determined uniquely, provided that the chosen set of extreme unit vectors $\mathbf{W}_1$ belong to the subspace spanned by the first k eigenvectors. For such a choice of extremes, $\mathbf{W}_1 = \mathbf{L}_1\mathbf{B}_0$ for some non-singular (k×k) $\mathbf{L}_1$ and so,

$$\begin{aligned} \mathbf{W}' &= \mathbf{L}_0\mathbf{B}_0 \\ &= \left(\mathbf{L}_0\mathbf{L}_1^{-1}\right)\left(\mathbf{L}_1\mathbf{B}_0\right) \\ &= \mathbf{L}_2\mathbf{W}_1 \end{aligned} \qquad (2.13)$$

The derivation of this equation is similar to that for equation (2.5) except that $\mathbf{W}_1$ is of order (k×p) and not necessarily a submatrix of (n×p) $\mathbf{W}'$ (nor in general a submatrix of (n×p) $\mathbf{W}$). The rows of $\mathbf{W}_1 = \mathbf{L}_1\mathbf{B}_0$ are required to be k linearly independent unit vectors which clearly belong to the subspace spanned by the rows of $\mathbf{B}_0$, for which all the elements of (n×k) matrix $\mathbf{L}_2$ in the equation (2.13) are non-negative. With these stated conditions, it is not necessary that $\mathbf{W}_1$ or the (k×p) matrix $\mathbf{X}_1$ of endmembers that corresponds to $\mathbf{W}_1$ be known. Using this result, Renner (1982) published plots of the estimated against the observed values of 10 (major and trace) variables for a 4-endmember representation of 60 marine sediments. Thus the goodness-of-fit of the approximation had been made available for graphical appraisal without knowledge of the actual compositions of the endmembers. (All but one of the plots were remarkably linear, despite transformations of the type (2.12) followed by projections onto the surface of the unit hypersphere. Hence there seemed to be strong evidence for some kind of mixing process).

Miesch (1976a), Klovan and Miesch (1976) and Miesch (1976b) proposed that the goodness-of-fit of the successive approximations corresponding to k = 1,2,3,...., could be illustrated on a 'factor variance' diagram. This was an overlaid plot of the coefficients of determination between the estimated values (columns of $\mathbf{W}'$) and observed values (columns of $\mathbf{W}$) for each of the p variables, against k the number of 'factors'. Miesch (1976b) provided a modest demonstration that the eigenvalues and sample 'communalities' could be misleading where they are used as indicators of the degree to which the 'factor model' can be used to reproduce the original data. In fact, the problem seems to be more serious than he suggested (this problem will be examined in Chapters 3 and 5). Without doubt, if it is the purpose of a mixture analysis to account for the observed values of the variables in terms of parsimonious linear combinations of a small number of endmembers, then all other things being equal, the success of that analysis must be judged by the precision of the estimates. Miesch tried to accomodate variables that seemed to be accounted for only by their own 'unique' factors by

incrementing the estimates due to the remaining endmembers. Geochemically, an element that seems to be accounted for by its own endmember alone is probably just that.

Like the paper by Imbrie and Van Andel (1964) before them, one or other of these three papers continue to be cited in the research literature.

The textbook on geological factor analysis by Jöreskog, Klovan and Reyment (1976) presented an extensive coverage of R- and Q-mode techniques. An entire chapter was devoted to basic mathematical and statistical concepts, and included a derivation of the singular value decomposition of a rectangular matrix. Sixteen years after the paper by Chayes (1960), there was no discussion of the constant sum problem associated with the R-mode analysis of compositional data. The chapter on Q-mode methods asserted that the similarity matrix was the 'mainstay' of Q-mode 'factor' analysis. The text compared the results of 'Imbrie Q-mode factor' analysis, coordinates analysis and correspondence analysis, after each were applied to the (10 $\times$10) array of contrived data originally published by Imbrie (1963). 'Imbrie Q-mode factor' analysis revealed that rank of the data matrix was exactly 3.

Clarke (1978) noted that most 'factor' analysis solutions had failed to 'satisfactorily decompose' sets of mixture data. He observed that it was extremely unlikely that orthogonal 'factors' would all lie in the positive 'quadrant' (orthant) of p-space. Unless they did so, the original compositional data, expressed as linear combinations of such 'factors', could not easily be interpreted as mixtures. He proposed an oblique solution in which the oblique factors (a) belonged to the space spanned by the rows of (k$\times$p) $\mathbf{B}_0$ (see equation (2.13)), (b) lay 'on the edge' of the positive orthant of p-space and (c) were closest to an appropriately chosen set (k$\times$p) $\mathbf{D}$ of reference vectors. If no 'natural' set $\mathbf{D}$ of directions presented itself, Clarke suggested using the initial 'factors' $\mathbf{B}_0$. He set out the algebra, and the source code of

two subroutines to be called from the program published by Klovan and Imbrie (1971). To outline the algebra, let (1×p) $\mathbf{b}_i$ and $\mathbf{d}_i$ be the i-th rows of the the oblique solution matrix (k×p) **B** and the (k×p) matrix **D** of reference vectors respectively. The linear programming method was employed to maximize the objective function $\mathbf{b}_i\mathbf{d}_i^T$ subject to the constraints (i) that $\mathbf{b}_i$ and $\mathbf{d}_i$ were both in the space spanned by the rows of $\mathbf{B}_0$, and (ii) that $\mathbf{b}_i$ was a composition vector. This method must normally determine a solution for $\mathbf{b}_i$ in the 'edge' of the positive orthant. Clearly if $\mathbf{d}_i$ had been chosen in the positive orthant then the linear programming solution would be $\mathbf{b}_i = \mathbf{d}_i$.

It is interesting that Clarke's extreme vectors were constructed in the coordinate hyperplanes in order to enforce the non-negative components in the solutions that were required of extreme composition vectors such as endmembers. If the object of the exercise had been to move oblique vectors (chosen from within the positive orthant) outwards in order to enforce non-negative mixture coefficients, then the coordinate hyperplanes would also have been the boundaries. The non-negativity conditions that must be imposed on the mixture coefficients are as important for the purposes of interpretable solutions as those imposed on the endmember compositions. Clarke's procedure did not enforce both sets of non-negativity constraints, and so feasible complete solutions for the decomposition of mixtures continued to elude the Q-mode 'factor' analysts.

Because it could not guarantee non-negative mixture coefficients, Full, Ehrlich and Klovan (1981) dismissed Clarke's method as 'deficient'. They were also critical because the endmembers chosen by his algorithm would always lie on the edge of the positive orthant when the possibility existed that there were satisfactory solutions 'closer' to the data points. The title of their paper proclaimed 'an objective definition of external endmembers in the analysis of mixtures' and the paper itself reported on a revision to the model-building computer program by Klovan and Miesch (1976). Their criteria for the detection of 'proper' endmembers, shedding the factor analytic

terminolgy, required quite simply that in equation (2.13), the elements of the matrices $\mathbf{L}_2$ and $\mathbf{W}_1$ be positive and that the 'endmembers must minimize in some way the hypervolume of the space defined by the data'. These criteria were included in a section entitled 'Definition Of Endmembers' and were presumably what the authors intended by that title. There is obvious confusion here between the definition of endmembers and an objective method for estimating them. Any set of vectors which satisfied the complete set of non-negativity constraints would define the directions of a set of endmembers. Geological viability would be one criterion for accepting or discarding such a set. The objectivity of the methods available at the time had become a major worry to the rigorous minded. The problem of choosing the number of endmembers was, according to Full, Ehrlich and Klovan (*ibid*, p333) who cited Bezdek, 'acknowledged to be the most critical unsolved problem of cluster analysis'. (No matter that it was not critical, it was not unsolved and it was not cluster analysis, except perhaps to factor analysts!). Having chosen the number of endmembers by any method, to proceed to tinker either with the components of $\mathbf{L}_2$ and solve for $\mathbf{W}_1$, or with the components of $\mathbf{W}_1$ and solve for $\mathbf{L}_2$, with the object of satisfying the non-negativity constraints, was not universally perceived as being 'objective'.

Full, Ehrlich and Klovan (*ibid*) did not include the source code for their computer algorithm, and the description of it was incomprehensible. However it is possible to conclude that their intention was to adjust when necessary, the bounding hyperplanes of the polytope whose vertices were the current extreme points. If, for the position vectors of these extreme points, there existed negative elements in the associated loading matrix, then not all data points were internal to the polytope. This situation was to be rectified by moving the 'edges' of the polytope outwards, parallel to the original 'edges', until the most remote external data points were just internal. This was done iteratively, as the negative loadings associated with each new set of extreme points determined further adjustments. The vertices of the final polytope would define the terminal solution for the endmembers. It is to be assumed that Full, Ehrlich and Klovan

intended the hypervolume of the convex hull of these vertices to be a minimum (which is not what they said). No proof was offered that, from a given set of initial extreme points, the terminal hypervolume was the minimum for all polytopes whose vertices defined extreme vectors that satisfied the non-negativity conditions.

The choice of initial extreme points remained a problem. Full, Ehrlich and Bezdek (1982) proposed another modification to the model-building algorithm of Klovan and Miesch (1976) which would employ the 'fuzzy c-means algorithm' (due to Bezdek) to locate initial extreme points in the space defined by the intersection of the surface of the unit hypersphere with the 'factor' subspace spanned by the varimax axes. (But still, the problem of the determination of the 'proper' number of endmembers they stated 'is acknowledged to be the most critical unsolved problem in cluster analysis' citing Bezdek yet again). Their paper assumed that the 'proper' number of endmembers had been determined. The situation that the proposed algorithm was intended to avoid was that in which an 'abberant' outlier with no apparent relationship to the remainder of the data points would be chosen as an initial extreme point and thus bias the orientation of the entire sequence of iteratively constructed polytopes thereafter. The choice of any single point as an initial extreme, whether it was an outlier or not, seemed to risk introducing a bias. Thus the 'fuzzy c-means' algorithm was intended to generate 'cluster centres' well inside the convex hull of the dataset which would serve as initial vertices of a polytope that would be expanded by the iteration procedure. Such cluster centres would represent the combined properties of many points rather than just one point.

There is no reason to expect that the distribution of the data points on the unit hypersphere will in general permit the identification of (fuzzy) clusters whose centres would provide estimates of the terminal locations, or even the correct orientation of the vertices of the required polytope. Outliers in the data will usually exhibit large residuals in relation to their estimates in 'factor' space and always require attention of one form or another, even possibly exclusion. An outlier with an acceptable residual may be evidence

of a dataset which is merely sparse in its region of p-space. Certainly, no outlier should qualify for selection as an initial extreme unless its associated residual is unexceptional. This raises an interesting point. In the Q-mode 'factor' analysis literature, which spans roughly two decades, very little attention has been paid to an accurate specification of the mixing model and the approximation to it. It is impossible to determine in most papers whether the authors are discussing the raw compositional data or some approximation. Apart form Miesch (1976b), the same symbols are used for both. The Q-mode 'factor' analysts had the 'communality' to test for the presence of outliers. They also had access to the elements of the similarity matrix (n×n) $\mathbf{R}_Q = \mathbf{W}\mathbf{W}^T$ to confirm such a test (consider (n×1) $\mathbf{W}\mathbf{w}_i^T$). One of the advantages of the computation of the appropriate submatrices of $\mathbf{R}_Q$, is that a nearest and furthest neighbours table can be constructed. Outliers will show up on such a table as remote from everything else. The real advantage of the table however, is that when the datapoints occupy a region in p-space approximately shaped in the form of the required [k-1] dimensional polytope, then k distinct groups of objects will be detectable near the vertices, such that objects within a group are near neighbours, and objects in separate groups are far neighbours. One object only from each group will then serve as an initial extreme.

Another modification to the Q-mode 'factor' procedure was published by Leinen and Pisias (1984). Their method was to move each non-feasible 'varimax' axis towards the 'mean' in incremental steps until there were no negative components in the vector representing the axis. It required the determination of the position vector $\overline{\mathbf{x}}$ ('mean composition') of the centroid with respect to the 'varimax' reference axes (transformed into compositions presumably). Each 'varimax' reference axis $\mathbf{b}_i^v$ i = 1,2,...,k was checked for negative components. If at least one non-trivial negative component was present, then an oblique vector was formed, given by $(1 - \zeta)\mathbf{b}_i^v + \zeta\overline{\mathbf{x}}$ where $\zeta = \alpha/100$, $\alpha = 1,2,...,100$. Incrementing α from 0 in steps of 1, the 'composition' of each new oblique vector was tested for the presence of non-trivial negative 'concentrations'. When a vector was reached with only positive and at most, trivially negative components, then the latter if present were set to zero and the vector

thus modified became one of the k required solutions. Suppose $\overline{\mathbf{x}}$ and $\mathbf{b}_i^v$ were the position vectors of the points $\overline{X}$ and B_i^v respectively, then $(1 - \zeta)\mathbf{b}_i^v + \zeta\,\overline{\mathbf{x}}$ would be the position vector of a point on the straight line $\overline{X}B_i^v$. This line would lie in the hyperplane defined by the constant row-sum constraint if this characterized the data. But the procedure could also be applied to the transformed points defined by the rows of (n×p) **W**, on the surface of the unit hypersphere. Like all previous papers, this one relied on the varimax rotation to construct an initial matrix of non-negative loadings, a property which was expected to be preserved in the subsequent loadings, through the progressive tilting of the reference vectors towards the centroid.

The title of the paper by Leinen and Pisias (*ibid*) was 'An objective technique for the determining endmember compositions and for partitioning sediments according to their sources'. In this context, 'partitioning' implied the determination of feasible mixture coefficients which was not fully discussed. It must be assumed that the relationship between the 'varimax' vectors and the selected oblique vectors is in general non-singular and permits the substitution for the 'varimax' vectors in the matrix approximation (see for example, equation (2.13)). The real issue however, was objectivity. Leinen and Pisias stated repeatedly that theirs were objective means of determining endmembers. They incorrectly dismissed the method based on the 'objective definitions' of Full, Ehrlich and Klovan (1981) on the grounds that it required the presence of 'pure' (undefined) endmembers within the dataset, and that it was unlikely to construct endmembers with zero concentrations for some of the variables. They also criticized the approach by Clarke (1978) on much the same grounds as had Full, Ehrlich and Klovan (1981). The former criticism resulted in a terse rebuttal. Full and Ehrlich (1986) pointed out that the paper by Full, Ehrlich and Klovan (1981) made it explict on at least six different occasions that it was not necessary to have sampled endmembers in order to execute their procedure. They also remarked that the majority of their solutions produced endmembers on the 'edges of the positive orthant' and therefore possessed zero components. Full and Ehrlich then went on to criticize Leinen and Pisias

for permitting negative 'abundances' (mixture coefficients), for assuming that the mean vector is a meaningful measure for a multivariate collection of samples, and finally for assuming that the varimax factors contain explicit or unambiguously useful information. What is particularly interesting about this last criticism is that it ever so briefly challenged the universally accepted procedure for rotating the principal axes of the initial Q-mode 'factor' solution.

Leinen and Pisias (1986) followed Full and Ehrlich (1986) with a rejoinder

2.3 NORMATIVE ANALYSIS AND LINEAR PROGRAMMING

Because compositional data must lie in the positive orthant of p-space, Q-mode 'factor' analysts had assumed since the earliest paper by Imbrie (1963) that rotation of the orthogonal 'principal factors' according to the varimax criterion would result in an (n×k) matrix $\mathbf{L}^{v}$ of predominantly non-negative loading coefficients. Such negative entries as existed should have been negligible and were treated as zeros. So although the loadings on the varimax rotated axes were originally intended by Imbrie (*ibid*)to expose extreme samples in the data, the construction of $\mathbf{L}^{v} > \mathbf{0}$ whenever that was accomplished was also half a solution to the whole problem of enforcing all the non-negativity constraints.

Another principal was maintained by the application of the varimax criterion. Once the number k of endmembers had been specified, the analytical procedure which concluded with the varimax rotation of the principal axes was seen to be totally objective. It produced the unique varimax 'decomposition' of (n×p) $\mathbf{W}$ by a sequence of essentially optimizing algorithms (first on the sums of squared projections (1.40) then on the variance of the squared loadings). But objective strategies for the analytical rotation of the 'varimax factor vectors' into an oblique configuration conforming to the non-negativity constraints set out by Full, Ehrlich and Klovan (1981) seemed usually to be defeated by real data.

Heath and Dymond (1977) completely bypassed the Q-mode 'factor' procedure, both for assessing the approximate rank of their particular compositional dataset and for seeking feasible extremes by simply specifying the number and composition ratios of the required endmembers.

The samples used for the Heath and Dymond (*ibid*) study were a subset of size 22 from almost 200 surface samples taken from the Western Nazca Plate. In order to

partition each element among four possible sources (which were described as Detrital, Hydrothermal, Hydrogenous (or Authigenic), and Biogenous), Heath and Dymond constructed simple algebraic relations based on 'interelement relations and on previous knowledge' of the Northwestern Nazca Plate geochemistry. One of these relations made the Biogenous element ratios redundant so that, rather than dwelling in detail on the particular representation reported in their paper, the method for the general case is described below.

Leinen (1987) explained that 'normative partitioning techniques estimate element contributions to a mixture from various endmember sources based on ratios of the elements to a key element. An element which is strongly concentrated in a source is usually chosen as the normalizing element'.

At the heart of the normative analysis is a (k×p) matrix of element ratios $\mathbf{B}_N$ which correspond to the (k×p) matrix of endmembers **B**. The procedure is quite readily appreciated if argued backwards. Let (n×p) matrix **X'** contain the estimated sample concentrations, let (n×k) **L** be a matrix of mixture coefficients and let (k×p) matrix **B** contain the estimated endmember concentrations then,

$$\mathbf{X}' = \mathbf{L}\mathbf{B} \tag{2.14}$$

In each endmember (row of **B**), there is a 'normalizing element' distinct from that in each of the other endmembers. Define the (k×k) diagonal matrix $\mathbf{R}_N$ to contain the concentration of the normalizing element of the i-th endmember in the i-th diagonal position for each i = 1,2,...k.

Then,

$$\mathbf{X}' = \left(\mathbf{L}\mathbf{R}_N\right)\left(\mathbf{R}_N^{-1}\mathbf{B}\right) \tag{2.15}$$

and so,

$$\mathbf{X}' = \mathbf{L}_N\mathbf{B}_N \tag{2.16}$$

The (k×p) matrix $\mathbf{B}_N$ is the array of 'elemental ratio coefficients'. Naturally, the entry in the i-th row of $\mathbf{B}_N$ corresponding to the normalizing element is 1. It was this matrix that Heath and Dymond (1977) and later Dymond (1981) actually specified in their respective papers from considerations of prior knowledge and on inspection of the data.

The paper by Heath and Dymond (1977) seemed to be a prelude to the 1981 paper by Dymond. The two papers presented studies of Nazca Plate Sediments. Both had four conjectured 'sources' (endmembers) in common. The first paper was based on a very few samples however, and the elements of $\mathbf{L}_N$ were implied by the collection of simple algebraic relations, as were the elements of $\mathbf{R}_N$. These relations permitted the recovery of the decomposition (2.14) (*op. cit.*, Table 4) but would probably have been unreliable for a larger study. The results of a Q-mode 'factor' analysis were included towards the end of the paper to confirm what was described as 'essentially a form of normative analysis'.

Dymond (1981) reported on a much larger study. The data consisted of 425 samples selected from cores during the cruises conducted by Oregon State University and Hawaii Institute of Geophysics as part of the Nazca Plate Project. Nevertheless, only 8 variables were analysed, and no mention was made of the accomodation into the analysis of the 50 samples which had at least one missing value. A 'normative sediment analysis model' was employed to evaluate 'five components of sediment'. These were defined *a priori* as (1) Detrital, (2) Hydrothermal, (3) Biogenic, (4) Hydrogenous (Authigenic), and (5) Dissolution Residue. The 'elemental ratio coefficients' matrix corresponding to these five sources (endmembers) and denoted here by (5×8) $\mathbf{B}_N$ was specified (Dymond (1981, Table 3)). That is the 40 matrix elements of $\mathbf{B}_N$ were chosen by Dymond. The overdetermined system to be solved would appear to have been,

$$\mathbf{X} \approx \mathbf{L}_N\mathbf{B}_N \qquad (2.17)$$

in which the matrix on the left is the (425×8) matrix of original compositional data. The components of the i-th row of **X** are given by 8 equations (if there are no missing values) in the 5 unknown elements of the i-th row of (425×5) $\mathbf{L}_N$. Dymond solved these equations for the 425 rows of $\mathbf{L}_N$ by linear programming, the analytical details of which are described in the next section (it may be assumed that a solution was obtained for all cases since the number of missing values per sample did not exceed 3). The non-negativity constraints of the linear programming method assured Dymond of the feasible solutions for each row of $\mathbf{L}_N$ that he was looking for. He also found by trial and error that scaling up the equations for the trace elements by a factor of 20 produced the most satisfactory sums for the residuals. Finally, the row-sums of the (425×8) matrix of compositional data were not constant (note that a vector of compositional data does not imply a composition). Therefore, since the (5×5) diagonal matrix $\mathbf{R}_N$ (equation (2.15)) was unknown, it was not possible to derive equation (2.14) back via equations (2.16) and (2.15). Dymond overcame that obstacle by specifying also the concentrations of the normalizing elements for each endmember (Dymond (1981, p.143)), which consequently defined $\mathbf{R}_N$. Those concentrations were 'taken from the same literature sources that were used to obtain the elemental ratios'.

Although Dymond (1981) cited Narula and Wellington (1977) as the reference for his particular application of the linear programming method, a complete formulation of the actual linear programming problem to be solved appeared in an appendix of the paper by Dymond *et al.* (1984), together with an iterative procedure for adjusting putative endmembers to maintain the non-negativity constraints. This 1984 paper, which reported on an analysis of ferromanganese nodules from the National Science Foundation supported Manganese Nodule Program, is examined in Chapter 4. Dymond *et al.* described it as a 'normative nodule analysis' even though the appendix does not deal with element ratios. Nor do they appear anywhere else in the paper. The appendix does state that the compositions of the endmembers must be specified, and echoes the statement on page 938 that the extreme samples within the dataset were assumed to have

compositions close to the proposed 'true endmembers'.

2.3.1 Partitioning by Linear Programming

Consider the overdetermined system (not in general an equality) relating a given (1×p) compositional vector **x**, an unknown (1×k) vector of mixture coefficients $\boldsymbol{l}$, and a given (k×p) matrix **B** of the concentrations of k endmembers $\mathbf{b}_1,\mathbf{b}_2,...,\mathbf{b}_k$, given by

$$\mathbf{x} \approx \boldsymbol{l}\mathbf{B} = \sum_{j=1}^{k} l_j \mathbf{b}_j \qquad (2.18)$$

The system (2.18) determines p equations for the components of the estimate of **x** in the k (< p) unknown mixture coefficients $l_1,l_2,...,l_k$. The 'partitioning' problem is to find a feasible solution for the mixture coefficients. (The early papers by Dymond set out to solve not the system (2.18) but rather the system $\mathbf{x} \approx \boldsymbol{l}_N\mathbf{B}_N$ in which **x** and $\boldsymbol{l}_N$ are corresponding rows of **X** and $\mathbf{L}_N$ respectively in equation (2.17) (see Dymond (1981)). There is no loss in generality in proceeding with the problem as stated).

Dymond (1981), Dymond *et al.* (1984) and Owen (1987) stated one way or another that the least squares solution $\boldsymbol{l}^*$ for $\boldsymbol{l}$ which minimizes the sum of squares of the residuals below,

$$\sum_{j=1}^{p} (x_j - x_j^*)^2 \qquad (2.19)$$

where $\mathbf{x}^* = \boldsymbol{l}^*\mathbf{B}$ (exactly), is unusable because a non-negativity constraint can not be imposed on each of $l_1^*,l_2^*,...,l_k^*$. Instead, they recommended the linear programming solution $\boldsymbol{l}''$ for $\boldsymbol{l}$ which minimizes the sum of the absolute values of the residuals

$$\sum_{j=1}^{p} |x_j - x''_j| \qquad (2.20)$$

where $\mathbf{x}'' = \boldsymbol{l}''\mathbf{B}$ (exactly), given all the required non-negativity constraints.

The least squares solution will be fully discussed in Chapter 3, but at this point a geometrical comparison between the two approaches is revealing. Suppose the endmember vectors span a k-dimensional space S, a subspace of which is the convex cone $C = \{\mathbf{y} : \mathbf{y} = \mathbf{aB}, (1\times k)\ \mathbf{a} \geq \mathbf{0}\}$. The point X whose position vector with respect to the origin O is $\mathbf{x}$, is a fixed datapoint. Minimizing expression (2.19), is to locate the point X*, defined by $\mathbf{x}^*$, which must be in the space S and also lies on the hypersphere with centre X, whose radius is a minimum. Minimizing expression (2.20), is to locate the point X", defined by $\mathbf{x}''$, which must be in space C and also lies on the cross polytope with centre X, whose diagonal is a minimum. Whether X* is also in C or not, it is the point in S which is closest to X. The diagonals of the cross polytope are parallel to the $Ox_1x_2...x_p$ reference system in Euclidean p-space. In general, a solution in the interior of C will be where a vertex of the cross polytope touches C from p-space. The position of that point is determined by the orientation of the reference axes in relation to C. In general no diagonal (axis) will be orthogonal to C. (As an aid to visualizing the form of the cross polytope, let $y_j = x_j - x''_j$ $j = 1,2,..,p$, translating the reference system. In 2 dimensions, $|y_1| + |y_2| = d$ defines the sides of a square whose four vertices have (y_1,y_2) coordinates $(\pm d,0)$, $(0,\pm d)$ respectively, rather like a plane diamond. In 3 dimensions, $|y_1| + |y_2| + |y_3| = d$ defines the sides of an octahedron, with its six vertices also on the axes, given by $(\pm d,0,0)$, $(0,\pm d,0)$, $(0,0,\pm d)$, rather like a solid diamond. Minimizing expression (2.20) creates the 'diamond' with smallest possible diagonal 2d. When $\boldsymbol{l}'' > \mathbf{0}$, X" is a vertex of this 'diamond').

If the linear programming techniques advocated by Dymond (1981), Dymond *et al.* (1984) and Owen (1987) are employed to solve the overdetermined system (2.18) for $\boldsymbol{l}$, then the resultant estimate $\mathbf{x}''$ (where $\mathbf{x}'' \in C$) is not in general the position vector

of the nearest point X* in S to X unless $\mathbf{x} \in C$, in which case $\mathbf{x}'' = \mathbf{x}$ (as it is with the least squares approach). It is proven a little further on that when $\mathbf{x} \notin C$ the linear programming method, which minimises the absolute error sum given all the non-negativity constraints, will solve $q \leq k$ of the p equations implicit in (2.18) exactly. The number q being the number of non-zero loadings obtained in the solution $\boldsymbol{l}''$. Therefore $q \leq k$ element concentrations are determined exactly, the remainder contribute to error term (2.20). The location of X" relative to the extreme points $B_1, B_2, \ldots, B_k$ (defined by $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_k$) is not as apparent as with the least squares solution X^* (see the next section).

The possibility that one or more of the $B_1, B_2, \ldots, B_k$ are not extreme points for the data is obscured by the linear programming method since no solution for the loading vector $\boldsymbol{l}$ can have negative components.

Dymond (1981) cited an algorithm by Narula and Wellington (1977) for the linear programming solution of the overdetermined system (2.18) subject to non-negativity constraints on all components, and the minimisation of the absolute error sum (2.20). The vector $\mathbf{x}'' = \boldsymbol{l}''\mathbf{B}$ is the linear programming estimate of $\mathbf{x}$, where point X" must belong to convex cone C but not necessarily to H, the convex hull of the points $B_1, B_2, \ldots, B_k$.

The linear programming problem with the required solution is formulated in the following way. First, the j-th error is expressed as the difference of non-negative variables, that is,

$$x_j - x''_j = \upsilon_j - \nu_j, \qquad j = 1,2,\ldots,p \tag{2.21}$$

where $\upsilon_j \geq 0$, $\nu_j \geq 0$. One or other of these will be shown to be zero for each j (see below). Then since $\mathbf{x}'' = \boldsymbol{l}''\mathbf{B}$, there are p constraint equations in the (k+2p) variables $l_1, l_2, \ldots, l_k, \upsilon_1, \upsilon_2, \ldots, \upsilon_p, \nu_1, \nu_2, \ldots, \nu_p$. These are obtained by substituting for x''_j in the

j-th error (equation (2.21) and are given by

$$\sum_{m=1}^{k} l''_m b_{mj} + \upsilon_j - \nu_j = x_j, \qquad j = 1,2,...,p \tag{2.22}$$

The non-negativity conditions are

$$[l_1,l_2,...,l_k, \upsilon_1,\upsilon_2,...,\upsilon_p, \nu_1,\nu_2,...,\nu_p] \geq \mathbf{0} \tag{2.23}$$

and the objective function to be minimized in this case is

$$\sum_{j=1}^{p} (\upsilon_j + \nu_j) \tag{2.24}$$

The three statements (2.22), (2.23) and (2.24) together constitute a linear programming problem (see Hadley (1962) and Bazaraa and Shetty (1979)). To show that it will yield the required solution, it is necessary only to prove the following result.

For any optimal feasible solution, at least one of the pair υ_j,ν_j *is zero for each* j.

Proof:

Suppose $\upsilon_j > 0$ and $\nu_j > 0$ are two components of an optimal feasible solution. Let $\xi = \text{Inf}(\upsilon_j,\nu_j)$ then setting $\upsilon'_j = \upsilon_j - \xi$, $\nu'_j = \nu_j - \xi$, it follows that $\upsilon'_j - \nu'_j = \upsilon_j - \nu_j$ so that υ'_j, ν'_j are also components of a solution, and one or other of the pair is zero. However, $\upsilon'_j + \nu'_j = \upsilon_j + \nu_j - 2\xi \geq 0$, meaning that for the new solution, the objective function has been reduced by 2ξ. Hence its former value can not be a minimum unless $\xi = 0$ which requires one or other of υ_j, ν_j to be zero. It is obvious therefore that, given the non-negativity constraints, minimizing the objective function (2.24) is equivalent to minimizing expression (2.20).

The proof of the next theorem shows that the linear programming solution to the overdetermined system (2.18) of p equations, is simply the exact solution to q ($q \leq k$) of these equations.

The optimal feasible solution to the linear programming problem (2.22), (2.23), (2.24) *will contain* $q \leq k$ *exact solutions to the* p *linear equations* $\mathbf{x} = \boldsymbol{l}\mathbf{B}$, *the remaining* [p-q] *inequations contribute to the error* (2.20).

Proof:

Assuming that the rank of the augmented matrix associated with the set of linear equations (2.22) is p, then the optimal basic feasible solution (see Hadley (1962)) to the linear programming problem (2.22), (2.23), (2.24) will contain at least (k+p) zero values among the (k+2p) variables. If in particular q ($q \leq k$) of the $l_1, l_2, \ldots, l_k$ are non-zero, then at most (p-q) of the u_j, v_j are non zero. That is, there are at most (p-q) linear equations (2.22) in which one or other (but not both) of the u_j, v_j is non-zero leaving at least p-(p-q) = q equations in which both u_j, v_j are zero. Therefore $q \leq k$ of the components of $\mathbf{x}$ will be estimated exactly.

The overdetermined system $\mathbf{x} = \boldsymbol{l}\mathbf{B}$ represents p estimates in the k unknowns $l_1, l_2, \ldots, l_k$. In general, the linear programming problem (2.22), (2.23), (2.24) provides an exact solution $\boldsymbol{l}''$, for k of these p equations, subject to the non-negativity constraints. It will always provide the same feasible solution no matter how ill-fitting the remaining [p - k] linear expressions are. In other words, [p - k] of the components of $\mathbf{x}$ may be made arbitrarily unsuitable without altering the linear programming solution.

CHAPTER 3

THE RESOLUTION OF COMPOSITIONAL DATASETS INTO CONVEX COMBINATIONS OF EXTREME VECTORS

SUMMARY

In this chapter, the mixing or convex model is introduced from first principles, and the most important of its properties and those of its estimates are derived. A geometrical interpretation of the model which should be mimicked by its estimates, is that κ linearly independent endmembers in the positive orthant of Euclidean p-space are the position vectors of the κ vertices (extreme points) of a convex polytope inside which, all points representing mixtures of the endmembers must belong.

An intuitively reasonable result is shown to be true under quite mild conditions. The result is that a convex representation for a composition in terms of a given set of endmember compositions can be uniquely transformed into a convex representation for a subcomposition in terms of the corresponding subcompositions of the given endmembers. The ratios of the components of a subcomposition are equal to the ratios of the corresponding components of its full composition, therefore the relative magnitudes of the components of the endmembers are invariant under such a transformation.

The first problem examined, is that of resolving a single composition into a convex combination of known endmember compositions. This problem can be formulated algebraically as an overdetermined system of equations for the mixture coefficients. It is proposed that the best solution to this system is a vector of mixture coefficients which is parallel to the vector of least squares regression coefficients.

When one or more of these regression coefficients is negative, then at least one of the endmembers is not extreme. A method for adjusting non-extreme endmembers outwards is developed which is based, in part, on the magnitudes of the mixture coefficients of such endmembers. Further, the new set of endmembers always remains in the space spanned by the original set.

Moving on from the single to many compositions, the major problem examined in this chapter is that of constructing a convex representation for a compositional dataset in the absence of any prior information on an underlying mixing process. The proposed solution to this problem contains three distinct stages. The first is the identification of an estimate space for the unknown natural mixtures, and the orthogonal projection of the raw data into that space. The next is the identification of near extremes in the projected dataset which can be treated as initial endmembers. In the third and most complex stage, an iterative algorithm constructs least squares estimates for the matrix of mixture coefficients associated with the initial endmembers. If any mixture coefficients are negative, the algorithm adjusts the endmembers to new positions in the estimate space then recomputes a new matrix of mixture coefficients. This process can be repeated until either all mixture coefficients are positive or an adequate approximate solution has been reached.

An illustration compares four sets of solutions which were obtained for the same raw data by altering the intial extremes and adjustment methods. Two sets of solutions are exact and were the result of the procedure converging, and two were not.

The chapter concludes with a description of some of the computer algorithms that have been created to undertake mixture analyses.

3.1 CONVEX MODELS

A multivariate sample of compositional data $\mathbf{X}$ (n×p) contains measurements on p variables for each of n objects or geological samples. Hence, as has been noted in the last chapter, the vector of variables associated with a single object is a row rather than a column vector. The components of the (1×p) vector $\mathbf{x}$ are the coordinates of the point X in Euclidean p-space, thus $\mathbf{x}$ is also the position vector of X with respect to the origin O. The terms rows, points, object vectors and position vectors will be used interchangeably when there is no ambiguity concerning the object being referred to.

The correct formulation of the traditional geochemical mixing model should be a matrix relation of the form $\mathbf{X} = \mathbf{\Lambda\beta} + \boldsymbol{\varepsilon}$, in which each row of $\mathbf{X}$ (n×p) is composed of a convex combination of the κ rows of the fixed matrix $\boldsymbol{\beta}$ (κ×p) together with an error vector.

The κ rows of $\boldsymbol{\beta}$ must be linearly independent and are the *true endmember* or *source* compositions. The *mixture coefficients* that make up each row of $\mathbf{\Lambda}$ (n×κ) are non-negative and sum to 1. The (n×p) matrix $\mathbf{X}_0 = \mathbf{\Lambda\beta}$ is the true or theoretical array of exact mixtures, and $\boldsymbol{\varepsilon}$ (n×p) is an error matrix.

The rank of $\mathbf{X}_0$ is κ. Therefore its n rows are the position vectors with respect to the origin of n points in a κ-dimensional subspace $\mathcal{A}$ in the positive orthant of Euclidean p-space. Further, these n points are interior to the convex polytope whose vertices are defined by the rows of $\boldsymbol{\beta}$.

If the elements ε_{ij} of matrix $\boldsymbol{\varepsilon}$ are always very small, then the rows of $\mathbf{X}$ are approximately given by linear combinations of the rows of $\boldsymbol{\beta}$, and so the 'approximate rank' of $\mathbf{X}$ is κ, the number of true endmembers.

It is generally the case that, although a mixing process may be believed to be responsible for the geochemical dataset **X**, the number κ and the matrices $\mathbf{\Lambda}$, $\boldsymbol{\beta}$ and $\boldsymbol{\varepsilon}$ are unknown.

When matrix **X** is given and matrix estimates **L** and **B** are obtained for $\mathbf{\Lambda}$ and $\boldsymbol{\beta}$ respectively, then an approximate form of the model is given by the linear relation

$$\mathbf{X} = \mathbf{LB} + \mathbf{E} \tag{3.1}$$

The estimates **L**, **B** and **E** have many properties in common with their theoretical counterparts. For the remainder of this thesis, the four matrices **X**, **L**, **B**, **E** in equation (3.1) will be defined as follows.

(i) **X** (n×p) is a matrix of compositional data. Its components x_{ij} are the concentrations of p minerals in n geological samples usually associated with each of n locations. Accordingly, $x_{ij} \geq 0$ all i,j. Since the data are compositional, then for each i = 1,2,...,n, either,

$$\sum_{j=1}^{p} x_{ij} = A \tag{3.2}$$

or it is possible to introduce a 'fill up' value (Aitchison (1986)),

$$x_{ip+1} = A - \sum_{j=1}^{p} x_{ij} \tag{3.3}$$

Equation (3.2) is often described as the 'constant row sum' property of compositional data. When A = 1, equation (3.2) defines a *composition*. This will usually be the case for the theoretical discussion that follows, but in all applications, the data will be expressed as percentages.

(ii) **B** (k×p) is an estimate of the fixed basis matrix $\boldsymbol{\beta}$. It is of rank k, the estimate for κ, and its components b_{ij} are concentrations of the same mineral types as **X**. Hence $b_{ij} \geq 0$ all i,j and whichever of the equations (3.2) or (3.3) is true for **X**, must also govern the row sums of **B**.

(iii) **L** (n×k) is a matrix of estimated *loadings* or *mixture coefficients* l_{ij} such that $l_{ij} \geq 0$ all i,j, and for each i,

$$\sum_{j=1}^{k} l_{ij} = 1 \tag{3.4}$$

(iv) **E** (n×p) is a matrix of residuals e_{ij}.

An important restriction on these matrix dimensions which governs the interpretation of the model is that $k < p$.

A fifth matrix **X'** (n×p) given by

$$\mathbf{X'} = \mathbf{L}\ \mathbf{B} \tag{3.5}$$

is the estimate of the theoretical array of mixtures $\mathbf{X}_0 = \boldsymbol{\Lambda\beta}$ specified by the model. The rank of **X'** is k the estimate of κ. It is usually assumed that the true errors ε_{ij} are very small. Hence, **X'** should be a good approximation to **X**, whose approximate rank will be k, the estimated number of endmembers.

The matrix **X'** will frequently be referred to as the estimated mixture matrix, just as its rows will be the estimated mixture compositions. It is treated as a surrogate for the observed matrix **X**, particularly for the purpose of geochemical interpretation.

Each row of **X'** is a convex combination of the rows of **B** by Definition (iii).

The matrix of residuals is the difference between the observed data matrix and estimated mixture matrix

$$\mathbf{E} = \mathbf{X} - \mathbf{X}' \tag{3.6}$$

by equations (3.1) and (3.5).

The k rows $\mathbf{b}_1,\mathbf{b}_2,...,\mathbf{b}_k$ of $\mathbf{B}$ are called *endmembers*. They are estimates of the *true endmembers* $\beta_1,\beta_2,...,\beta_\kappa$ which are the κ rows of β. From equation (3.1), the vector of concentrations $\mathbf{x}_i$ for the i-th geological sample may be written

$$\mathbf{x}_i = \boldsymbol{l}_i \mathbf{B} + \mathbf{e}_i = \sum_{j=1}^{k} l_{ij} \mathbf{b}_j + \mathbf{e}_i \tag{3.7}$$

where $\boldsymbol{l}_i$ and $\mathbf{e}_i$ are the i-th rows of $\mathbf{L}$ and $\mathbf{E}$ respectively. Similarly, the corresponding vector of estimated mixture concentrations $\mathbf{x}'_i$ is by equation (3.5),

$$\mathbf{x}'_i = \boldsymbol{l}_i \mathbf{B} = \sum_{j=1}^{k} l_{ij} \mathbf{b}_j \tag{3.8}$$

The endmember vectors $\mathbf{b}_1,\mathbf{b}_2,...,\mathbf{b}_k$ are the position vectors with respect to the origin O of the k vertices $B_1,B_2,...,B_k$ of a convex set (see Hadley (1962)) which is the convex hull H of these points (see Bazaraa and Shetty (1979)).

Various subspaces are defined by altering the constraints on the coefficients of the endmembers in equation (3.8).

(a) When there are no restrictions on the loadings l_{ij}, the endmembers form a basis for a k-dimensional space whose intersection with the positive orthant of p-space, is the *estimate space* S. This space is the estimate of $\mathcal{A}$ the true mixture space. The rows of $\mathbf{X}'$ are the position vectors of points in S.

(b) The non-negativity constraints $l_{ij} \geq 0$ of Definition (iii), determine points in the convex cone C whose generators are $\mathbf{b}_1,\mathbf{b}_2,...,\mathbf{b}_k$.

(c) The sum-to-one requirement (3.4) defines the (k-1) dimensional hyperplane P through the points $B_1,B_2,...,B_k$.

(d) Finally, the intersection of the sets contained in the convex cone C and the hyperplane P which is implied by both the non-negativity constraints and the sum-to-one requirement, is the convex hull H of $B_1,B_2,...,B_k$ (see also, Full, Ehrlich and Klovan (1981) and Full, Ehrlich and Bezdek (1982)). It follows that if k = 2, H is a line segment B_1B_2 (see Figure 3.1), if k = 3, H is a plane triangle $B_1B_2B_3$, and if k = 4, H is a tetrahedron $B_1B_2B_3B_4$, where p is always greater than k. In general, H is a convex polytope by definition (Bazaraa and Shetty (1979)).

Let X'_α and X'_β be two points belonging to H with position vectors $\mathbf{x}'_\alpha$ and $\mathbf{x}'_\beta$ respectively. Then the position vector of any point X' on the line joining these points is given by $\mathbf{x}' = \lambda\mathbf{x}'_\alpha + (1 - \lambda)\mathbf{x}'_\beta$. When $0 < \lambda < 1$, then λ and $(1 - \lambda)$ are positive, the point X' clearly belongs to H and it lies on the line between the other two. If X' is an *extreme point* of H, then there are no distinct points X'_α, X'_β of H for which this configuration is possible.

The vertices B_1, B_2,..., B_k of the convex polytope H are extreme points of H (see Bazaraa and Shetty (1979)).

In the 2-dimensional illustration provided by Figure 3.1 below, the plane of the page is the estimate space S. It is not sufficient to ascertain that $\mathcal{A}$, the true mixture space, is just 2-dimensional. Assuming that the datapoints form a 'fuzzy' line around B_1B_2 on Figure 3.1, a plane at right angles to the page for example would be a very poor alternative estimate space S. Alternatively a plane fixed by the origin and two outliers

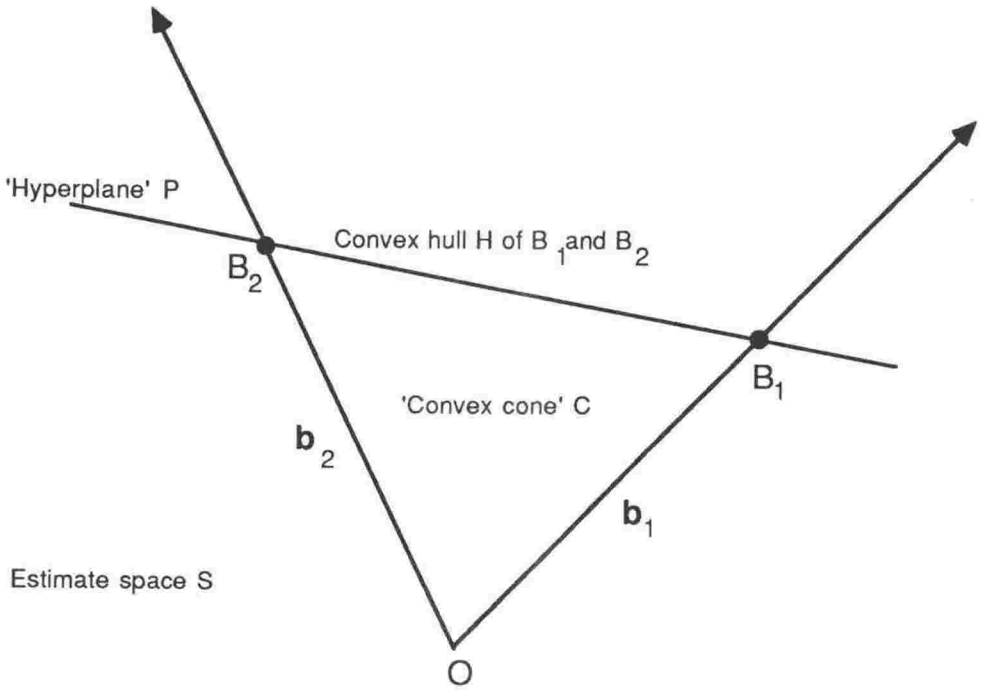
among the datapoints could be remote from the 'fuzzy' line and therefore also a poor estimate space. It is for this reason that the choice of S will later be based on a least squares criterion involving all the datapoints.

It is not important that the points of S do not necessarily represent compositions. Any p-component vector of quantities measured on the same scale can be transformed into a unique composition vector whose direction is unaltered. That is, $\mathbf{x} = \mathbf{w}/\Sigma w_j$ forms the unique parallel composition vector **x** from **w**. The choices for $\mathbf{b}_1$ and $\mathbf{b}_2$ may not be extreme which, in Figure 3.1, would place at least one point estimate outside the cone C. Thus the convex cone C must be a subset of the estimate space S.

All *feasible mixture estimates* $\mathbf{x}'_i$ are convex combinations of the endmember vectors and are represented by points that belong to the convex set H. Hence the rows of $\mathbf{X}'$ define points that form a subset of H and equations (3.5) or (3.8) are estimates of the *convex model* $\mathbf{X}_0 = \mathbf{\Lambda\beta}$. A particular solution of (3.1), (3.5) or (3.8), based on a realization of **X**, is a *convex representation* for the estimate $\mathbf{X}'$ of the mixture array $\mathbf{X}_0$ of geochemical dataset **X**.

By equations (3.4) and (3.8), the composition of the i-th sample is approximately resolved into a mixture of the endmembers in which the proportional contribution to the whole sample of the j-th endmember is l_{ij}. This interpretation has been conveyed historically by the term 'mixing' model, and the computation of the loadings l_{ij} as *linear unmixing*. Equivalently, the i-th sample may be regarded as *partitioned* somewhat in the set-theoretic sense, into k disjoint sources whose relative concentrations identify them respectively with the $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_k$ (see Dymond *et al.* (1984), Leinen and Pisias (1984), Full and Ehrlich (1986) and Leinen (1987)).

Figure 3.1. Subspaces Defined by Linear Combinations of Two Endmembers. (Estimated mixture $\mathbf{x}' = \mathbf{l}\mathbf{B}$). In this 2-dimensional illustration, the estimate space S is in the plane of the page, convex cone C is the region bounded by the line pair OB_1 and OB_2, and the hyperplane P reduces to the straight line through B_1B_2. The convex set H of feasible estimated mixtures is the line interval B_1B_2.



Given the compositional dataset $\mathbf{X}$, the construction of a convex representation (equation (3.1)) strictly requires, first the identification of k-dimensional space S, which implies k the estimated number of endmembers, together with the residual matrix $\mathbf{E}$, and then the solutions if they exist, for the matrices $\mathbf{L}$ and $\mathbf{B}$ in equations (3.1) or (3.5). Since in general, solutions for $\mathbf{B}$ are indeterminate in number, each $\mathbf{b}_i$ i=1,2,...k, should be chosen in some sense as close as possible to the convex hull of the points $X'_1, X'_2, \ldots, X'_n$ (see for example, Full, Ehrlich and Klovan (1981)). Such a choice usually guarantees that the endmember vectors have geologically feasible compositions, at the risk of underestimating their most extreme possible displacements. In geochemical terms, this may mean for example the detection of a clay with some extraneous materials instead of a 'pure' clay, the true extreme.

3.1.1 Subcompositions

Compositional data which are not 'closed' or are incomplete are commonplace in geochemistry. The data may be measured in percentages but not sum to 100% because components of no interest have been discarded or not recorded. For a single sample, such data will be referred to as a *part composition*. The components of a part composition may each be divided by the sum of all the components, thus forming a *subcomposition*. Alternatively, if, for example, all the components are measured on a percentage scale, an additional variable may be formed which is equal to 100 minus their sum. Dividing each of this enlarged collection of variables by 100 would create a *partial composition*.

Convex representations for one or more composition vectors are readily modified for subcollections of the variables. Let the composition $\mathbf{x}$ (1×p) be an approximate mixture given by

$$\mathbf{x} = \boldsymbol{l}\,\mathbf{B} + \mathbf{e} \tag{3.9}$$

(dropping the row subscripts from equation (3.7)). Then the exact mixture is given by

$$\mathbf{x}' = \boldsymbol{l}\,\mathbf{B} \tag{3.10}$$

Consider a vector formed from a subcollection of q of the components of $\mathbf{x}$, where $k < q < p$, and which, without loss of generality, may be taken to be $[x_1, x_2, ..., x_q]$. This vector is a part composition. Denote the first q columns of $\mathbf{B}$ (k×p) by $\mathbf{B}_1, \mathbf{B}_2, ..., \mathbf{B}_q$. By equation (3.10),

$$x'_j = \sum_{m=1}^{k} l_m\, b_{mj} = \boldsymbol{l}\,\mathbf{B}_j \tag{3.11}$$

hence,

$$[x'_1,x'_2,\ldots,x'_q] = l\,[\mathbf{B}_1,\mathbf{B}_2,\ldots,\mathbf{B}_q] \tag{3.12}$$

That is, any linear representation including a convex representation projects orthogonally from p to q-space.

Suppose now subcomposition vectors $\mathbf{x}'^s$, $\mathbf{b}^s{}_1$, $\mathbf{b}^s{}_2$, ..., $\mathbf{b}^s{}_k$ are formed from $[x'_1, x'_2, \ldots, x'_q]$, and the k rows of $[\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_q]$ (k×q). The row-sum t of $[x'_1, x'_2, \ldots, x'_q]$ and the k row-sums s_i of $[\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_q]$ are given by,

$$\sum_{j=1}^{q} x'_j = \sum_{j=1}^{q}\sum_{m=1}^{k} l_m b_{mj} = t \quad \text{and} \quad \sum_{j=1}^{q} b_{ij} = s_i, \quad i = 1,2,\ldots,k \tag{3.13}$$

Assume $t > 0$ and $s_i > 0$, i = 1,2,...k, then for each i and j = 1,2,...,q,

$$x'^s_j = x'_j / t, \qquad b^s_{ij} = b_{ij} / s_i \tag{3.14}$$

The (k×k) diagonal matrix $\mathbf{S} = \mathrm{diag}(s_1,s_2,\ldots,s_k)$ is nonsingular by assumption, so from equation (3.12),

$$(1/t)\,[x'_1,x'_2,\ldots,x'_q] = (1/t)\,l\,\mathbf{S}\,\mathbf{S}^{-1}[\mathbf{B}_1,\mathbf{B}_2,\ldots,\mathbf{B}_q]$$

and this can be written,

$$\mathbf{x}'^s = l^s\,\mathbf{B}^s \tag{3.15}$$

where $l^s = (1/t)l\mathbf{S}$ and $\mathbf{B}^s = \mathbf{S}^{-1}[\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_q]$. Clearly, $\mathbf{x}'^s$ and the k rows of $\mathbf{B}^s$ are all compositions with unit sums by (3.13).

It will now be shown that equation (3.15) is a convex representation for the subcompositional vector $\mathbf{x}'^s$. Since $l^s = (1/t)l\mathbf{S}$,

$$\sum_{m=1}^{k} l_m^s = (1/t) \sum_{m=1}^{k} l_m s_m$$

$$= (1/t) \sum_{m=1}^{k} l_m \sum_{j=1}^{q} b_{mj}$$

$$= (1/t) \sum_{j=1}^{q} \sum_{m=1}^{k} l_m b_{mj}$$

$$= 1$$

by the equation for t on the left of definitions (3.13). Hence the nonnegative loadings l_j^s, j = 1,2,...k, also sum to one. So, the creation of a subcomposition in which the subcollections of q components of $\mathbf{x}'$ and the corresponding q components in each of $\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_k$, all sum to one, results in a convex representation (3.15). That is, provided $\mathbf{S}$ is nonsingular and the rank of $\mathbf{B}^s$ is k, the subcomposition vector $\mathbf{x}'^s$ can be identified with the same, but similarly transformed sources as $\mathbf{x}'$, although the loadings differ. This means that the convex set H^s of feasible estimated mixtures defined by equation (3.15) should be of the same geometrical form as the convex set H, a line segment if k = 2, a plane triangle if k = 3 and so forth.

Aitchison (1986) stated an elementary result (by equation (3.14)), that the ratio x^s_j/x^s_m of any two components of a subcomposition is the same as the ratio x_j/x_m of the corresponding components in the full composition (which accounts for the fixed covariance structure of the 'logratios' (Aitchison (1986, p.65)). It follows from equations (3.14) and (3.15) that, provided $b_{im} \neq 0$,

$$b^s_{ij} / b^s_{im} = b_{ij} / b_{im}$$

The important result that has been established above is that provided $\mathbf{S}$ is nonsingular and the rank of $\mathbf{B}^s$ is k, a convex representation for a composition $\mathbf{x}'$ in terms of a given set of endmember compositions $\mathbf{B}$ can be uniquely transformed into a convex representation for a subcomposition $\mathbf{x}'^s$ of $\mathbf{x}'$, in terms of the corresponding subcompositions $\mathbf{B}^s$ of the given endmembers $\mathbf{B}$. The ratios of the components of a

subcomposition are equal to the ratios of the corresponding components of its full composition, so the relative magnitudes of the components of the endmembers are invariant under such a transformation.

Aitchison (1986, Table 3.1) illustrated the substantial changes in correlations between selected pairs of variables which follow successive movements from a full composition to a number of subcompositions. There is nevertheless one possible value of the correlation coefficient between two variables of any subcompositional dataset $\mathbf{X}^s$ (n×q), which can not change and must equal the correlation between the corresponding variables of the full compositional dataset $\mathbf{X}$ (n×p). That value is 1. Suppose that in the dataset $\mathbf{X}^s$, the ratio $x^s_{ij}/x^s_{im} = v$ for all i = 1,2,...,n. Then the n ordered pairs (x^s_{ij}, x^s_{im}), i = 1,2,...,n, lie on a straight line through the origin with slope 1/v, and therefore have correlation 1. But by the ratio property, it is also the case that $x_{ij}/x_{im} = v$ for all i = 1,2,...,n, so the same result applies to the (j,m)th variables of $\mathbf{X}$. Nearly linear associations, that is, correlations greater than about 0.90 between two components of a composition, occur quite frequently in practice. For example, a correlation of this order between the oxides Al_2O_3 and SiO_2 is a common indicator of a silicate (clay) endmember which, if identified, often implies negligible quantities of these oxides in the remaining endmembers. In any event, the presence of any high correlations between the variables of $\mathbf{X}$ should be reflected by approximately constant ratios for the appropriate components of the endmembers.

The unit sum obtained above for the components of $\boldsymbol{l}^s$ is a special case of a more general result concerning row sums. By equation (3.11) with q = p, the row sum given by

$$\sum_{j=1}^{p} x'_j = \sum_{j=1}^{p} \sum_{m=1}^{k} l_m b_{mj}$$

$$= \sum_{m=1}^{k} l_m \sum_{j=1}^{p} b_{mj} \qquad (3.16)$$

If $\sum_{j=1}^{p} x'_j = \sum_{j=1}^{p} b_{mj} = A$ (any constant > 0), m = 1,2,...,k, then $\sum_{m=1}^{k} l_m = 1$ necessarily. Alternatively, if $\sum_{j=1}^{p} b_{mj} = 1$, m = 1,2,...,k, and one of $\sum_{j=1}^{p} x'_j$, $\sum_{m=1}^{k} l_m$ is equal to 1, then by (3.16) so must be the other.

These results can be exploited in algorithms for constructing the matrices **L** and **B** when the rows of **X'** are compositions. For example, suppose the matrices $\mathbf{L}_0$ and $\mathbf{B}_0$ are exact feasible solutions to the matrix equation **X' = LB**, however, the rows of $\mathbf{B}_0$ are not compositions but unit vectors. Then dividing each of the rows of $\mathbf{L}_0$ and $\mathbf{B}_0$ by their respective row-sums would result in an exact convex representation for **X'** in terms of feasible endmember compositions. This is a much simpler operation than that described by equation (2.6). Moreover equation (2.6) requires the magnitudes of the final endmember vectors, and these may not exist if the endmembers do not belong to **X'**. Since Q-mode factor analysis starts with the transformation of the data into unit vectors, it was considerations such as these that were at the basis of the advantages claimed for processing constant sum data that were made by Miesch (1976a,b) and Klovan and Miesch (1976).

3.1.2 A Note on Partial Compositions

There is an alternative to the derivation of equation (3.15), in which the part composition was transformed into a subcomposition. That alternative is to reconstruct equation (3.12) into a convex combination of partial compositions.

Suppose a (q+1)th component is added as a 'fill up' value to each part composition in equation (3.12). Then each part composition acquires an extra component and becomes a *partial composition*. For the vector $[x'_1, x'_2, ..., x'_q]$, the additional component x'_{q+1} is redefined as follows,

$$x'_{q+1} = 1 - \sum_{j=1}^{q} x'_j$$

Similarly, for the part composition $[b_{m1},b_{m2},...,b_{mq}]$ of the m-th endmember,

$$b_{m\,q+1} = 1 - \sum_{j=1}^{q} b_{mj} \quad \text{for } m = 1,2,...,k$$

Forming convex combinations of both sides of the last equation, from the given vector $\boldsymbol{l}$ of mixture coefficients,

$$\sum_{m=1}^{k} l_m b_{m\,q+1} = \sum_{m=1}^{k} l_m - \sum_{m=1}^{k}\sum_{j=1}^{q} l_m b_{mj}$$

The components of $\boldsymbol{l}$ sum to 1 by definition, so on interchanging the order of the double sum on the left,

$$\sum_{m=1}^{k} l_m b_{m\,q+1} = 1 - \sum_{j=1}^{q} x'_j$$

$$= x'_{q+1}$$

Hence the (q+1) dimensional compositional vectors $[x'_1,x'_2,...,x'_{q+1}]$, and $[b_{m1},b_{m2},...,b_{m\,q+1}]$, m = 1,2,...,k, satisfy equations (3.11), (3.12) but with (q+1) replacing q.

An immediate corollary of this result is that the rank k of an (n×q) matrix of exact mixtures of part compositions, is unaltered by the addition of an (n×1) column of constructed 'fill up' values (which correspond to geochemical residues).

Databases consisting of part compositions, that is, retaining the measurements of q < p of the components of full compositions, are commonplace in geochemistry. In Chapter 5 one such database is analyzed after its conversion to partial compositions, and the endmembers obtained are then transformed into subcompositions. In view of the results of the last two sections, the same endmember subcompositions should be

obtained from a sound mixture analysis whether the subcompositions are formed before or following the analysis.

3.1.3 A Note on Sampling Distributions

A theory for the joint probability distribution of a random composition vector $\mathbf{x}$ given by

$$\mathbf{x} = \boldsymbol{\lambda}\boldsymbol{\beta} + \boldsymbol{\varepsilon} = \sum_{j=1}^{\kappa} \lambda_j \boldsymbol{\beta}_j + \boldsymbol{\varepsilon} \tag{3.17}$$

would usually need to incorporate two distinct components of variation.

The first arises from the sampling procedure. For statistical purposes, geochemists routinely report the collection of 'random' samples of geological specimens such as marine sediments, rock samples and so forth (see, for example, Woronow and Love (1988)). Such reported randomness is usually a perception of the collection method. A designed sampling procedure may for example, follow a uniform probability distribution defined on the region from which the samples are to be taken or a systematic selection of sampling points (stations) uniformly distributed on the region. But other probability sampling methods are also valid. (It should be remarked that most sampling schemes are purposive. Samples are usually recovered from sites which possess particular attributes of interest).

Given any valid probability sampling method, then by the assumed model (3.17), there is associated with each point in the sample space a unique (but unknown) realisation of the mixture loading vector $\boldsymbol{\lambda}$ which is, therefore, a random vector. The unknown joint probability distribution of the components of $\boldsymbol{\lambda}$ is defined on a $(\kappa\text{-}1)$-dimensional hyperplane $\mathfrak{H}$ in the positive orthant of κ-space. It is, in some obscure way, related to both the sampling scheme and the mixing process that is under study. It would be an error to assume that a 'random' collection method implied a

(κ-1)-dimensional uniform distribution defined on $\mathcal{H}$.

The second component of variation involves the additive error vector $\boldsymbol{\varepsilon}$ in equation (3.17). An intuitively more satisfactory way to describe the deviation from the true mixture $\boldsymbol{\mu} = \boldsymbol{\lambda}\boldsymbol{\beta}$, is to suppose each component of $\boldsymbol{\mu}$ is rescaled by the corresponding component of a 'perturbing' vector $\boldsymbol{\rho}$ (Aitchison (1986)) so that,

$$x_j = \mu_j \rho_j / \sum_{m=1}^{p} \mu_m \rho_m \qquad (3.18)$$

If $\mathbf{x} = \boldsymbol{\mu} \circ \boldsymbol{\rho}$ denotes a perturbation of $\boldsymbol{\mu}$ then $\mathbf{x} = (\boldsymbol{\lambda}\boldsymbol{\beta}) \circ \boldsymbol{\rho} = \boldsymbol{\lambda}^p(\boldsymbol{\beta} \circ \boldsymbol{\rho}) = \boldsymbol{\lambda}^p\boldsymbol{\beta}^p$, where $\boldsymbol{\beta}^p$ is the result of perturbing each of the k rows of $\boldsymbol{\beta}$ by the same perturbation vector $\boldsymbol{\rho}$, and $\boldsymbol{\lambda}^p$ is a mixture loading vector by equation (3.16). If $\rho_j, x_j > 0$, j= 1,2,...,p, and $\boldsymbol{\rho}$ itself is the product of many similar independent perturbations, then the random vector $\mathbf{z}$ (1×(p-1)), defined by $z_j = \log_e(\rho_j/\rho_p)$ and hence $\mathbf{y}$ (1×(p-1)) where,

$$y_j = \log_e(x_j/x_p) = \log_e(\mu_j/\mu_p) + \log_e(\rho_j/\rho_p), \quad j = 1,2,...,p-1 \qquad (3.19)$$

will, under certain regularity conditions, follow multivariate normal distributions (implying that $\mathbf{x}$ will follow an additive logistic normal distribution (Aitchison (1986)). Thus provided all matrix elements are non-zero (an unrealistic condition in general), then by equation (3.19) the rows $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ of (n×p) dataset $\mathbf{X}$ can be transformed into the 'logratio' row vectors $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n$ of (n×(p-1)) matrix $\mathbf{Y} = \mathbf{M} + \mathbf{Z}$ where,

$$y_{ij} = \log_e(x_{ij}/x_{ip}) = \log_e(\mu_{ij}/\mu_{ip}) + \log_e(\rho_{ij}/\rho_{ip}), \quad j = 1,2,...,p-1. \qquad (3.20)$$

Under appropriate hypotheses, $\mathbf{Y}$ may be analysed by the family of procedures based on the multivariate normal distribution. But equation (3.20) and therefore the matrix equation $\mathbf{Y} = \mathbf{M} + \mathbf{Z}$ are particular cases of the familiar 'response = signal + noise' model. When the structure of $\mathbf{M}$ is assumed, then the validity of that assumption can be

tested by the consequent properties of **Z**. Ideally those should be of a random sample from a multivariate normal distribution whose mean vector is **0**.

This Chapter however, is concerned principally with the determination of x'_{ij}, the components of the estimated mixture matrix **X'**, which is the first step in solving the mixing problem. Testing the validity of any solution is the next step and that matter will be raised again in Chapters 4 and 5. The severest practical measure of the inadequacy of the estimated mixtures, **X'**, is the proportion of the coefficients of determination (between the observed and estimated mixture variables (see Miesch (1976b)) which are less than some predetermined cutoff value which in this work has been chosen to be 0.5.

3.2 PARTITIONING PROCEDURES

Before examining the general problem of constructing a convex representation (3.1) for a dataset **X** of n samples, the case of the single geological sample is considered first.

The simplest partitioning problem is that where an endmember assemblage **B** (k×p) has been estimated and, given the composition **x** (1×p) associated with a single sample, it is required to find the loading vector ***l***. (1×k). This is the 'linear unmixing' problem (after Full, Ehrlich and Bezdek (1982)) reduced to one case, and it really embodies the two questions,

(a) is the given sample a mixture of the given endmembers, within tolerable errors;

(b) if it is, then what is the contribution l_j of the endmember $\mathbf{b}_j$ to the sample, where j = 1,2,...,k ?

3.2.1 Partitioning by Least Squares

Let **x** be the position vector of the data point X with respect to the origin O. It is proposed in this Section that the best approximation to **x** in the space S spanned by the k rows of **B**, is the position vector of the orthogonal projection of X into S. That approximation will require transforming into a composition. The answers to questions (a) and (b) above are then determined by the precision of the latter approximation.

In the single sample case $k < p$, $n = 1$, **x** (1×p) and **B** (k×p) are known. The the problem therefore becomes that of finding the solution for $\boldsymbol{l}$ (1×k) in an equation of type (3.7) without subscript i as below

$$\mathbf{x} = \boldsymbol{l}\,\mathbf{B} + \mathbf{e} = \sum_{j=1}^{k} l_j \mathbf{b}_j + \mathbf{e} \tag{3.21}$$

When $\boldsymbol{l}$ is obtained, the estimated mixture **x**' is

$$\mathbf{x}' = \boldsymbol{l}\,\mathbf{B} = \sum_{j=1}^{k} l_j \mathbf{b}_j \tag{3.22}$$

The orthogonal projection of the point X *onto the estimate space* S *defines a unique point* X* *in* S *with position vector* $\mathbf{x}^* = \boldsymbol{l}^*\mathbf{B}$, *where* $\boldsymbol{l}^*$ *is the vector of least squares regression coefficients.*

Proof:

Let $\mathbf{x}^* = \boldsymbol{l}^*\mathbf{B}$ be the position vector with respect to O of the orthogonal projection of X onto S. The point X* must be in S but is not necessarily on hyperplane P. Since the rows of **B** span S, any other (1×p) vector $\mathbf{y} \in S$ has the form $\mathbf{y} = \mathbf{aB}$ for some **a** (1×k). It is required that line XX* is orthogonal to S so,

$$(\mathbf{x} - \mathbf{x}^*)\mathbf{y}^T = 0$$

That is,

$$(\mathbf{x} - \boldsymbol{l}^*\mathbf{B})\mathbf{B}^T\mathbf{a}^T = 0 \quad \text{for all } \mathbf{a},$$

which implies

$$(\mathbf{x} - \boldsymbol{l}^*\mathbf{B})\mathbf{B}^T = \mathbf{0} \quad (1\times k),$$

thus,

$$\boldsymbol{l}^* = \mathbf{x}\mathbf{B}^T(\mathbf{B}\mathbf{B}^T)^{-1} \tag{3.23}$$

Matrix $(\mathbf{B}\mathbf{B}^T)$ (k×k) must be nonsingular since **B** is assumed to be of rank k. Hence,

$$\mathbf{x}^* = \boldsymbol{l}^*\mathbf{B} = \mathbf{x}\mathbf{B}^T(\mathbf{B}\mathbf{B}^T)^{-1}\mathbf{B} \tag{3.24}$$

The vector $\boldsymbol{l}^*$ (1×k) given by equation (3.24) is the vector of least squares regression coefficients (Rao (1973)) obtained by minimising

$$\sum_{j=1}^{p} (x_j - x^*_j)^2 \tag{3.25}$$

in the solution of the overdetermined system,

$$\mathbf{x} \approx \boldsymbol{l}\mathbf{B} = \sum_{j=1}^{k} l_j \mathbf{b}_j \tag{3.26}$$

Which completes the proof.

This is an opportune point at which to make an obvious comment. If $\mathbf{x} = \mathbf{a}\mathbf{B}$ for some **a** (1×k), then system (3.26) would become an equation whose augmented matrix would be of exact rank k. By equation (3.23) $\boldsymbol{l}^*$ would be equal to **a**. That is, least squares procedures will construct exact solutions to systems like (3.26) when they exist. This somewhat obvious result permits the employment of a single least squares

algorithm to solve a variety of exact as well as overdetermined systems which arise in the analysis of mixtures. A frequent application for it is to construct the (n×k) matrix of mixture coefficients **L** for the equality $\mathbf{X}' = \mathbf{LB}$ given that the rows of (n×p) $\mathbf{X}'$ belong to the space spanned by the rows of (k×p) **B**.

The sum of squares (3.25) is equal to the squared distance $(XX^*)^2$. Since this is minimised, X* is the nearest point in k-space S to point X.

It is evident that angle XOX* happens also to be a minimum. In the triangle XOX*, the side XX* is normal to OX*, the hypotenuse OX is fixed, and XX* is a minimum distance, so XX*/OX = sin(XOX*) is a minimum as must therefore, be the angle XOX*. It follows that the 'coefficient of proportional similarity' (Imbrie and Purdy (1962)), cos(XOX*), is a maximum.

(A 2-dimensional illustration of the foregoing is provided in Figure 3.2 below).

The least squares approximation **x*** thus has a relative composition which is most similar to **x** among those position vectors of points in S (for detailed discussions of proportional similarity see the Q-mode factor accounts of Imbrie and Van Andel (1964), Klovan (1966), Klovan and Imbrie (1971), Jöreskog, Klovan and Reyment (1976) and Miesch (1976b)).

Returning to question (a) posed at the start of this section, if angle XOX* were zero, then **x** would be an exact mixture of $\mathbf{b}_1,\mathbf{b}_2,...,\mathbf{b}_k$. If angle XOX* were merely 'small', this could be perceived as falling within a 'tolerable error'. (If **x** were in fact an approximate mixture of $\mathbf{b}_1,\mathbf{b}_2,...,\mathbf{b}_k$ but the error vector was not orthogonal to S, then angle XOX* would be less than the true angular error). Finally, if angle XOX* were 'large' then either the error vector was also 'large' or the hypothesis that the sample is a linear combination of the given endmembers would not be plausible.

Assuming that angle XOX* is 'small', there are now two possibilities. Either all $l^*_j \geq 0$ so that X* is in convex cone C, or at least one $l^*_j < 0$ and X* is outside C. In either case, line OX* can be produced onto point X' on hyperplane P by creating $\boldsymbol{l}$ where,

$$l_j = l^*_j / \sum_{m=1}^{k} l^*_m, \qquad j = 1,2,...,k \tag{3.27}$$

The components l_j, j = 1,2,...,k, of $\boldsymbol{l}$ obey equation (3.4) and so the estimated mixture **x'** given by,

$$\mathbf{x}' = \boldsymbol{l}\,\mathbf{B} = \mathbf{x}^* / \sum_{m=1}^{k} l^*_m \tag{3.28}$$

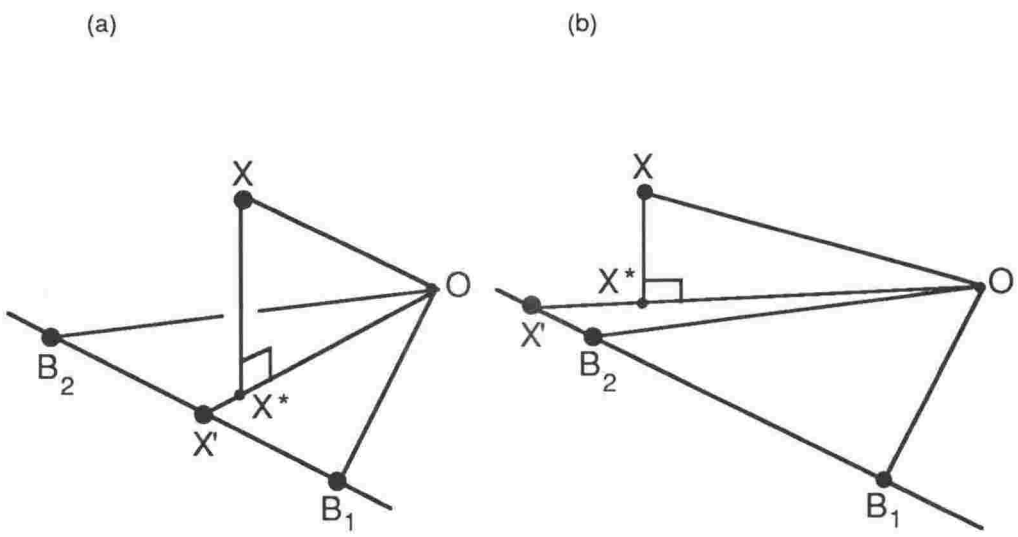
is the position vector of a point X' on hyperplane P by equation (3.27). It follows from equation (3.16) that provided $\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_k$ are compositions, then **x'** is also a composition.

Angles XOX* and XOX' are equal since **x'** is parallel to **x*** by equation (3.28). Hence **x'** remains most similar to **x** and, therefore, the best approximation to **x** among the position vectors of points of hyperplane P.

If all $l^*_j \geq 0$, then by equation (3.27), all $l_j \geq 0$, X' is a point in H, and the problem is solved (see Figure 3.2 (a)). The required partitioning of the sample into the k given endmembers is defined by the components l_j, j = 1,2,...,k of the loading vector $\boldsymbol{l}$ constructed at equation (3.27). That is the answer to question (b) at the start of the section.

If $l^*_j < 0$, then $l_j < 0$ (the denominator of equation (3.27) being positive). The point X' is on the hyperplane P but outside the convex set H, meaning that at least one of the $\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_k$ is not an endmember (see Figure 3.2 (b)).

Figure 3.2. Orthogonal projection of point X onto the estimate space spanned by two endmembers. $\mathbf{x}^* = f^*_1\mathbf{b}_1 + f^*_2\mathbf{b}_2$. Line XX^* is perpendicular to S, which in this case is the plane through OB_1B_2. Hyperplane P is the line through B_1B_2. Convex cone C is the region bounded by the line pair OB_1, OB_2. In a good representation, angle XOX^* would be small. (a) If $f^*_1, f^*_2 > 0$ then X^* is inside C. (b) If $f^*_1 < 0$, $f^*_2 > 0$ then X^* is outside C as shown. In either case, OX^* produced must intersect line B_1B_2 in point X'.



Assuming that the k+1 vectors $\mathbf{x}, \mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_k$ are approximately linearly dependent as above, a new problem arises, namely to find an alternative set of k vectors which define extreme points whose convex hull will include X', and usually $B_1, B_2, \ldots, B_k$. This problem is considered in the next section.

Finally, from equation (3.24) it follows that any point R with position vector (1×p) $\mathbf{r}$ may be projected orthogonally into estimate space S according to the relation

$$\mathbf{r}^* = \mathbf{r}\mathbf{B}^T(\mathbf{B}\mathbf{B}^T)^{-1}\mathbf{B} \tag{3.29}$$

The matrix $\mathbf{B}^T(\mathbf{B}\mathbf{B}^T)^{-1}\mathbf{B}$ is an orthogonal projection operator (Rao (1973)). It may be employed in order to construct the nearest point in S to any point outside S, otherwise it

behaves as an identity operator. Postmultiplying any matrix of constructed vectors such as adjusted endmembers by this operator guarantees that all points remain in S.

3.2.2 Partitioning by Linear Programming

The incorporation of the overdetermined system (3.26) into an appropriately formulated linear programming problem has been discussed in Section 2.3.

The principal advantage of the linear programming solution is that, for each sample vector $\mathbf{x}$ and from a specified set of (feasible) endmembers $\mathbf{b}_1,\mathbf{b}_2,...,\mathbf{b}_k$, the non-negativity constraints guarantee a feasible solution $\boldsymbol{l}''$ for the loadings $\boldsymbol{l}$. The approximation $\mathbf{x}'' = \boldsymbol{l}''\mathbf{B}$ to $\mathbf{x}$ is also feasible.

The disadvantages are,

(i) in general, when the point X'', whose position vector is $\mathbf{x}''$, is in the interior of the convex cone C, it is not the closest point in C to X,

(ii) in general, when X'' is on the surface of C, at least one of $\mathbf{b}_1,\mathbf{b}_2,...,\mathbf{b}_k$ is not an endmember and the components of the solution $\boldsymbol{l}''$ do not measure the magnitude of the consequent discrepancy.

In the light of these disadvantages, it will be seen that question (a), posed at the start of Section 3.2, is not readily resolved by this method. The answer to question (b), of course, is $\boldsymbol{l}''$ as described in Section 2.3.1.

Dymond *et al.* (1984) described an iterative algorithm which adjusted the endmembers in order to account for the errors resulting from the linear programming partition. The algorithm works only for a data matrix $\mathbf{X}$ (n×p) in which $n \geq k$. Briefly,

it amounts to a least squares solution for $\Delta\mathbf{B}$ to the overdetermined system $\mathbf{L}(\mathbf{B} + \Delta\mathbf{B}) \approx \mathbf{X}$, when $\mathbf{B}$ has been specified, each row of $\mathbf{L}$ has been constructed by the linear programming method, and $\mathbf{X}$ of course is known. The substitution of $\mathbf{B}' = \mathbf{B} + \Delta\mathbf{B}$ for $\mathbf{B}$ in the system (3.26) establishes a new set of constraints for another linear programming solution for a new loading matrix. From that point, an iterative cycle has been defined which can be repeated until some error criterion is satisfied.

There are aspects of this process which are unsatisfacory. These will be mentioned in Section 3.4.3.

3.3 ENDMEMBER ADJUSTMENT

Remaining with the case of the single sample, suppose vectors $\mathbf{b}_1,\mathbf{b}_2,...,\mathbf{b}_k$ are a given set of endmembers for a geochemical dataset. It is assumed that the space S spanned by $\mathbf{b}_1,\mathbf{b}_2,...,\mathbf{b}_k$ is the best fitting subspace of Euclidean p-space for the dataset. However, in the particular case of a compositional vector $\mathbf{x}$, the least squares partition results in loading vector $\boldsymbol{l}$ (by equations (3.23), (3.27)) and estimated mixture $\mathbf{x}'$ (equation (3.28)) for which angular error XOX' is small, but where a number s of the components of $\boldsymbol{l}$, denoted by $l_\alpha,l_\beta,...,l_\delta$, are less than zero $0 \le s < k$. So that within a tolerable error, the composition of $\mathbf{x}$ is a linear combination of any set of basis vectors of S but as noted earlier, X' lies outside the convex hull of $B_1,B_2,...,B_k$ indicating that at least one of $\mathbf{b}_1,\mathbf{b}_2,...,\mathbf{b}_k$ is not an endmember.

The location of X' relative to B_1, B_2 ,..., B_k determines the components of the loading vector $\boldsymbol{l}$. Since l_α, l_β, ..., l_δ are less than zero, line segments $X'B_\alpha$, $X'B_\beta$, ..., $X'B_\delta$ are intersected internally by s bounding hyperplanes containing the faces of the convex polytope $B_1B_2...B_k$. If subspace S is to be preserved as the estimate space for

the dataset, then the construction of endmembers for $\mathbf{x}$ and possibly $\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_k$, requires moving these bounding hyperplanes outwards within S until X' is no longer an exterior point. This means that the s points $B_\alpha, B_\beta, .., B_\delta$ will be fixed while the remaining (k-s) vertices of the polytope must be moved outwards (see Figure 3.3 below).

(Note: a bounding hyperplane through the $q < k$ vertices $B_a, B_b, ..., B_d$ of the convex polytope H, is the set of points $\{\mathbf{y} : \mathbf{y} = l_a\mathbf{b}_a + l_b\mathbf{b}_b + ... + l_d\mathbf{b}_d, l_a + l_b + ... + l_d = 1\}$. The subset of this for which $l_a, l_b, ..., l_d \geq 0$, is clearly also a convex polytope. A face of H is the convex hull of (k-1) of the vertices and is contained in the bounding hyperplane through those vertices).

Setting the s negative components of $\boldsymbol{l}$ to zero and rescaling the remainder to sum to one creates a corrected loading vector $\boldsymbol{l}^0$ from which the violation of the non-negativity constraint has of course, been removed. The new mixture $\mathbf{x}^0 = \boldsymbol{l}^0\mathbf{B}$, remains a convex combination of the endmembers as required. Assuming (k-s) of the components of $\boldsymbol{l}^0$ are non-zero, then $\mathbf{x}^0$ is the position vector of a point X^0 in that bounding convex polytope whose (k-s) vertices are not extreme for the dataset. There is an error vector $\mathbf{f}$ between the 'best' approximation X' and the new point X^0 which results from this intervention. Thus, it is possible to employ $\mathbf{f}$ to adjust the non-extreme vertices outwards, and by that means, to move the s bounding hyperplanes outwards.

The correction described above is equivalent to a redefinition of $\boldsymbol{l}$ in order to obtain feasible loadings, namely:

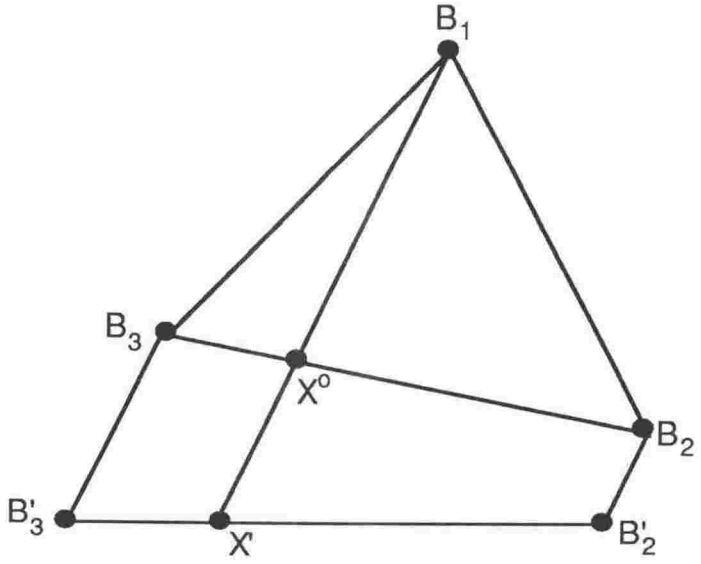
If the components $l_\alpha, l_\beta, ..., l_\delta$ of $\boldsymbol{l}$ are less than zero then,

$$\text{set} \qquad l_\alpha = l_\beta = \cdots = l_\delta = 0,$$

$$\text{followed by,} \qquad l^0_j = l_j \,/\, \sum_{m=1}^{k} l_m, \quad j = 1,2,...,k \qquad (3.30)$$

This creates the loading vector $\boldsymbol{l}^0$. Since $l_\alpha = l_\beta = ... = l_\delta = 0$ by definition, then by the second expression in (3.30), $l^0{}_\alpha = l^0{}_\beta = ... = l^0{}_\delta = 0$ and the complete set of components of $\boldsymbol{l}^0$ are the coefficients of a convex combination. In the case that s = 0, that is, when none of the components of $\boldsymbol{l}$ are negative, it is consistent to define $\boldsymbol{l}^0 = \boldsymbol{l}$ and $\mathbf{x}^0 = \mathbf{x}'$. Hence $\boldsymbol{l}^0$ and $\mathbf{x}^0$ are defined whether or not $\mathbf{x}'$ is external to polytope $B_1B_2...B_k$.

Figure 3.3. Adjustment of Two Endmembers in a Three Endmember Representation. $\mathbf{x}' = l_1\mathbf{b}_1 + l_2\mathbf{b}_2 + l_3\mathbf{b}_3$ where $l_1 < 0$. $X'B_1$ is intersected internally (at X^0) by side B_2B_3 of plane triangle $B_1B_2B_3$. Moving B_2 to B'_2 and B_3 to B'_3 is an outward displacement of the side B_2B_3. X' belongs to the convex hull of $B_1B'_2B'_3$ as required, but this not always the case for all points in $B_1B_2B_3$.



(It should be evident that a computer algorithm that had constructed $\boldsymbol{l}^*$ by equation (3.23) for each composition vector $\mathbf{x}$ belonging to a dataset $\mathbf{X}$ could then implement correction (3.30) automatically for each $\boldsymbol{l}$).

The point X^0, whose position vector is $\mathbf{x}^0 = \boldsymbol{l}^0\mathbf{B}$, lies on the hyperplane through the (k-s) points B_a, B_b, ..., B_d which are the points B_1, B_2, ..., B_k excluding B_α, B_β, ..., B_δ. The vector, $\mathbf{f} = (\mathbf{x}' - \mathbf{x}^0) = (\boldsymbol{l} - \boldsymbol{l}^0)\mathbf{B}$, lies in hyperplane P in a direction out of polytope $B_1B_2...B_k$, and is the error vector created by correction (3.30). If vector $\mathbf{f}$ were added to each of $\mathbf{b}_a$, $\mathbf{b}_b$, ..., $\mathbf{b}_d$, then recalling from the last paragraph that s of the components of $\boldsymbol{l}^0$ are zero while the remaining (k-s) components sum to one,

$$\sum_{h=1}^{k} l_h^0(\mathbf{b}_h + \mathbf{f}) = \sum_{h=1}^{k} l_h^0\mathbf{b}_h + (\mathbf{x}' - \mathbf{x}^0) \sum_{h=1}^{k} l_h^0$$

$$= \mathbf{x}^0 + (\mathbf{x}' - \mathbf{x}^0)$$

$$= \mathbf{x}'$$

Hence, X' lies on the hyperplane through the points whose position vectors are given by $\mathbf{b}_h + \mathbf{f}$, h = a,b,...,d. These points could serve as new vertices to replace B_a, B_b, ..., B_d. Since X' is external to the original polytope, this adjustment moves only the vertices that are not extreme just far enough to place X' on the new boundary. It is equivalent to defining an adjustment $\Delta\mathbf{B}$ (k×p) to matrix $\mathbf{B}$ as a linear function of the error vector $\mathbf{f}$ (1×p). That is, the new or adjusted matrix of endmembers $\mathbf{B}'$ (k×p) is given by

$$\mathbf{B}' = \mathbf{B} + \Delta\mathbf{B} \tag{3.31}$$

where

$$\Delta\mathbf{B} = \mathbf{G}\mathbf{f} \tag{3.32}$$

In the case described above,

$$g_h = \begin{cases} 1 \text{ for } h = a, b, ...,d \\ 0 \text{ otherwise} \end{cases}$$

In the single sample case, matrix **G** is a (k×1) column vector but, for a number n of samples, **G** would be k×n, and **f** would be replaced by n×p matrix **F** of error row vectors.

The principal shortcoming of the simple adjustment in which **f** is added to each of $\mathbf{b}_a$, $\mathbf{b}_b$, ..., $\mathbf{b}_d$, is that it moves all non-extreme vertices by the same displacement. Apart from the possibility that points representing observations which were internal to polytope $B_1B_2...B_k$ may be external to the new polytope, such an adjustment does not necessarily satisfy the geochemical requirement that extreme points should be as close to the convex hull of the data points as possible. That requirement would suggest that vertices that were remote from the external point X' should be moved the least, as in Figure 3.3. This criticism also applies to the method proposed by Full, Ehrlich and Klovan (1981) who stated that moving the edges of the polytope outwards, parallel to the 'original edges', was a strategy designed to keep the terminal hypervolume 'defined by the data' to a minimum.

An alternative expression for g_h in equation (3.32) which accomplishes a displacement directly proportional to loading is given by,

$$g_h = l_h^0 / |\boldsymbol{l}^0|^2 \tag{3.33}$$

Hence, combining equations (3.31) and (3.32),

$$\mathbf{b}'_h = \mathbf{b}_h + (l_h^0 / |\boldsymbol{l}^0|^2)(\mathbf{x}' - \mathbf{x}^0) \tag{3.34}$$

and

$$\sum_{h=1}^{k} l^0_h \mathbf{b}'_h = \sum_{h=1}^{k} l^0_h \mathbf{b}_h + \sum_{h=1}^{k} (l^0_h)^2 / |l^0|^2 (\mathbf{x}' - \mathbf{x}^0)$$

$$= \mathbf{x}^0 + (\mathbf{x}' - \mathbf{x}^0)$$

$$= \mathbf{x}' \qquad (3.35)$$

Therefore X' lies on the boundary hyperplane through $B'_a, B'_b, \ldots, B'_d$ as before.

Note that in the special case that all l^0_h are zero except l^0_m, then $l^0_m = 1$ and by equation (3.33), $g_h = 0$ all $h \neq m$, and $g_m = 1$. Further, $\mathbf{x}^0 = \mathbf{l}^0\mathbf{B} = \mathbf{b}_m$ so that by equation (3.34) $\mathbf{b}'_h = \mathbf{b}_h$ all $h \neq m$, and $\mathbf{b}'_m = \mathbf{x}'$, which is reasonable.

The possibility arises that some of the components of the new set of endmembers $\mathbf{b}'_a, \mathbf{b}'_b, \ldots, \mathbf{b}'_d$ are negative. A vector with one or more negative components is not in the positive orthant of p-space and a least squares projection onto estimate space S employing the orthogonal projection operator as in equation (3.29) with a non-negativity constraint, provides a feasible best solution.

Another possibility is that some members of the original dataset now have negative loadings on some of the $\mathbf{b}_\alpha, \mathbf{b}_\beta, \ldots, \mathbf{b}_\delta$, $\mathbf{b}'_a, \mathbf{b}'_b, \ldots, \mathbf{b}'_d$. If this is the case, then the partitioning procedure and endmember adjustment outlined above form the basis of an iterative algorithm for repeatedly adjusting the positions of successive sets of k trial endmembers until they are extreme (see also Section 3.4.3 below).

An obvious property of matrix adjustments which are linear combinations of the errors, like equations (3.32) and (3.34), is that if all the points are interior to the polytope H, then $\mathbf{f} = \mathbf{0}$ (or $\mathbf{F} = \mathbf{0}$) and so $\Delta\mathbf{B} = \mathbf{0}$. Hence an automated algorithm which adjusted endmembers by the incremental matrix of equation (3.32) (or more generally, by $\Delta\mathbf{B} = \mathbf{GF}$), could not move from a set of proper extreme points. These extreme points could be the initial vertices, or they could have been constructed as the

outcome of a sequence of such adjustments.

Another important property of these adjustments is that a new set of endmembers must always belong to S. By equations (3.30), (3.31) and (3.32), the adjusted matrix is,

$$\begin{aligned} \mathbf{B}' &= \mathbf{B} + \Delta\mathbf{B} \\ &= \mathbf{B} + \mathbf{G}\mathbf{f} \\ &= \mathbf{B} + \mathbf{G}(\mathbf{x}' - \mathbf{x}^0) \\ &= (\mathbf{I}_k + \mathbf{G}(\boldsymbol{l} - \boldsymbol{l}^0))\,\mathbf{B} \end{aligned} \tag{3.36}$$

From the last line above, it is clear that each of the new endmembers (rows) of (k×p) $\mathbf{B}'$ is a linear combination of the rows of $\mathbf{B}$ and therefore a vector belonging to estimate space S.

Provided each of the rows of $\mathbf{B}$ is a composition, then the rows of $\mathbf{B}'$ will also be compositions. That is, the adjustments all take place in hyperplane P, the subset of estimate space S to which all compositions belong. This observation is readily apparent for the 3 endmember configuration of Figure 3.3. By way of proof, it need only be established that the row sums of the matrix $(\mathbf{I}_k + \mathbf{G}(\boldsymbol{l} - \boldsymbol{l}^0))$ all total 1 (see equation (3.16)) . Since the i-th row-sum of this matrix is

$$1 + g_i \sum_{j=1}^{k} (l_j - l_j^0)$$

the result follows at once. Thus if the rows of $\mathbf{B}$ are compositions, then so are the rows of $\mathbf{B}'$. And since it has already been established that the latter must belong to estimate space S, they must then define points on hyperplane P.

3.4 GEOCHEMICAL DATASETS

So far, the development of algorithms for the determination of mixture coefficients and the adjustment of endmembers has been restricted to the special case of the single geological sample with a given set of endmember estimates. In this section, the generalized problem of constructing a convex representation for a number of samples will be examined. It will be assumed that only the matrix of observed compositional data **X** (n×p) is given, and that it is required to resolve **X** into the form (3.5) in the absence of prior knowledge of matrices **L** and **B**. That is, all the information needed to determine these two matrices is contained only in **X**.

3.4.1 The Estimate Space

It has already been noted that if the observed data resulted from some unknown mixing process with small random errors, then there should be a subspace S whose dimension is $k < p$, such that the rows of **X** (n×p) are approximately linear combinations of any k basis vectors of S. When that is the case, the approximate rank of the matrix **X** is k which is an estimate of the number of true endmembers. Thus, the first step in solving equation (3.5) for **L** and **B** is to identify S.

Ideally, this is accomplished by locating an orthogonal reference system in p-space for which the ordinates of the observed datapoints on some axes are large, and on the remaining axes are negligible. Then S is the subspace that is spanned by the unit vectors which define the first set of axes. This is because the object (data) vectors will approximately be linear combinations (the large ordinates) of those unit vectors. In practice, such outcomes as 'large' and 'negligible' ordinates on distinct sets of axes are not usual. What is common nevertheless is a rapid diminishing of the magnitudes of the spread around the origin O in the directions of certain eigenvectors taken in turn.

If $\mathbf{v}$ (p×1) is any unit column vector, then the components of the (n×1) vector $\mathbf{Xv}$ are the orthogonal projections of the n rows of $\mathbf{X}$ onto $\mathbf{v}$. Thus the scalar $\mathbf{v}^T\mathbf{X}^T\mathbf{Xv}$ is the sum of the squares of those projections (*cf.* equation (1.40)). The critical (turning) values of this sum of squares are equal to the p eigenvalues $\psi_1 \geq \psi_2 \geq ... \geq \psi_p \geq 0$ of the symmetric matrix $\mathbf{X}^T\mathbf{X}$, and occur when $\mathbf{v}$ is in the directions of the corresponding eigenvectors $\mathbf{v}_1,\mathbf{v}_2,...,\mathbf{v}_p$, respectively (see sections 1.4 and 1.5).

If $\psi_m = 0$, then the sum of squares $\mathbf{v}_m{}^T\mathbf{X}^T\mathbf{X}\mathbf{v}_m = 0$ so that the orthogonal projection of each row of $\mathbf{X}$ onto $\mathbf{v}_m$ is zero. It follows that for all j, $m \leq j \leq p$, $\psi_j = 0$ and $\mathbf{v}_j$ is orthogonal to every row vector of $\mathbf{X}$. If the eigenvectors were taken as an alternative orthogonal reference system, then the coordinates of every datapoint of $\mathbf{X}$, as measured on the m-th to p-th axes, would be zero demonstrating that the data occupied a space of at most (m-1) dimensions. Further, each row of $\mathbf{X}$ would then be an exact linear combination of the first (m-1) eigenvectors so that the rank of $\mathbf{X}$ must at most be (m-1).

When ψ_m is not zero but nonetheless is very small, then all the results of the preceding paragraph become approximations.

Symmetric matrices $\mathbf{XX}^T$ (n×n) and $\mathbf{X}^T\mathbf{X}$ (p×p) have the same non-zero eigenvalues $\psi_1 \geq \psi_2 \geq ... \geq \psi_p \geq 0$ (see the derivation of equations (1.41) and (2.7)). These are associated with both the n×p matrix $\mathbf{U} = [\mathbf{u}_1,\mathbf{u}_2,...,\mathbf{u}_p]$ of unitized eigenvectors of $\mathbf{XX}^T$ and the p×p matrix $\mathbf{V} = [\mathbf{v}_1,\mathbf{v}_2,...,\mathbf{v}_p]$ of unitized eigenvectors of $\mathbf{X}^T\mathbf{X}$. If p×p $\mathbf{\Psi}^{1/2} = \text{diag}(\sqrt{\psi}_1,\sqrt{\psi}_2,...,\sqrt{\psi}_p)$, then the singular value decomposition (see sections 1.5 and 2.2.1) for $\mathbf{X}$ is given by

$$\mathbf{X} = \mathbf{U}\,\mathbf{\Psi}^{1/2}\mathbf{V}^T \tag{3.37}$$

This result is an immediate consequence of the readily verifiable relation $\mathbf{Xv}_j = \sqrt{\psi}_j\mathbf{u}_j$ as in the derivation of equation (2.8). Letting j = 1,2,...,p, it follows at once that

$\mathbf{XV} = \mathbf{U}\Psi^{1/2}$, and equation (3.37) is obtained by postmultiplying both sides of this by $\mathbf{V}^T = \mathbf{V}^{-1}$.

The sum of the eigenvalues is trace(Ψ) which also equals the trace of $\mathbf{XX}^T$ and $\mathbf{X}^T\mathbf{X}$. Hence the the total sum of squares for the data is,

$$\sum_{j=1}^{p} \psi_j = \sum_{i=1}^{n} \sum_{j=1}^{p} x_{ij}^2 = \sum_{i=1}^{n} OX_i^2 \qquad (3.38)$$

which will be invariant for all orthogonal transformations in p-space.

If the rows of $\mathbf{X}$ have been transformed into unit vectors so that $\mathbf{XX}^T$ is a similarity matrix, then the right hand side of (3.38) is equal to n. Such a transformation is the basis of Q-mode 'factor' and cluster analysis. With or without the transformation, an assessment of the approximate dimensionality of the data rests on the magnitude of the quotient

$$\sum_{j=1}^{k} \psi_j \Big/ \sum_{j=1}^{p} \psi_j \qquad (3.39)$$

for $k < p$. If, as is often the case for k much less than p, it happens that quotient (3.39) is large (for example 0.99), then the sum of the squares of the orthogonal projections of the rows of $\mathbf{X}$ on the eigenvectors $\mathbf{v}_{k+1}, \mathbf{v}_{k+2}, \ldots, \mathbf{v}_p$ is a negligible proportion of the total (equation (3.38)). The approximate rank of $\mathbf{X}$ is k and the consequent existence of a linear model (3.1) to account for the data seems to follow.

Equation (3.37) can be expanded as the matrix sum (3.40) below

$$\mathbf{X} = [\sqrt{\psi_1}\mathbf{u}_1, \ldots, \sqrt{\psi_k}\mathbf{u}_k] \begin{bmatrix} \mathbf{v}_1^T \\ \cdot \\ \cdot \\ \mathbf{v}_k^T \end{bmatrix} + [\sqrt{\psi_{k+1}}\mathbf{u}_{k+1}, \ldots, \sqrt{\psi_p}\mathbf{u}_p] \begin{bmatrix} \mathbf{v}_{k+1}^T \\ \cdot \\ \cdot \\ \mathbf{v}_p^T \end{bmatrix} \qquad (3.40)$$

Geometrically, the $\mathbf{v}_1,\mathbf{v}_2,...,\mathbf{v}_p$ are orthogonal unit vectors in p-space representing an alternative reference system. The coordinates of point X_i in this system are (from the two matrix addends of equation (3.40))

$$(\sqrt{\psi}_1 u_{i1}, \sqrt{\psi}_2 u_{i2}, ..., \sqrt{\psi}_k u_{ik}, \sqrt{\psi}_{k+1} u_{ik+1}, ..., \sqrt{\psi}_p u_{ip}) \qquad (3.41)$$

Now the u_{ij} are components of n-dimensional unit vectors and k exists such that for $j > k$ the ψ_j are negligible. Thus $\sqrt{\psi_j}\, u_{ij}$ is approximately zero for $j > k$ in (3.40) and (3.41) so that the rows of $\mathbf{X}$ occupy the k-dimensional space S defined by the $\mathbf{v}_1,\mathbf{v}_2,...,\mathbf{v}_k$ axes system within the errors determined by the $\sqrt{\psi_j}\, u_{ij}$, $j > k$.

The two matrices in the sum on the right of equation (3.40) can be associated with the terms in equations (3.1) and (3.5). The first matrix is $\mathbf{X}^*$ and the second is $\mathbf{E}^*$. Rewriting equation (3.40),

$$\mathbf{X} = \mathbf{X}^* + \mathbf{E}^* = \mathbf{X}' + \mathbf{E} \qquad (3.42)$$

where $\mathbf{X}'$ is the result of the premultiplication of $\mathbf{X}^*$ by an (n×n) diagonal matrix which rescales its rows into compositions. Assuming the rows of $\mathbf{X}$ sum to one and the rows of $\mathbf{X}'$ sum to one, then the rows of $\mathbf{E}$ sum to zero to maintain the matrix equation.

The vectors $\mathbf{v}_1,\mathbf{v}_2,...,\mathbf{v}_k$ form a basis for the estimate space S in the positive orthant of p-space. From equations (3.40) and (3.42), it follows that the rows of $\mathbf{X}'$, being scalar multiples of the corresponding rows of $\mathbf{X}^*$, must belong to S since each is a linear combination of $\mathbf{v}_1,\mathbf{v}_2,...,\mathbf{v}_k$. It is within S that the estimated endmembers $\mathbf{b}_1,\mathbf{b}_2,...,\mathbf{b}_k$ will be sought. So for the purpose of assessing the validity of a nascent convex representation, the estimates for $\mathbf{X}_0 = \boldsymbol{\Lambda\beta}$ and $\boldsymbol{\varepsilon}$ already exist as the matrix of estimated mixtures (n×p) $\mathbf{X}'$ and the residuals (n×p) $\mathbf{E} = \mathbf{X} - \mathbf{X}'$. Similar observations to these were made in Section 2.2.1 (see the derivation of equation (2.13)).

Being orthogonal, $\mathbf{v}_1,\mathbf{v}_2,...,\mathbf{v}_k$ tend to lie outside the positive orthant of p-space ($x_{ij} \geq 0$) and would not determine the directions of feasible solutions for $\mathbf{b}_1,\mathbf{b}_2,...,\mathbf{b}_k$. Various writers have recommended varimax and oblique rotations of the set $\mathbf{v}_1,\mathbf{v}_2,...,\mathbf{v}_k$ in the context of Q-mode 'factor' analysis (see the summaries in Section 2.2.1 of the papers by Imbrie (1963), Imbrie and Van Andel (1964), Klovan (1966), Klovan and Imbrie (1971), Jöreskog, Klovan and Reyment (1976), Miesch (1976a,b), Clark (1978), Full, Ehrlich and Klovan (1981), Full, Ehrlich and Bezdek (1982), Leinen and Pisias (1984)). In that context, such vectors, which are not possible endmembers in general, were chosen because of the availability of orthogonal rotation algorithms, in particular, the varimax method (Kaiser (1958)). These methods had been developed to construct objectively a 'simple structure' from the loading matrix that had been derived from an R-mode factor analysis. Such rotations are not constrained by non-negativity conditions on all matrix elements. Indeed, such a constraint is impossible on the components of the factor vectors. Nor are the factors they create compelled towards the position vectors of extreme or nearly extreme points. (Somewhat informally, the convex hull of a set of 'nearly extreme' points encloses most of the datapoints in dimension $q \leq p$).

The ideal outcome for an R-mode factor analysis is that in which the mean-corrected variable-vectors define disjoint, orthogonally located clusters of points (whether or not a factor model exists). The ideal outcome for a mixture analysis is that in which object vectors define uniformly distributed points within a convex polytope (if a mixing model exists). The varimax criterion is designed for and quite efficient at detecting the former configuration. There are no theoretical grounds to expect it to work in the latter.

The singular value decomposition creates (n×p) $\mathbf{X}^*$ the least squares approximation to (n×p) $\mathbf{X}$ in S. This is geometrically obvious because in the reference system defined by the eigenvectors $\mathbf{v}_1,\mathbf{v}_2,...,\mathbf{v}_p$, the coordinates

$$(\sqrt{\psi}_1\ u_{i1}, \sqrt{\psi}_2\ u_{i2}, ..., \sqrt{\psi}_k\ u_{ik}, 0, ..., 0) \qquad (3.43)$$

of the point X^*_i place it at the foot of the orthogonal projection of X_i onto the coordinate hyperplane S spanned by $\mathbf{v}_1,\mathbf{v}_2,...,\mathbf{v}_k$ (compare lines (3.41) and (3.43)) . Therefore, the rows of $\mathbf{X}^*$ are the orthogonal projections of the rows of $\mathbf{X}$ into S.

The angular error $X_iOX^* = X_iOX'_i$ can be examined for each i = 1,2,...,n. Large angular deviations identify both outliers among the samples and gross typographical errors in the dataset. Large in this context usually means more than four times the mean angular error

$$\frac{1}{n} \sum_{i=1}^{n} X_iOX'_i \qquad (3.44)$$

and is rare in a good linear representation. The quantity (3.44), together with the quotient (3.39), are two initial goodness of fit indicators for the estimated mixtures $\mathbf{X}'$ obtained from the singular value decomposition of $\mathbf{X}$.

Experience with quite modest datasets ($n \geq 60$) has shown that the singular value decomposition is robust in the sense that correcting or removing outliers has little effect on either the eigenvalues or eigenvectors. The space S is identified by all the information in all the samples. Its dimension k defines the approximate rank for (n×p) $\mathbf{X}$ and the estimate for κ, the true number of endmembers. The rows of (n×p) $\mathbf{X}'$ are the compositions formed by rescaling the orthogonal projections of the rows of $\mathbf{X}$ into S. Thus, $\mathbf{X}'$ is both the estimate of $\mathbf{X}_0 = \Lambda\beta$, and the 'best' approximation to $\mathbf{X}$ for a conjectured k-source mixing process given by equation (3.1). The next problem is to solve the equation (3.5) for (n×k) $\mathbf{L}$ and (k×p) $\mathbf{B}$, and the first step in the solution is to locate k extreme or nearly extreme points of $\mathbf{X}'$.

3.4.2 The Identification of Extreme Observations

Let the extreme points $B_1, B_2,..., B_k$ be defined by the set of endmembers $\mathbf{b}_1, \mathbf{b}_2,..., \mathbf{b}_k$. The lemma below establishes an elementary property of the coordinates x'_{ij}, $j = 1, 2,..., p$, of points X'_i lying inside the convex hull of $B_1, B_2,..., B_k$.

Provided $\mathbf{X}' = \mathbf{LB}$ *is an array of exact mixtures as in the convex representations* (3.1) *through* (3.5), *then for each* $i = 1,2,...,n$,

$$\textit{if} \quad b_{\beta j} \leq b_{aj} \leq b_{\alpha j}, \quad a = 1,2,...,k$$

$$\textit{then} \quad b_{\beta j} \leq x'_{ij} \leq b_{\alpha j}. \tag{3.45}$$

Proof:

Suppose a typical row of $\mathbf{X}'$ is given by $\mathbf{x}'$ so,

$$\mathbf{x}' = \boldsymbol{l}\mathbf{B} \quad \text{for some } \boldsymbol{l}$$

and,

$$x'_j = \sum_{i=1}^{k} l_i b_{ij}$$

Suppose also, $x'_j > b_{ij}$ for all $i = 1,2,...,k$.

Then since $l_i \geq 0$, $i = 1,2,...,k$ and $\sum_{i=1}^{k} l_i = 1$,

$$\sum_{i=1}^{k} l_i x'_j > \sum_{i=1}^{k} l_i b_{ij}$$

That is,

$$x'_j > x'_j$$

a contradiction.

A similar contradiction can be deduced if it is assumed that $x'_j < b_{ij}$ for each $i = 1,2,\ldots,k$.

Which completes the proof.

Hence, the endmembers contain the extreme values for each variable. This result is true in any reference system, and must also apply for example to the entries in the k columns of $[\sqrt{\psi}_1 \mathbf{u}_1, \sqrt{\psi}_2 \mathbf{u}_2, \ldots, \sqrt{\psi}_k \mathbf{u}_k]$, or even a varimax rotation on this matrix.

It is not necessarily the case that every endmember must contain extreme values for one or more of the variables. Consider

$$\mathbf{b}_1 = [1/2, 1/2, 0, 0], \quad \mathbf{b}_2 = [0, 0, 1/2, 1/2], \quad \mathbf{b}_3 = [1/8, 3/8, 1/8, 3/8]$$

These are 3 compositions. Now,

$$\beta_1\mathbf{b}_1 + \beta_2\mathbf{b}_2 = 1/2\,[\beta_1,\beta_1,\beta_2,\beta_2]$$

Clearly $\mathbf{b}_3$ can not be a linear combination of $\mathbf{b}_1$, $\mathbf{b}_2$ so that $\mathbf{b}_1,\mathbf{b}_2,\mathbf{b}_3$ are possible endmembers. None of the components of $\mathbf{b}_3$ is extreme in the current reference system. If, however, the axes are rotated in the direction of the eigenvectors, all three have extreme values in the rotated reference system.

If (3.45) does not hold for $b_{\beta j}$ or $b_{\alpha j}$ then $\mathbf{b}_\beta$ or $\mathbf{b}_\alpha$ is not an endmember. A sort on the magnitudes of the components in each of the p columns of (n×p) $\mathbf{X}'$ and in each of the k columns of (n×k) $[\sqrt{\psi}_1\mathbf{u}_1,\sqrt{\psi}_2\mathbf{u}_2, \ldots, \sqrt{\psi}_k\mathbf{u}_k]$, will reveal extreme samples.

In theory, k extreme samples which account for the maxima and minima for all estimated concentrations x'_{ij}, and the maxima and minima of the components $\sqrt{\psi}_j u_{ij}$ on

the eigenvectors $\mathbf{v}_1,\mathbf{v}_2,...,\mathbf{v}_k$, will serve as a set of k initial endmembers. In fact, k samples with all those properties not only do not necessarily exist, but such extreme samples as do exist are often outliers whose estimated mixtures are in S but remote from the body of the data. Accordingly nearly extreme samples are usually a more reliable choice.

Since the eigenvectors define the mutually orthogonal directions of the turning values of greatest spread about O, the neighbourhoods of the extremes of the components $\sqrt{\psi}_j u_{ij}$ on the eigenvectors $\mathbf{v}_1,\mathbf{v}_2,...,\mathbf{v}_k$ are more informative than those obtained from varimax rotated axes.

Note: Leinen (1987) stated that 'the experiment is therefore biassed by the choice of endmember compositions'. In fact, the process is a multivariate extension of the estimation of the 2 extremes of a bounded univariate distribution, and bias or not is then a consequence of the sampling procedure. Another kind of bias would be introduced if the extremes of raw data matrix **X** rather than **X'** were used as k initial endmembers as evidently undertaken by Dymond *et al.* (1984). These vectors do not as a rule span estimate space S which has been determined by all the samples. Consequently, estimated mixtures based on convex combinations of such vectors may be quite remote from S.

3.4.3 Adjustments to Endmembers

It is generally the case that extreme points are not contained in the rows of the dataset of estimated mixtures (n×p) **X'**. That is, if k initial (or trial) endmembers (k×p) $\mathbf{B}_1$ were chosen from the rows of **X'** and the exact solution for (n×k) $\mathbf{L}_1$ constructed for the matrix equation $\mathbf{X'} = \mathbf{LB}$ (equation (3.5)), then some of the elements of $\mathbf{L}_1$ would be negative. The solution to this problem should be to move outwards those initial

vertices that did not define bounding hyperplanes for the dataset. This would create new trial endmembers $\mathbf{B}_2$ and a new loading matrix $\mathbf{L}_2$. If negative loadings persisted, then further outward displacements of the current trial endmembers would be necessary. This procedure could be incorporated into an iterative algorithm which would be repeated in anticipation that, incrementally moving non-extreme trial vertices of the current polytope outwards would ultimately make all the data points of $\mathbf{X}'$ into interior points of the terminal polytope. This would be accomplished without the terminal vertices being more remote from the estimated mixture data points than was necessary. A description of iterative algorithms that possess some of these properties follows.

Assume that the rows of both (n×p) $\mathbf{X}'$, the estimated mixture matrix, and (k×p) $\mathbf{B}$, a set of trial endmembers (without subscript), are all compositions belonging to the estimate space S. Therefore, they are all position vectors of points on the hyperplane P which is a subset of S.

Each (1×k) loading vector $\boldsymbol{l}_i$ of (n×k) $\mathbf{L}$ is the exact solution of the equation (1×p) $\mathbf{x}'_i = \boldsymbol{l}_i\mathbf{B}$, and may be constructed by the operation (3.23), for i = 1,2,...,n. Hence, $\mathbf{X}' = \mathbf{LB}$.

Associated with each of the solutions for $\boldsymbol{l}_i$, there is the loading vector (1×k) $\boldsymbol{l}^0_i$ created by the correction (3.30), and the composition (1×p) $\mathbf{x}^0_i = \boldsymbol{l}^0_i\mathbf{B}$ which is also the position vector of a point in P. These n pairs of vectors are the rows of the matrices (n×k) $\mathbf{L}^0$ and (n×p) $\mathbf{X}^0$ respectively. Hence, $\mathbf{X}^0 = \mathbf{L}^0\mathbf{B}$. (Note that if $l_{ij} \geq 0$, j = 1,2,...,k then $\boldsymbol{l}^0_i = \boldsymbol{l}_i$, and $\mathbf{x}^0_i = \mathbf{x}_i$, which must be the position vector of an interior point of $B_1B_2...B_k$).

Let (n×p) $\mathbf{F} = \mathbf{X}' - \mathbf{X}^0$ be the matrix of error row vectors created by the n applications of correction (3.30). If $\mathbf{F} \neq \mathbf{0}$, then at least one point (row) of $\mathbf{X}'$ is external to H, the convex polytope $B_1B_2...B_k$. So at least one of the trial endmembers

is not extreme. Accordingly, the non-extreme vertices of $B_1B_2...B_k$ must be identified and moved outwards. Generalizing the method defined by equations (3.31) and (3.32) for the case of a single external point, let **G** be a (k×n) matrix of error vector coefficients. The new or adjusted matrix of endmembers (k×p) $\mathbf{B}' = \mathbf{B} + \Delta\mathbf{B}$ as before, where the incremental matrix adjustment (k×p) $\Delta\mathbf{B}$ is defined by,

$$\Delta\mathbf{B} = \mathbf{GF} \tag{3.46}$$

The h-th row of $\Delta\mathbf{B}$ is the linear form,

$$\Delta\mathbf{b}_h = \sum_{i=1}^{n} g_{hi}\,\mathbf{f}_i \tag{3.47}$$

All (1×p) error vectors $\mathbf{f}_i = (\mathbf{x}'_i - \mathbf{x}^0_i)$ lie in the hyperplane P, i = 1,2,...,n, as must each $\Delta\mathbf{b}_h$, h = 1,2,...,k. Consequently, the new endmembers must represent points belonging to hyperplane P and therefore to estimate space S. This can be demonstrated by following the same steps as for the derivation of equation (3.36). The adjusted matrix **B'** is,

$$\begin{aligned}\mathbf{B} + \Delta\mathbf{B} &= \mathbf{B} + \mathbf{GF} \\ &= (\mathbf{I}_k + \mathbf{G}(\mathbf{L} - \mathbf{L}^0))\mathbf{B}\end{aligned}$$

Therefore the rows of **B'** belong to the space spanned by the rows of **B**, which is S by assumption.

The i-th row-sum of the matrix $(\mathbf{I}_k + \mathbf{G}(\mathbf{L} - \mathbf{L}^0)$ is given by,

$$1 + \sum_{j=1}^{k}\sum_{\alpha=1}^{n} g_{i\alpha}\,(l_{\alpha j} - l^0_{\alpha j})$$

Reversing the order of the double sum in the line above reduces the term in parentheses to 1 - 1 = 0, and the entire expression to 1. Hence by equation (3.16), since all the rows of **B** are compositions, the i-th row of **B'** is also a composition, i=1,2,...k.

Therefore, the incremental matrix adjustments defined by equation (3.46), which are based on linear combinations of the errors $\mathbf{f}_i$, will move selected vertices to new positions on the hyperplane P in estimate space S. This satisfies a necessary condition for any solution to the equation $\mathbf{X}' = \mathbf{LB}$, namely that all vertices of the polytope $B_1B_2...B_k$ are points of the hyperplane P. Otherwise the required equality, $\mathbf{X}' = \mathbf{LB}$, where $\mathbf{X}'$ and $\mathbf{L}$ are as defined above, would be false.

If $\mathbf{F} = \mathbf{0}$, then the rows of $\mathbf{X}'$ represent points which are internal to $B_1B_2...B_k$. Further, $\Delta\mathbf{B} = \mathbf{0}$ by equation (3.46), and no displacements to any of the vertices could follow by implementing this method.

The last two paragraphs have established general properties of the adjustments (3.46). It remains now to specify the matrix $\mathbf{G}$ of error vector coefficients (see equations (3.46) and (3.47)). In fact, research into the the choice of $\mathbf{G}$ is not complete. The ultimate goal is an iterative procedure which would steadily diminish the errors at each cycle, and be guaranteed to converge to k extreme points (vertices) in the hyperplane P. Monitoring the errors is quite straightforward, and the single scalar given by trace($\mathbf{F}^T\mathbf{F}$)/np has proven adequate for tracking the approach toward total inclusion of all the estimated mixture data points within a polytope (see equation (3.50)). The construction of an algorithm that would be attracted towards k vertices from any k intial points within k respective neighbourhoods (of near extremes for example), has proven to be a good deal more difficult. Two *ad hoc* solutions which not only have proven successful on real data, but also could serve as starting points for more elaborate procedures, are described below.

The first employs a weighted mean error vector coefficient. In equation (3.47) define

$$g_{hi} = l^0{}_{ih}/n_h \tag{3.48}$$

where n_h is the number of vectors of the form $l^0_{ih}\mathbf{f}_i$, i = 1,2,...,n, with non-zero magnitudes, and the loading l^0_{ih} is the weight. This coefficient for the error vector $\mathbf{f}_i$ has similar properties to that defined by equation (3.33) for the error vector $\mathbf{f}$ associated with a single external point X'. The term $g_{hi}\mathbf{f}_i$ contributes a displacement to B_h which is directly proportional to l^0_{ih}, which in turn is a measure of the external displacement of X'_i from B_h. Since the sum of the components of $\boldsymbol{l}^0$ is 1, the denominator of the expression on the right of equation (3.33) is less than one. A more conservative adjustment then is to remove $|\boldsymbol{l}^0|^2$ from the denominator of the error vector coefficient and rely on the iterative procedure to compensate for the diminished displacements. Obviously when there are a number of points external to the polytope it would not be sensible to form the vector sum of all the displacements, consequently for each point B_h a mean displacement is constructed, hence the inclusion of n_h in the denominator. This is the number of vectors in the vector sum on the right of equation (3.47) whose magnitudes are non-zero. It is the inner product $\boldsymbol{\gamma\delta}$ of the two n-dimensional vectors $\boldsymbol{\gamma}$, $\boldsymbol{\delta}$ where $\gamma_i = 0$ if $l_{ih} = 0$, $\gamma_i = 1$ if $l_{ih} > 0$, $\delta_i = 0$ if $|\mathbf{f}_i|^2 = 0$, $\delta_i = 1$ if $|\mathbf{f}_i|^2 > 0$.

The second choice for the (k×n) matrix of error vector coefficients is,

$$\mathbf{G} = (\mathbf{L}^T\mathbf{L})^{-1}\,\mathbf{L}^T \qquad (3.49)$$

This form of $\mathbf{G}$ is that which minimises the sum of squared residuals formed by solving the overdetermined system $\mathbf{X}' \approx \mathbf{L}^0(\mathbf{B} + \Delta\mathbf{B})$ or equivalently, $\mathbf{F} \approx \mathbf{L}^0\Delta\mathbf{B}$ (see Rao (1973)). It is difficult to interpret adjustments to endmembers which are the regression coefficients for the orthogonal projections of each of the columns of (n×p) $\mathbf{F}$ into the space spanned by the columns of the estimated loading matrix (n×k) $\mathbf{L}^0$. Worse, experience has shown that the method can diverge sharply when used iteratively. Experience has also shown however, that it is very efficient for reducing the mean squared error (3.50) iteratively, and therefore can produce useful results if there is an

intervention when (3.50) attains a minimum.

Dymond *et al.* (1984) described a similar adjustment process except that they did not identify an estimate space S to which k endmembers must belong. Instead they specified k and solved the overdetermined system $\mathbf{X} \approx \mathbf{L}(\mathbf{B} + \Delta\mathbf{B})$ for $\Delta\mathbf{B}$. The left hand side of this system was the observed data **X**, initial extremes were chosen from **X**, and **L** was obtained by linear programming methods. Their results will be discussed in Chapter 4.

As in Section 3.3 these strategies lead to iterative procedures, the convergence of which can be monitored by computing a mean squared error

$$\frac{1}{np} \sum_{i=1}^{n} \sum_{j=1}^{p} (x'_{ij} - x^0_{ij})^2 \qquad (3.50)$$

which is trace($\mathbf{F}^T\mathbf{F}$)/np as remarked earlier. Thus the residual matrix **E** is a fixed property of the identification of k-dimensional space S. Mean squared error (3.50) is the additional penalty for stopping the iteration before all x^0_{ij} are equal to x'_{ij}. And that would imply that the convex hull of the current set of trial extreme points did not include all the X'_i. Such a situation arises if extreme points are pushed into the coordinate hyperplanes of the positive orthant of p-space without fully enclosing the data points X_i.

An illustration involving applications of the coefficients (3.48) and (3.49) is postponed until Section 3.4.5 where the question of the convergence of these procedures will be raised. The most exacting analysis requires that the data first be rescaled. In Section 3.4.4 which follows, there is a brief discussion of a column transformation that will be a standard procedure of the mixture analyses of this thesis.

3.4.4 Transformations

The expression (3.1) disguises a computational problem which should be evident from the interpretation placed on equation (3.40). That is, that the smallness of some of the eigenvalues may not be due to random departures from a low dimensional configuration of the datapoints, but to the presence of low scales of measurements on some of the variables. Many geochemical datasets combine observations on collections of major elements measured in percentages, and trace elements measured in parts per million. It is possible for the two classes of measurements to differ on a common scale by a factor of the order of 1 in 10,000. The apparent dimensionality of the complete dataset on such a common scale would reflect at most the number of major elements. Indeed the trace elements would determine eigenvectors that were very close to the axes on which they were measured.

A simple transformation, based on the observed data, which removes this difficulty is to divide each column of data matrix **X** by the maximum data value in that column (Imbrie and Van Andel (1964) and Miesch (1976b, 1980)). This rescales all element concentrations into the interval [0,1]. It also preserves the individual coefficients of variation. Equation (3.1) becomes, on post-multiplication by the column rescaling, nonsingular, (p×p) diagonal matrix **C**,

$$\mathbf{XC} = \mathbf{LBC} + \mathbf{EC}$$

or

$$\mathbf{X}^c = \mathbf{LB}^c + \mathbf{E}^c \tag{3.51}$$

So for example, where compositional constraint (3.2) defines a hyperplane in the positive orthant given by

$$\sum_{j=1}^{p} x_j = A \tag{3.52}$$

on which all points $X_1, X_2, ..., X_n, B_1, B_2, ..., B_k$ must lie, the post-multiplication by **C** sets up a correspondence with points $X^c_1, X^c_2, ..., X^c_n, B^c_1, B^c_2, ..., B^c_k$ on the hyperplane

$$\sum_{j=1}^{p} c_j x^c_j = A \tag{3.53}$$

If the error matrices **E** or $\mathbf{E}^c$ are zero (for an exact or contrived model), loading matrix **L** is unchanged by this transformation. In practice however, the singular value decomposition of matrix $\mathbf{X}^c$ produces different eigenvalues and eigenvectors as a result of the unit scale of measurement imposed on the p variables.

Both the partitioning and endmember-adjustment procedures described earlier take place in space S^c leading to the identification of the convex set $H^c \subset S^c$, and the determination of **L**. The inverse transformation $\mathbf{C}^{-1}$ creates the estimate space S, and the convex set H in which the relative positions of all points are preserved.

Post-multiplication of (n×p) **X** by the non-singular diagonal matrix (p×p) **C** is a special case of an elementary column operation. Throughout this work it will be used to improve the precision of the estimates for k, (k×p) **B** and (n×k) **L**, and it will always be referred to as the *column transformation.*

3.4.5 Illustration

A 10-dimensional array of compositional data of exact rank 3, originally due to Imbrie (1963, Table 9A), has been thoroughly worked through by others in the context of Q-mode factor analysis (see Section 2.2.1). These data are appropriately denoted by

(10×3) **X'**. They are are highly suitable for illustrating the methods for solving the matrix equation **X'** = **LB** (equation (3.5)) because the known contrived solution can be derived almost at once without the Q-mode rigmarole, and it is relatively easy to assess the different solutions constructed by the iterative methods described in Section 3.4.3. All computations described below take place following the column transformation of **X'** (see equation (3.51)). Indeed, the final procedure in any analysis is the inverse column transformation.

A singular value decomposition of the column transformed data was performed first. The relative magnitudes of the first three eigenvalues (equation (3.39) were 90.82%, 6.16% and 3.02%, which sum to 100%. The orthogonal projection of the data into the space spanned by the first 3 eigenvectors proved to be an identity transformation as expected. The orthogonal projections of the 10 samples on each of those eigenvectors were the coordinates of the data points in the reference system defined by the eigenvectors. Since extreme points must contain extreme values for the data (see Section 3.4.2), the samples were ranked from largest to least coordinate on each axis (eigenvector). The reordered sample numbers are set out on table 3.1. In each column of that table, the sample with the largest value is at the top, the sample with the least is at the bottom.

It is evident at once from Table 3.1 that samples 1, 2 and 3 have the highest and lowest coordinates on each axis (eigenvector) and therefore qualify as initial endmembers. This conclusion would be equally evident from a similar table constructed for each of the 10 variables. Using these sample as initial endmembers in the iterative algorithm, revealed at the outset that **F** = **0**, the 3 samples were true extremes, and that, together with the computed (10×3) loading matrix **L** (Table 3.2), confirmed the published results (see Imbrie (1963, Table 9B) without any iterations being performed.

The compositions of these 3 samples are displayed for purposes of comparison with later estimates on Table 3.3. They are the 'A' group of columns numbered 1, 2 and 3, in both the upper and lower tables respectively.

Although it did not feature in the analysis, a ternary diagram provides the simplest representation of the (untransformed) 10-dimensional data (see Figure 3.1)). The known loadings (Table 3.2) of all the samples on the first 3 samples, serve as coordinates of the points in a 3-space. But since such coordinates form the coefficients of convex combinations, the points that they represent all lie in the plane equilateral triangle of the diagram. The positions of the data points relative to each other are immediately apparent from this figure. Any three points within the positive orthant of 10-space whose convex hull enclosed the triangle of Figure 3.1, would constitute a feasible solution to the equation (10×3) $\mathbf{X}' = \mathbf{LB}$. Only one vertex of the triangle is in a coordinate hyperplane (Table 3.3, group 'A', column 2, variable 10 is zero), so it would appear that there are indefinitely many feasible solutions.

Table 3.1

Samples Ranked According to The magnitudes of Their Orthogonal Projections on Each Eigenvector For Data Due to Imbrie (1963)

Axis1	Axis2	Axis3
1	2	3
5	7	10
6	9	2
3	4	9
10	8	7
8	1	6
4	5	8
9	10	4
7	6	5
2	3	1

Table 3.2

Loadings on The First Three Samples, of The Ten Samples of Data Due to Imbrie (1963)

Sample	1	2	3
1	1.0	0.0	0.0
2	0.0	1.0	0.0
3	0.0	0.0	1.0
4	0.5	0.5	0.0
5	0.8	0.0	0.2
6	0.4	0.0	0.6
7	0.2	0.7	0.1
8	0.5	0.3	0.2
9	0.2	0.6	0.2
10	0.1	0.1	0.8

Suppose now that samples 1, 2 and 3 are ignored. From Table 3.1, another set of possible trial extremes are samples 5, 7 and 10, all three accounting quite well for the highs and lows on the 3 axes. Using Figure 3.1 *in lieu* of a nearest neighbours analysis, it is evident that these points are not only remote from each other, but are also the vertices of a triangle which is roughly similar to the that defined by samples 1, 2 and 3. It need hardly be pointed out that, because they belong to the data, samples 1, 2 and 3 constitute the 'best' solution to equation (3.5).

A solution was sought, initializing the iterative procedure with samples 5, 7 and 10, and employing the mean error vector coefficient defined by equation (3.48). The mean squared error (equation (3.50)) reduced monotonically from the beginning, reaching a local minimum of 2.8×10^{-7} at 10 iterations. Thereafter it slowly increased. The compositions of the estimated endmembers at the 10th iteration are set out on Table 3.3, in the upper table, under group 'B'.

Comparing the estimates with the corresponding compositions of samples 1, 2 and 3 (Table 3.3, group 'A'), the similarities are so striking that it must be asked, why did the algorithm not reach these 3 points or three external points in each of their respective neighbourhoods?

An examination of the uncorrected loadings revealed that neither the 2nd nor 3rd constructs were quite extreme. But the 2nd was already constrained by the coordinate hyperplane, Variable 10 = 0. A tentative answer then, is that the non-negativity constraint on the components of the estimates kept overriding the adjustment. Instead of moving outwards out of the positive orthant, the 2nd construct was forced to move in the coordinate hyperplane in a direction that kept the 4th point external (see Figure 3.1)).

Table 3.3

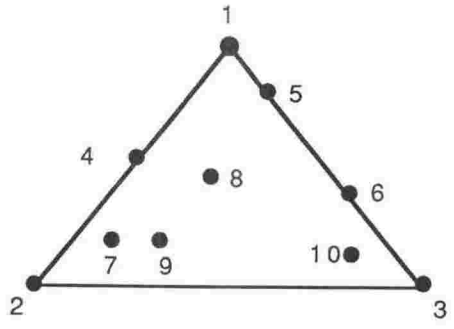
Endmember Compositions For Contrived Data

Originally Due to Imbrie (1963)

	A			B			C		
	1	2	3	1	2	3	1	2	3
1	5.00	10.00	3.00	5.00	10.05	3.00	5.48	10.00	1.25
2	25.00	30.00	6.00	25.00	30.21	6.01	28.22	30.00	0.00
3	15.00	17.00	10.00	15.00	17.06	10.00	15.87	17.00	8.25
4	5.00	17.00	13.00	4.99	17.01	13.00	4.10	17.00	12.00
5	5.00	8.00	25.00	5.00	7.84	24.99	1.87	8.00	29.25
6	20.00	8.00	15.00	20.01	7.96	15.00	20.42	8.00	16.75
7	10.00	5.00	13.00	10.01	4.96	13.00	9.36	5.00	15.00
8	5.00	4.00	8.00	5.00	3.97	8.00	4.48	4.00	9.00
9	5.00	1.00	5.00	5.00	0.97	5.00	4.87	1.00	6.00
10	5.00	0.00	2.00	5.01	0.00	2.00	5.32	0.00	2.50

	A			B			C		
	1	2	3	1	2	3	1	2	3
1	5.00	10.00	3.00	4.68	10.00	2.96	5.78	10.08	1.22
2	25.00	30.00	6.00	24.68	30.00	5.86	30.19	30.31	0.00
3	15.00	17.00	10.00	14.87	17.00	9.96	16.40	17.09	8.22
4	5.00	17.00	13.00	4.23	17.00	12.98	3.54	17.03	11.97
5	5.00	8.00	25.00	4.81	8.00	25.10	0.00	7.76	29.28
6	20.00	8.00	15.00	20.77	8.00	15.04	20.67	7.93	16.77
7	10.00	5.00	13.00	10.32	5.00	13.05	8.95	4.90	15.02
8	5.00	4.00	8.00	5.06	4.00	8.02	4.16	3.95	9.01
9	5.00	1.00	5.00	5.26	1.00	5.02	4.79	0.95	6.01
10	5.00	0.00	2.00	5.32	0.00	2.01	5.52	0.00	2.51

Figure 3.4. A ternary diagram of ten data points based on the contrived 10-dimensional compositional dataset of exact rank 3, originally due to Imbrie (1963). Since $\mathbf{x} = l_1\mathbf{b}_1 + l_2\mathbf{b}_2 + l_3\mathbf{b}_3$, the position of the point X in a 3-space may be defined by the coordinates (l_1, l_2, l_3). But $l_1 + l_2 + l_3 = 1$ and $l_1, l_2, l_3 \geq 0$, so the points lie in a plane equilateral triangle whose vertices are points 1, 2 and 3.



Samples 5, 7 and 10 were maintained as initial extremes but the procedure was re-executed with the matrix of error vector coefficients defined by equation (3.49). Again, the mean squared error (equation (3.50)) reduced monotonically, but this time it reached zero exactly ($\mathbf{F} = \mathbf{0}$), after 11 cycles. Hence, no further adjustments were possible. The estimated endmember compositions for this solution are set out on Table 3.3, in the lower table, under group 'B', and may be compared with those for samples 1, 2 and 3 under group 'A'. This solution is a set of true extremes. The second construct is identical to sample 2 but in the other two, the maximum (or minimum) for each variable, if it occurs, has been slightly increased (or decreased). So for example, Variable 1 takes the minimum value 3.00 in sample 3 and 2.96 in construct 3. Variable 5 takes the maximum value 25.00 in sample 3 and 25.10 in construct 3. Geochemically, this would have to be pronounced a satisfactory solution because the endmember estimates are true extremes which are proximate to particular samples in the dataset.

Returning to Table 3.1, the samples 4, 6 and 9 appear to be possible initial endmembers. A glance at Figure 3.1 suggests quite the opposite. Sample 4 is exactly

halfway between samples 1 and 2, on the side of the triangle. Nevertheless such graphical aids are not always available. The algorithm was executed with these trial extremes, first by employing the error vector coefficient defined by equation (3.48) and then by employing definition (3.49).

The solution to the first (definition (3.48)) was another set of true extremes, that is, the mean squared error (equation (3.50)) dropped to zero exactly, halting further adjustments. These extremes are set out on Table 3.3, in the upper table, under group 'C'. Once again construct 2 is identical to sample 2, variable maximum values on the other two constructs are driven up, and minima are driven down.

The second application (definition (3.49)), behaved in a fashion which had been observed before. The mean squared error fell monotonically, reaching a minimum (6.9×10^{-7}) at the 23rd cycle, then it appeared to diverge quite sharply. The estimated endmember compositions at that stage are set out on Table 3.3, in the lower table, under group 'C'. They are the most different from samples 1, 2 and 3 of the four sets of solutions, but not significantly so. That is, the patterns of variable associations within the constructs and the major and minor sources of each of the variables, is almost perfectly preserved.

For the four differing iterations involving the two types of error vector coefficients and two distinct sets of initial vertices, each error vector coefficient secured one exact solution by converging, and one estimated solution before diverging.

A monotonic increase in the mean squared error can be caused either by increasing numbers of points becoming external to the current polytope at each cycle or, and probably simultaneously, the magnitudes of individual error vectors becoming larger. This suggests that the trial vertices are not being moved outwards. With the matrix of error vector coefficients defined by equation (3.49), this may result from some

of the coefficients being negative, or of course, the non-negativity constraint preventing outwardly adjusted vertices from lying outside the positive orthant. The mean error vector coefficients defined by equation (3.48) are necessarily non-negative, and must result in an outward displacement of each vertex unless stopped by a coordinate hyperplane.

Clearly there are a number of modifications to the algorithm that could be examined. In order of increasing demands on machine time, three approaches are: (1) It would be desirable to find if there is a best direction to move when a vertex is placed in a coordinate hyperplane, as happened in this illustration. That is, a direction which would not only cause a continuation of the reduction in the mean squared error, but a greatest reduction. (2) The mean error vector coefficient (equation (3.48)) tends to absorb the magnitudes and directions of the major errors which could, perhaps, be picked off one at a time. (3) The only value that the mean squared error can converge to with the existing procedure, is zero. If the algorithm searched the neighbourhoods of the current trial extremes for those directions which optimized the reduction in the mean squared error, then convergence of this quantity to non-zero values would become a possibility. The estimates derived by such a process would be the best near extremes for the given initial set.

However, the chief virtue of coefficients (3.48) and (3.49), is that they are relatively readily computed. A satisfactory analysis hinges largely on the selection of the initial extremes, and that depends on the configuration of the data. If there had been a much larger number of samples in this illustration, then provided the data points were relatively uniformly distributed inside the triangle of Figure 3.1, samples 4, 6 and 9 would never have been selected, and possibly even better choices than samples 1, 2 and 3 would have been available. Alternatively, any number of points in a region bounded by a circle would contain no information on the positions of the true vertices. In the general case, the most satisfactory configuration for the data points of $(n \times p)$ $\mathbf{X}'$ is a

convex polytope which, it must be assumed, is similar to (and inside) the true polytope. Then, the possible divergence of procedures based on either definition (3.48) or (3.49) is not a serious problem in practice, provided the mean squared error (3.50) decreases monotonically from the initial set of vertices. The polytope which results in the minimum mean squared error should, like the first solution above, have been achieved by incremental steps into a nearly true extreme conformation.

3.5 STATISTICAL ALGORITHMS

The procedure for constructing a convex representation (3.5) is broken down into a series of tasks which are allocated to specific computer programs. By describing these programs in the order in which they are executed, it is intended to illustrate in this section how the relevant results of the preceding sections are linked together to form a step by step approach to a particular solution of the form (3.5).

All programs have been written in either FORTRAN 77 or SAS 5.16 (SAS Institute Inc. (1985)) and, at default input array sizes (800×40), will run under CMS on an IBM 4381 with 4Mb of core storeage. Larger arrays are presently limited only by a 16Mb maximum on core for this machine. All FORTRAN programs are set up to take task specifications interactively but, due to the large array sizes, are programmed to be sent automatically to the batch machine. The source code for SVD FORTRAN (in two parts) and LSQSEEK FORTRAN (also in two parts) which are described below, appear in the Appendix.

It should not need mentioning that scanning the raw data as well as producing basic summary statistics, before launching into a mixture analysis, can reveal assorted anomalies like missing values, typographical errors and so forth, which must be attended to. Sometimes a decision must be made whether or not to exclude from the analysis a

variable which appears to be almost dichotomous. For example, a trace element may take mostly zero values and perhaps one or two other values. The presence of such variables, which can not usually be modelled by a continuous mixing process, will simply degrade the overall analysis.

SETUP SAS

This program reads the raw data. It keeps the required variable list and drops samples which have missing values or otherwise belong to an exclusion list. If not already in the form of compositions, the retained variable list is usually transformed to sum to 100% by the formation of a subcomposition or partial composition for each sample. All measurements which are initially in ppm are divided by 10000 before this transformation is made. The hyperplane so-defined is a permanent reference space in the positive orthant for the all the subsequent algorithms. That is, all estimates or transformed datapoints are ultimately projected onto, or transformed back into this hyperplane. The final form of the required (n×p) data matrix is written to disk as raw data, a typical file-id would be CONSTSUM DATA, for input into the following programs.

SVD FORTRAN

The subroutines in this program include: SCALE which, according to directions from the console, divides each input variable by its maximum, forms fractional ranges or leaves the data unchanged; UNIT which only on request projects each datapoint onto the unit hypersphere (as for a Q-mode factor analysis), but otherwise leaves the data unchanged; CONSLQ will project any estimate with a negative component onto the nearest coordinate hyperplane. Algebraically, the algorithm is a constrained least squares. It is not necessary for the rowsums of the input raw data to be constant. But, if the rowsums are 100%, the main algorithm of this program restores this sum to the rows of the estimated mixture matrix.

The program reads CONSTSUM DATA, and following the execution of subroutines SCALE and UNIT, a partial singular value decomposition is performed on the transformed data. The number of dimensions (eigenvectors) sought initially is usually set at 10 or p (the number of variables), whichever is the smaller. For the first execution, the program will on request output only the maximum and minimum for each variable and the largest 10 (or p) eigenvalues.

If the number of endmembers (eigenvectors) k together with full output are specified, the program will write to disk the files: LOADINGS DATA consisting of the (n×k) components of the n samples on the k eigenvectors which span space S^c (not the matrix **L**); ESTIMATE DATA which is the (n×p) matrix **X'** in space S; EIGENVEC DATA which contains the first k eigenvectors in the space S^c; and finally SVD LISTING which contains all the test statistics such as the eigenvalues, their relative magnitudes and cumulative sums, the angles between each observed vector in S^c and its approximation in S^c, and the mean angular error.

It has been found that gross angular deviations for individual samples often arise from errors in the data. A possible cleanup of the data may take place at this stage resulting in re-executions of SETUP and SVD.

RSQUARE SAS

In an exact representation ($\mathbf{E} = \mathbf{0}$ in equation (3.1)), the n ordered pairs of observed and estimated values for each variable are the coordinates of n points on a straight line through the origin with slope 1. Clearly, a necessary condition for an exact solution is that the coefficients of determination (r^2) for all p sets of ordered pairs be equal to 1. The success of any solution depends on the reliable accounting for the values of each of the variables, otherwise the validity of the derived mixing process could be cast into doubt. So, although the test statistics produced by SVD may appear satisfactory for some value of k, it is necessary to check the p values for r^2 (as above)

before fixing k and hence the file ESTIMATE DATA. (It may also be necessary to examine the residuals, an option which is examined in Chapter 5). A table of the p values of r^2 for k = 2,3,...., may sometimes provide advance information on the structures of the endmembers, and indicate which variables are not accounted for by the estimated mixing process (see Chapter 5).

Program RSQUARE SAS reads files CONSTSUM DATA and ESTIMATE DATA, performs lineprinter plots if required and computes the p values for r^2.

LOADINGS SAS

When SVD has constructed both files ESTIMATE DATA and LOADINGS DATA for some specified k, LOADINGS SAS can read LOADINGS DATA and plot pairs of variables whose values are the coordinates of the orthogonal projections of the data into the k-dimensional subspace of S^c spanned by the first k eigenvectors. This program has been used to examine the locations with respect to the estimated data points, of derived endmembers which had been appended to ESTIMATE DATA. It also provides an immediate visual appraisal of the soundness of choosing the special values 2 or 3 for k. If the plotted points are collinear, then k = 2. If one set of plotted points is collinear and the remainder are triangular, then k = 3.

CORR SAS

Very high correlations between variables are evidence of the existence of invariant linear associations. Such associations in turn arise from the existence of endmembers which contain the extreme concentrations of these same variables. This program will read either CONSTSUM DATA or ESTIMATE DATA and produce the product moment correlation matrix for all the variables. Either correlation matrix may be useful for confirming later estimates of the endmembers.

EXTREMES FORTRAN

This program transforms the variables of any input raw data matrix **A** into fractional ranges. That is, the transformed element $a^c_{ij} = (a_{ij} - min_j)/(max_j - min_j)$. It reads bandwidth ζ from the console and then writes out all observation vectors for which any component lies in the intervals $[0,\zeta]$ and $[1-\zeta,1]$. It therefore provides a rapid dump of the extreme observations that it finds in **A**, and is usually executed twice, one run reading LOADINGS DATA, the other reading ESTIMATE DATA.

VARSORT SAS

An alternative method for identifying extreme observations is to sort on the magnitudes of each variable taken one at a time, and to list the m largest and m least observations. VARSORT SAS will read ESTIMATE DATA or LOADINGS DATA and sort each of the respective variable lists. A table can be prepared showing which sample numbers have the largest and least values on each of the variables. Ultimately, k extreme samples are to be chosen as initial endmembers.

NEIGHOBJ FORTRAN

A nearest and furthest neighbours table also identifies extreme samples. Outliers should have been detected by the table of angular deviations output by SVD, however any sample whose nearest neighbour is remote and which is also consistently furthest from most the others, would be a biassed choice for an initial extreme sample. Ideally, in a k-dimensional estimate space, k families of samples will be identified as k distinct groups of nearest and furthest neighbours. Executing this program to identify the initial extremes, one from each group, is the straightforward alternative to the algorithm proposed by Full, Ehrlich and Bezdek (1981). It is important of course, that near neighbours not be mistaken for distinct extremes. Another important application for the nearest neighbours table is for Q-mode clustering. This particular algorithm has been employed for this purpose by Glasby, Hunt and Renner (1985), Churchman, Hunt, Glasby, Renner and Griffiths (1988), Glasby, Stoffers, Walter, Davis and Renner (1988) and Kunzendorf, Gwozdz, Glasby, Stoffers and Renner (1988).

NEIGHOBJ FORTRAN includes the subroutines SCALE and UNIT described earlier. It reads ESTIMATE DATA and always executes UNIT. The datapoints, whether transformed by SCALE or not, are projected onto the surface of the unit hypersphere so that their proximities (similarities) to each other are given by the inner products of their position vectors. This program optionally writes out the similarity matrix to disk for processing by other clustering algorithms.

LSQSEEK FORTRAN

The principal objective of this program is to find, by the iterative reduction of the mean squared error (3.50), k extreme points $B_1,B_2,...,B_k$, such that every row vector of $\mathbf{X}'$ can be expressed as a convex combination of their position vectors $\mathbf{b}_1,\mathbf{b}_2,...,\mathbf{b}_k$, subject always to the non-negativity constraints. The program contains all the subroutines described above. It is initialized by reading the row numbers of the initial near extremes of $\mathbf{X}'$ from the console. At the end of any cycle, the current matrix $\mathbf{L}^0$ is that computed by orthogonal projections of the samples into S^c and corrected for negative loadings on the current extreme vectors in S^c. The current extreme vectors (after the initial set) are those which were adjusted to remove errors due to redefining negative loadings to zero in the previous $\mathbf{L}^0$. There are two methods available for endmember adjustment. Both employ the error vectors created by removing the negative loadings from $\mathbf{L}$. The first moves non-extreme points outwards from the convex polytope by computing the adjustments defined in equations (3.47) and (3.48), the second employs the least squares approach, definition (3.49), to fit new extreme points to the corrected matrix $\mathbf{L}^0$. The method selected is read from the console as is the maximium number of iteration cycles permitted. Output from the program includes the mean squared error at each cycle, the compositions of the current endmembers (at end of the final cycle), the loadings for each sample on the current endmembers, individual angular errors and the mean angular error for the column transformed data.

When none of its components is negative, no corrections are made to the matrix **L**, then there are no errors (**F** = **0**) and consequently no adjustments to the extreme points ($\Delta\mathbf{B} = \mathbf{0}$). So, from such a stage, the iterative algorithm would endlessly reproduce the same estimates **B** and **L** until stopped. This may happen with the initial choice of extreme points or at some later cycle. In most cases, it is more practical to intervene when the rate of reduction of the mean squared error slows to the point where excessive machine time contributes little improvement to it. It has been found then that the absolute values of the components of $\Delta\mathbf{B}$ are negligible so that the the compositions of the endmembers differ trivially from cycle to cycle.

To prevent underflow, errors f_{ij} of absolute magnitude less than 10^{-20} are redefined to zero. Thus, a squared error is greater than or equal to 10^{-40} or zero. (Underflow occurs on the IBM4381 at about 5×10^{-79}).

LSQMODEL FORTRAN

A problem sometimes arises where it is required to resolve one or more samples into a set of given endmembers. If the samples and the endmembers are concatenated into one file, LSQMODEL will read that file, read the row numbers of the endmembers from the console, project the samples orthogonally into the space spanned by the endmembers (transformed if necessary), scale the regression coefficients to sum to 1 for output, and compute the angles between observed and approximated samples.

Unlike LSQSEEK above, the program makes no corrections to the loadings, so it will be evident at once if one or more of the endmembers are not extreme. It will also be evident from the magnitude of the angle between them whether or not a sample is too remote from the least squares approximation to it, to be regarded as a mixture of the given endmembers. LSQMODEL can be used to check that the rank of a matrix like ESTIMATE data is exactly k. There should be no angular errors whatever rows are chosen as endmembers. It can also be used to monitor the final output from LSQSEEK.

If the constructed endmembers are appended to ESTIMATE DATA, the new file can be read by LSQMODEL and the loadings of the old file computed against the endmembers in the new. A judgement might then be made that the uncorrected loadings output from LSQMODEL were or were not sufficiently similar to those output by LSQSEEK.

CHAPTER 4

APPLICATIONS

SUMMARY

In this chapter, mixture analyses employing the techniques described in the last chapter are conducted on three compositional datasets. The first is a reanalysis of a small 'well-behaved' study of ferrmanganese nodules which confirms results that had already been published. The other two datasets have not previously been subjected to a mixture analysis.

An illustration is also provided in the form of a comment of the application of some of the procedures described in the last chapter to assess a putative set of endmembers.

4.1 FERROMANGANESE NODULES FROM *MANOP site H*

The raw data for this first application appeared in Dymond *et al.* (1984, Table 1). The paper itself was a report of a study which was part of the United States National Science Foundation supported Manganese Nodule Program (MANOP). Site H was a region of the eastern equatorial Pacific within 6°N to 7°'N, and 92°'W to 93°'W. Ferromanganese nodules and crusts from site H had been analysed for the p = 14 elements Na, Mg, Al, Si, K, Ca, Ti, Mn, Fe, Co, Ni, Cu, Zn and Ba in each of 16 nodule tops, 16 nodule bottoms, 17 whole nodules and 3 crusts, thus n = 52. Results obtained by Dymond *et al.* (1984) are included in this section in order to compare their linear programming based method with the proposed least squares approach.

4.1.1 A Linear Programming Based Analysis

Dymond *et al.* (1984) proposed three accretionary processes (and hence 3 endmembers) to account for the data. These were identified as: Hydrogenous precipitation, meaning the direct precipitation or accumulation of colloidal metal oxides from seawater; Oxic diagenesis, involving reactions in oxidized sediments that add transition metals to nodules; Suboxic diagenesis, where the reduction of manganese from the (IV) to the (II) valence in the sediments and the oxidation to the (IV) valence result in nodule accretion. They based their description of subsequent nodule chemical compositions upon the model that nodule compositions, both mineralogical and chemical, respond consistently to the seafloor environment.

Accordingly, they initialized an iterative search for a 3 endmember basis by assuming that 3 extreme samples in the dataset were close to pure endmembers. Their linear programming method in which a linear reformulation of sum (2.20) defines both the constraint equations and objective function as described in Section 2.3.1, was

employed to partition each of the composition vectors. From the notes included in an appendix, it would seem that endmember adjustments were determined by applying $\mathbf{G} = (\mathbf{L}^T\mathbf{L})^{-1}\mathbf{L}^T$ to the matrix $\mathbf{E}$ of residuals (see Dymond *et al.* (1984, Appendix 1)).

Endmember compositions obtained after 3 iteration cycles (Dymond *et al.* (1984, Table 6)) are reproduced here, in parentheses, in Table 1. The coefficients of determination (proportions of explained variance r^2) between the observed and their estimated values for each element (after Miesch, 1976b) are also reproduced in parentheses in Table 2 after Dymond *et al.* (1984, Table 6).

4.1.2 A Least Squares Based Analysis

A 'fill-up' value was constructed (equation (3.3)) for all samples, creating $\mathbf{X}$ (52×15) of partial compositions which was then column transformed into $\mathbf{X}^c$ according to equation (3.51). The singular value decomposition of $\mathbf{X}^c$ showed that the relative contributions of the first three (largest) eigenvalues to the sum of squares (equation (3.38)) were 94.39%, 2.76%, 2.03%, totalling 99.18% (see quotient (3.39)), the 4th largest contribution being 0.30%. A subspace of 3 dimensions was therefore identified as the transformed estimate space S^c, and the mean angular error (equation (3.44)) for angles between the rows of $\mathbf{X}^c$ and its approximation $\mathbf{X}'^c$ in S^c was 4.9° (mean similarity 0.9963).

Three extreme vectors belonging to $\mathbf{X}'^c$ were used to initialize an iterative search for $\mathbf{B}^c$ based on least squares methods for determining both $\mathbf{L}^0$ with correction (3.30), and $\Delta\mathbf{B}^c$ as in equations (3.46) and (3.49). However, the elements of the matrix of error vector coefficients $\mathbf{G}$ were defined by $g_{hi} = ((\mathbf{L}^T\mathbf{L})^{-1}\mathbf{L}^T)_{hi}$ only for $l_{ih} > 0$. Otherwise $g_{hi} = 0$ thus preventing an 'inward' component to the adjustment $\Delta\mathbf{b}^c_h$ due to the product of a negative loading with the i-th error $\mathbf{f}^c_i$. Iterations were stopped after 2

cycles when the mean squared error (3.50) had been reduced to 4.7×10^{-5}.

Endmember compositions $\mathbf{B} = \mathbf{B}^{c}\mathbf{C}^{-1}$ appear without parentheses in Table 4.1. Coefficients of determination (proportions of explained variance) between $\mathbf{X}$ and $\mathbf{X}' = (\mathbf{X}'^{c})\mathbf{C}^{-1}$ are set out, also without parentheses, in Table 4.2.

4.1.3 Comparisons

It is evident from Table 4.1 that corresponding pairs of endmembers constructed by algorithms which incorporated partitioning by least squares and linear programming respectively are not fundamentally geochemically distinct. The mean angular errors associated with each algorithm were, for untransformed data, both of the order of 1.1°. It is difficult to assess the relative positions of the 2 sets of endmembers since one of them is maintained in estimate space S and extreme for dataset $\mathbf{X}'$. Nevertheless, there are 10 extreme values of variables constructed by the linear programming approach which are not extreme for the same variables in either the raw data $\mathbf{X}$ or the estimated mixtures $\mathbf{X}'$.

Comparing the coefficients of determination between estimated and observed values of the elements in Table 4.2, it will be seen that the least squares based analysis created generally closer estimates than the linear programming method, most notably in the case of Mn. It created inferior estimates for Na, which with K and Zn were the least well accounted for by either analysis. However, the database was small and particularly tractable, so it is therefore reassurring that overall the results obtained by the two methods were very similar.

Plots of the least squares estimates of each of the variables against their observed values appear in Figure 4.1 (see for example, Renner (1982). See also

Dymond *et al.* (1984, Fig.8) for a comparison of these results with theirs). The plots permit a graphical assessment of the goodness of fit, and in a perfect representation, each set of points would lie on a straight line through the origin with slope one. A detail that the plots show quite clearly is a group of three points which appear as distinct outliers, located together but remote from the rest, for the plots of Na, Ca, Ti, Mn, Fe, Co, and Zn. Such configurations often inflate the coefficient of determination because of the apparent linearity between the centres of disjoint clusters. In each plot, these outlying points represent the three crusts P11-2, P11-4, P11-5, Dymond *et al.* (1984, Table 1)).

Accordingly, these crusts were removed from the database, and the remaining 49 nodule compositions were transformed as for equation (3.51). The singular value decomposition of the resulting 49×15 array revealed that a remarkable 99.05% of sum (3.38) was attributable to the first two eigenvalues. A least squares based analysis, orthogonally projecting the data into the 2-space spanned by the corresponding eigenvectors, determined two endmembers rather close respectively to the compositions of nodule top V48-1 and nodule bottom V52-1 (Dymond *et al.* (1984, Table 1)). The mean angular error for the two endmember representation of the transformed data was 5.49° (mean similarity 0.9954), and the mean squared error (3.50) after one iteration was 1.5×10^{-7}. The subsequent coefficients of determination were depressed further for Na, K and Zn (which were least well accounted for with 3 endmembers) but lay in the ranges 0.92 - 0.94 for Al, Si, Mn, Fe, Co and Cu, and the range 0.69 - 0.89 for Mg, Ca, Ti, Ni and Ba. In other words, the 49 nodule compositions were accounted for, almost as well with 2 endmembers, as the same data plus 3 crusts were with 3 endmembers.

These latter results suggest that a greater mathematical parsimony is possible in interpreting the data than was implied by the initial geochemical assumption of three accretionary processes. This suggestion would seem to be confirmed by the very low loadings associated with the Hydrogenous endmember in Dymond *et al.* (1984, Table 7)

for all but the 3 crusts.

It is to be inferred from the paper by Dymond *et al.* (1984) that the three proposed accretionary processes would account for different compositions measured on a nodule top, its bottom, and the whole nodule. An inference which was confirmed in part by the determination of an oxic endmember which was abundant among the tops and a suboxic endmember abundant among the bottoms. The whole nodules, on the other hand, were generally mixtures of these two. Statistically, it would have been an error to treat the nodule data as a compositional multivariate sample of order (49×15) as has been done here, without declaring (testable) assumptions concerning the independence of the observation vectors. There were in fact only 17 sampling units (nodules) present. One was too small for measurable top and bottom compositions so that two of these vectors were missing. It must be assumed that the 3 sets of composition vectors per nodule were related, though possibly perturbed from each other by the systematic processess described by Dymond *et al.* (1984). In any event, the effectively small nature of the database made for a straightforward mixture analysis by either process and hence to the very similar results.

Table 4.1

Endmember Compositions (%) Iteratively Adjusted to Fit Partitioning by Least Squares and by Linear Programming (in Parentheses) for MANOP data

Element	Hydrogenous		Oxic		Suboxic	
Na	1.75	(1.04)	2.53	(1.61)	4.04	(3.28)
Mg	1.12	(1.04)	2.34	(2.30)	1.36	(1.38)
Al	1.19	(1.18)	2.61	(2.71)	0.59	(0.75)
Si	5.14	(5.22)	5.73	(5.90)	1.25	(1.63)
K	0.51	(0.49)	0.84	(0.82)	0.60	(0.62)
Ca	2.55	(2.60)	1.55	(1.52)	1.20	(1.25)
Ti	0.51	(0.53)	0.17	(0.17)	0.0245	(0.0365)
Mn	20.60	(22.20)	32.28	(31.65)	46.86	(48.00)
Fe	18.23	(19.00)	4.92	(4.45)	0.10	(0.49)
Co	0.13	(0.13)	0.03	(0.028)	0.0012	(0.0035)
Ni	0.53	(0.55)	0.98	(1.01)	0.38	(0.44)
Cu	0.06	(0.05)	0.59	(0.62)	0.079	(0.115)
Zn	0.064	(0.075)	0.25	(0.25)	0.21	(0.22)
Ba	0.141	(0.148)	0.43	(0.44)	0.17	(0.20)

Table 4.2

Coefficients of Determination Between Estimated and Observed Elements Obtained from Partitioning by Least Squares and by Linear Programming (in Parentheses) for MANOP data

Element	Coefficient of Determination (% explained variance)	
Na	55.3	(64.0)
Mg	84.5	(84.1)
Al	95.0	(94.8)
Si	93.0	(91.6)
K	55.8	(54.7)
Ca	95.2	(91.1)
Ti	98.1	(97.7)
Mn	96.8	(86.2)
Fe	98.9	(99.5)
Co	99.1	(99.6)
Ni	80.3	(81.9)
Cu	97.6	(97.3)
Zn	47.7	(47.6)
Ba	76.9	(75.6)

Figure 4.1 Least squares estimates *vs.* observed compositions for MANOP data. The estimates were obtained by projecting the raw data orthogonally into the 3-dimensional estimate space, then rescaling to form compositions.

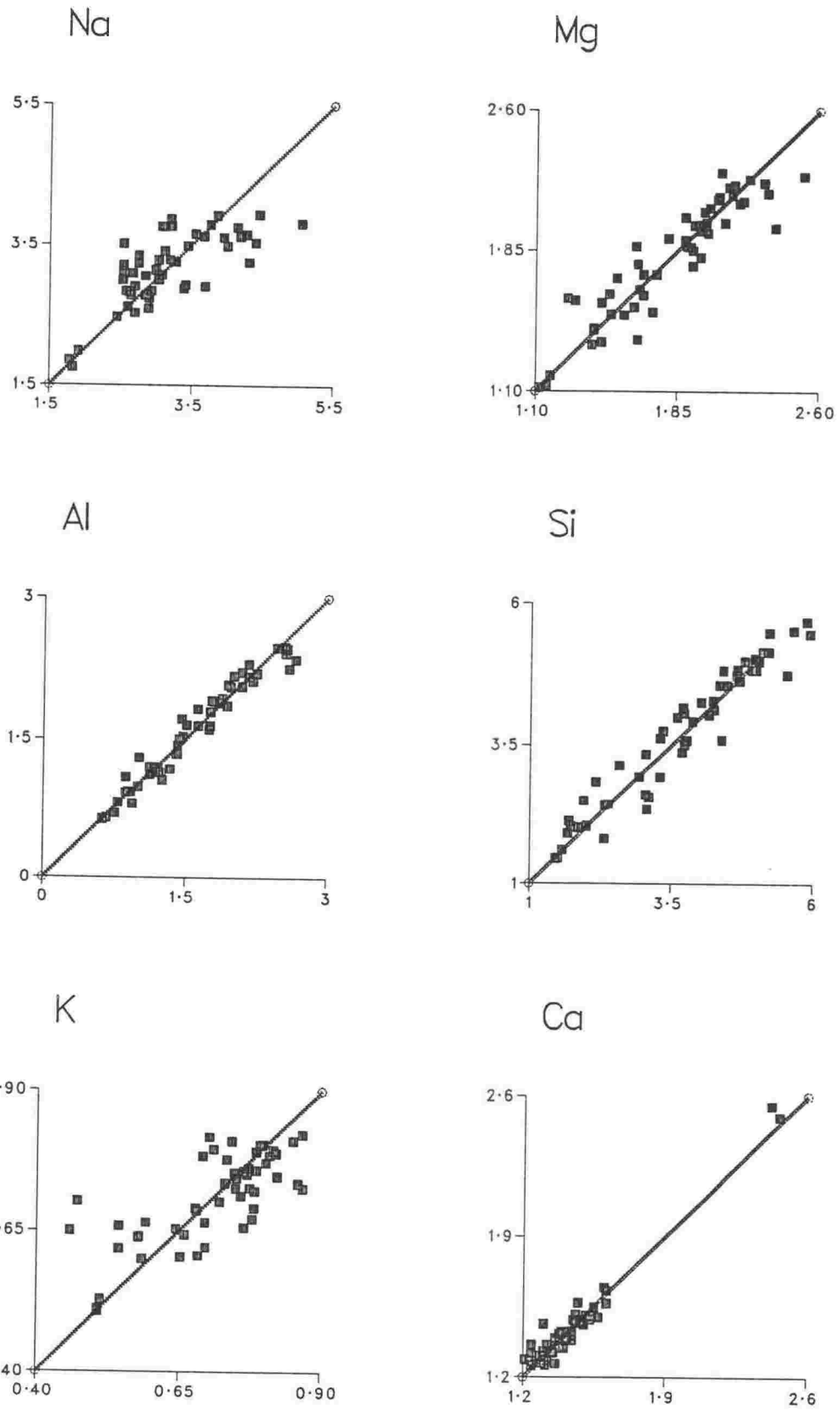


Figure 4.1. (continued)

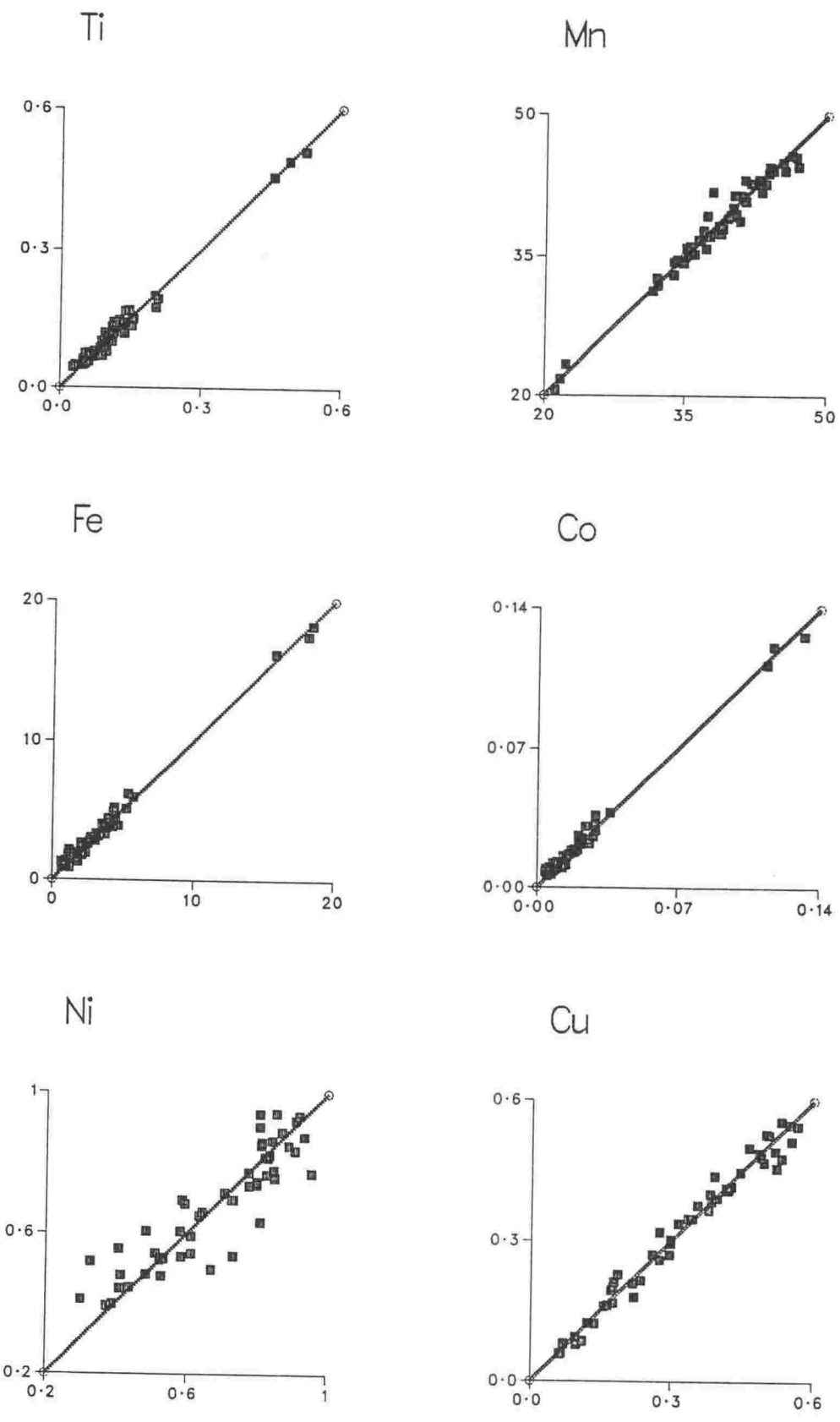
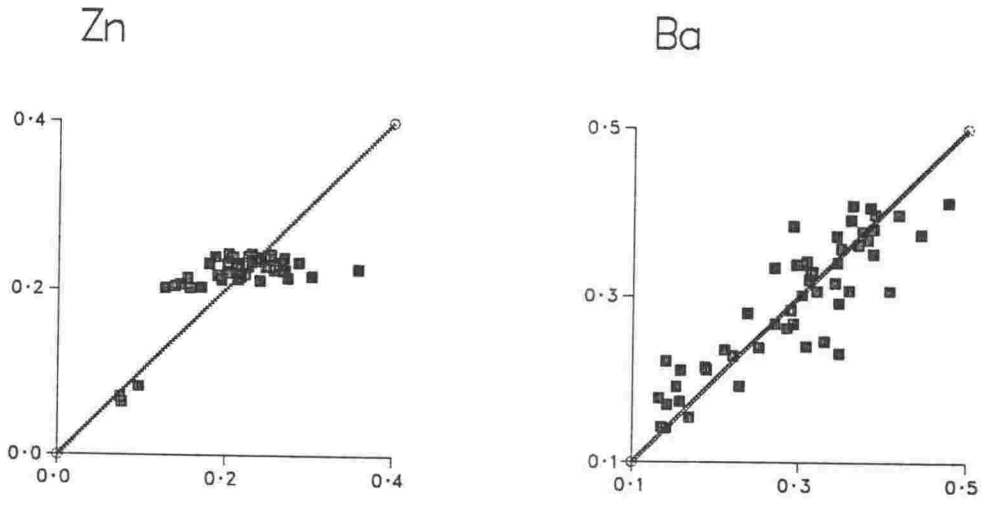


Figure 4.1. (continued)



4.2 MID-PACIFIC COBALT-RICH MANGANESE CRUSTS

The raw data for this second application came from the Mid-Pacific subset (170°E to 150°W, 18°S to 32°N) of cobalt-rich manganese crust data of the United States Geological Survey world ocean-ferromanganese-crust database (Lane *et al.*, 1986).

Measurements on p = 22 oxides SiO_2, TiO_2, MnO_2, Fe_2O_3, Al_2O_3, Co_3O_4, NiO, CuO, CaO, MgO, Na_2O, K_2O, CO_2, P_2O_5 and H_2O, and minor elements As, Ce, Mo, Pb, Sr, V and Zn featured in the analysis. Athough in many cases only the lower limits of detectable concentrations had been recorded for the minor elements. Sixteen samples were found to have exceptionally large individual angular deviations from their corresponding orthogonal projections in an estimate space of 10 dimensions, following an exploratory singular value decomposition. Of these, 3 were heavily contaminated with serpentinite or other material and were excluded, 4 had MnO_2/Fe_2O_3 ratios greater than 7.5 and were also excluded on the grounds of having a significant hydrothermal component. The remainder were found either to have errors which were corrected, or to have genuine outliers which indicated faulty measurements, and were also excluded. Ultimately, the number of samples available for analysis totalled n = 275.

This data was scaled to sum to 100% creating $\mathbf{X}$ (275 × 22) which was then column transformed into $\mathbf{X}^c$ according to equation (3.51). A singular value decomposition of $\mathbf{X}^c$ determined that the relative magnitudes of the first 4 eigenvalues were 91.26%, 3.59%, 1.41% and 0.92%, which sum to 97.18% (see equations (3.38) and 3.39)). A rather parsimonious 4 endmember representation was conjectured to account for the data because the remaining eigenvalues at 0.61% or less characterized a rapidly diminishing variation along individual eigenvectors. The total of 15 out of 22 coefficients of determination (between the observed and estimated variables) which exceeded 0.5 (Table 4.3) for k = 4 increased only slowly by progressing to 5, 6 then 7

endmembers. (A fuller discussion of this issue appears in Section 5.2 concerning endmember hypothesis testing).

Four extreme vectors belonging to $\mathbf{X}^{'c}$ were used to initialize the iterative search for $\mathbf{B}^c$, employing the least squares solution for $\mathbf{L}$ and the weighted mean error vector coefficient (equations (3.46) and (3.48)) to adjust current endmembers. Iterations were stopped after 10 cycles when the mean squared error (3.50) had dropped to 6.9×10^{-4}.

The 4 resultant endmember compositions constructed by this method are set out in Table 4.3. Maximum values for each element are displayed in **boldface**. These endmembers can be identified with each of

(i) a silicate (clay) phase, rich in Si, Al, Mg, Na, K, retaining manganese oxides;

(ii) a cobalt-rich manganese oxide phase, with a high ratio Mn/Fe = 3.77 and rich in Co,Ni, but low in Cu;

(iii) a biogenic phosphate phase, highest in CaO, CO_2, P_2O_5 and Sr all with biogenous associations;

(iv) a hydrogenous phase with the ratio Mn/Fe = 0.85, high in Fe, As, Ce and Pb, and which are associated with the iron oxide phase.

The coefficients of determination (r^2) between the estimated values for each variable in the 4 endmember representation and their corresponding observed values are also set out in Table 4.3.

The iterative construction of a clay endmember with SiO_2 ($r^2 = 0.92$) and Al_2O_3 ($r^2 = 0.86$), and a biogenic endmember with CaO ($r^2 = 0.96$), CO_2 ($r^2 = 0.72$), P_2O_5 ($r^2 = 0.90$) and Sr ($r^2 = 0.77$), is extremely reassuring but not extremely interesting. Sources such as these, would be expected to contribute

components to marine manganese deposits over extensive regions of the ocean floor.

Turning to the other two endmembers, 11 out of 22 of the variables, consisting of the oxides MnO_2, Fe_2O_3, Co_3O_4, NiO and H_2O and the elements As, Ce, Mo, Pb, V and Zn, were found to be most highly concentrated on either the cobalt-rich or the hydrogenous endmembers (Table 4.3). Their coefficients of determination (Table 4.3) range from 0.10 (H_2O) to 0.96 (MnO_2). The goodness of fit for each of these 11 variables can be assessed from Figure 4.2 which displays plots of the estimated against their observed values. There are 275 points on each plot which, ideally, would lie on a line through the origin with slope 1. Evidently, the plot for As ($r^2 = 0.64$) is fair, and those for Ce ($r^2 = 0.48$) and Zn ($r^2 = 0.37$) are poor. It would have to be concluded that H_2O ($r^2 = 0.10$) had not been fitted at all. Adopting the rule that a value of $r^2 < 0.5$ indicates an inadequate estimate, these latter 3 elements are not explained by this 4-endmember mixing process. Otherwise, the remaining 7 plots appear satisfactory.

Table 4.3

Endmember Compositions (%) Iteratively Adjusted to Fit Partitioning by Least Squares and Coefficients of Determination (r^2) Between Estimated and Observed Values for Mid-Pacific Cobalt-Rich Manganese Crust Data

Element	Silicate	Cobalt-rich	Biogenic	Hydrogenous	r^2
SiO_2	**32.83**	0.00	1.78	9.50	0.92
TiO_2	**2.41**	1.36	0.91	2.31	0.44
MnO_2	14.67	**60.46**	30.64	33.40	0.96
Fe_2O_3	16.88	14.48	11.41	**35.10**	0.83
Al_2O_3	**10.54**	0.00	0.90	1.07	0.86
Co_3O_4	0.45	**2.40**	0.29	0.67	0.76
NiO	0.45	**1.23**	0.68	0.17	0.83
CuO	**0.16**	0.07	0.12	0.10	0.06
CaO	3.83	3.47	**25.12**	2.53	0.96
MgO	**3.79**	2.50	1.56	1.29	0.25
Na_2O	**2.97**	2.92	1.89	2.17	0.22
K_2O	**2.09**	0.78	0.44	0.36	0.62
CO_2	0.64	0.39	**3.02**	0.30	0.72
P_2O_5	0.64	0.48	**13.93**	0.60	0.90
H_2O	7.52	8.69	6.61	**9.62**	0.10
As	0.000	0.024	0.018	**0.036**	0.64
Ce	0.004	0.108	0.114	**0.182**	0.48
Mo	0.000	**0.093**	0.075	0.045	0.79
Pb	0.000	0.203	0.150	**0.244**	0.71
Sr	0.032	0.168	**0.203**	0.186	0.77
V	0.019	0.068	0.070	**0.080**	0.69
Zn	0.069	**0.102**	0.078	0.050	0.37

Figure 4.2 Least squares estimates *vs.* observed compositions for Mid-Pacific data. The estimates were obtained by projecting the raw data orthogonally into the 3-dimensional estimate space, then rescaling to form compositions.

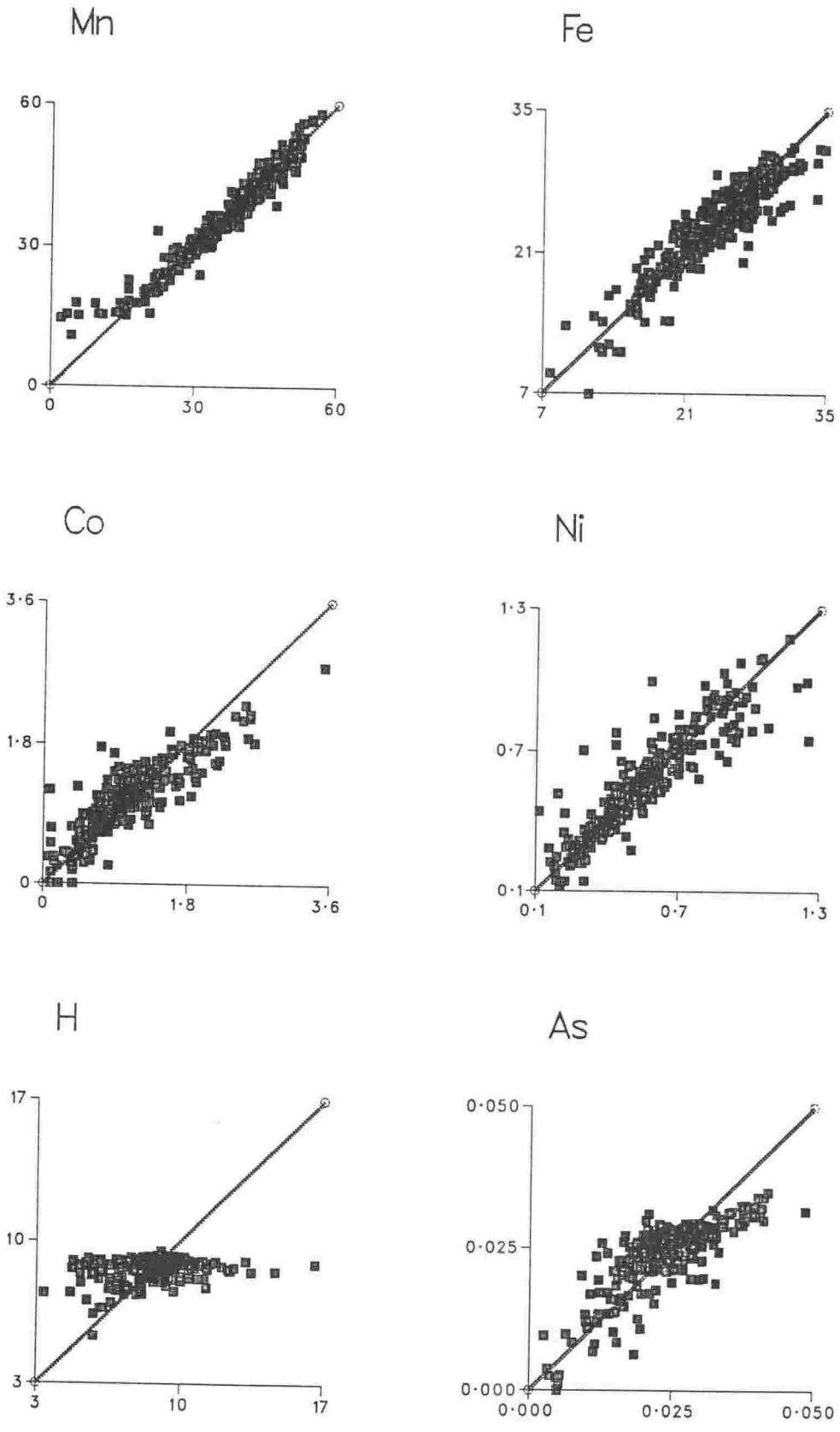
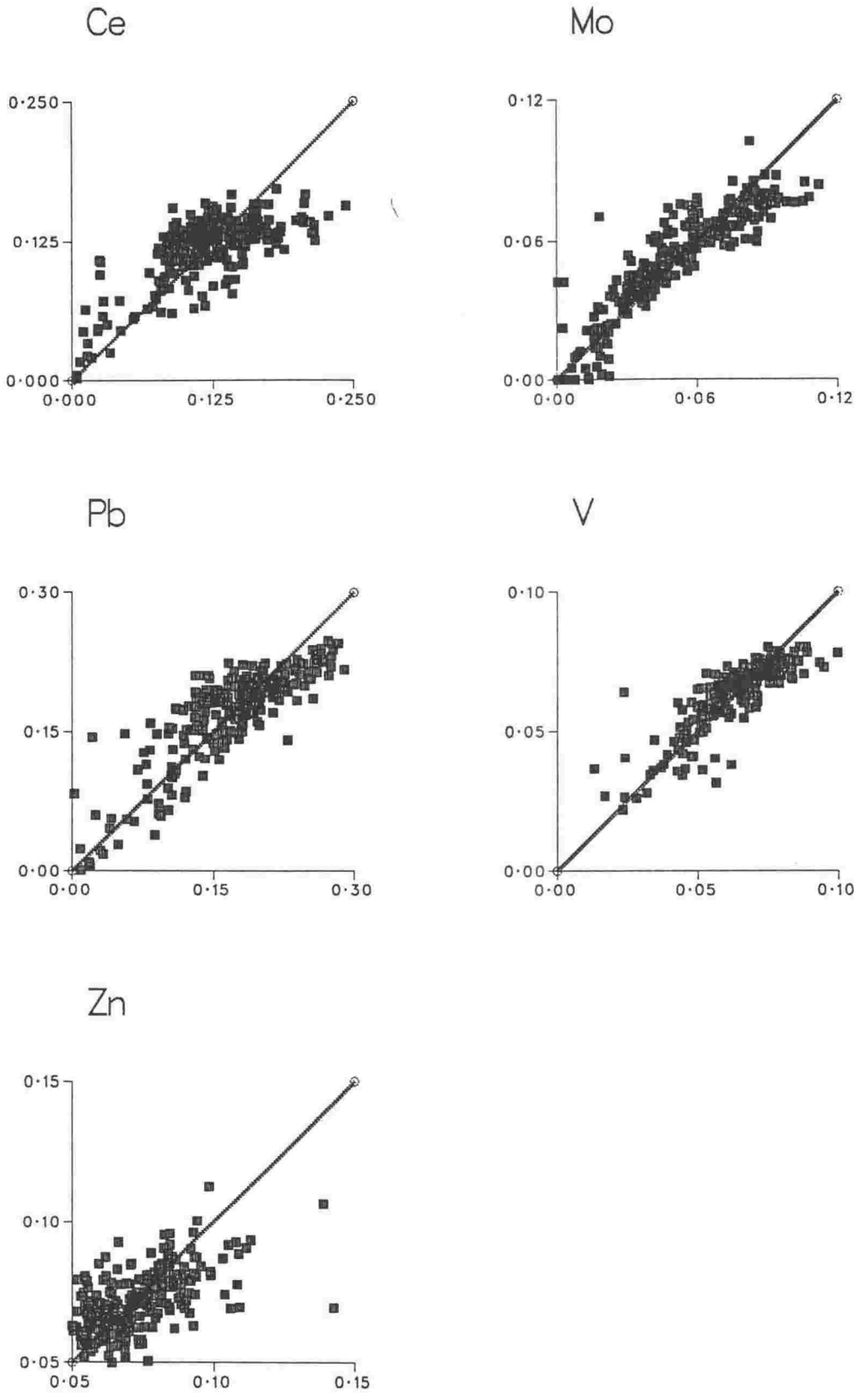


Figure 4.2. (continued)



4.3 BEDIASITE SOURCE MATERIALS

Love and Woronow (1988) have outlined a procedure to determine the minimum number of endmembers in an endmember mixture, and estimate the compositions of those endmembers. Their techniques include an examination of the hypothesis of complete subcompositional independence and, if that is rejected, tests on the correlation matrices of mixtures of proposed endmembers to determine which endmembers, if any, contribute to the observed data. The transformation of the raw data to logratios and the subsequent application of statistical tests based on the multivariate normal distribution (see Aitchison (1986)) are innovations in the study of the problem of resolving compositional datasets into mixtures of latent endmembers. Reporting on the application of their procedure to an array of 31 bediasite compositions, Love and Woronow (1988) concluded that a mixture of just two endmembers 'does satisfy the data', and they provided inner and outer endmember compositions for generating a two-endmember representation.

The purpose of this comment is to demonstrate that such a representation is not compatible with the relative positions of the compositional data-points in 9-space, and consequently does not create satisfactory approximations to the bediasite compositions.

1) If just two endmembers do satisfy the data, then within a tolerable error, the 31 bediasite compositions (Love and Woronow (1988), Table 4.4) are the position vectors of 31 collinear points in 9-space. Further, the endmembers will be the extreme points of that collinear set. This is a geometrical consequence of the conventional endmember 'mixing model' (see Figure 3.1). Such collinearity is invariant under transformations to subcompositions (see Aitchison (1986)) of rank greater than 2 (see Section 3.1.1), as well as column transformations such as changes of scale (see Section 3.4.4). Thus, given the diverse magnitudes of the ranges of the 9 major oxides in the bediasite data, a tolerable error would have to imply evident collinearity even following the column

transformation equivalent to the division of each major oxide by its observed maximum (Miesch (1976b, 1980)).

Accordingly in this analysis, the 31 bediasite composition vectors were first rescaled to sum to 100%, thus locating the 31 datapoints on an 8-dimensional hyperplane in 9-space. Then the 4 inner and outer endmember compositions (Love and Woronow (1988)) were appended, and finally this enlarged array was column transformed, each major oxide being divided by its maximum. The rows of the resultant (35×9) matrix **X** are the position vectors with respect to the origin of the transformed datapoints, which now lie on a second 8-dimensional hyperplane in 9-space.

It is an intuitively obvious result of algebraic geometry that the orthogonal projection of a straight line onto any plane is another straight line, except if the former is normal to that plane. So the 35 transformed datapoints representing the 31 bediasites and the 2 pairs of inner and outer endmembers were projected onto each of two mutually orthogonal planes as displayed in Figure 4.3.

Although any pair of non-parallel planes would have served, the chosen two provide perspectives of the greatest spread of the datapoints from the origin. This is because each of the planes is spanned respectively by two of the three 9-dimensional eigenvectors associated with the 3 largest eigenvalues of the symmetric (9×9) matrix $\mathbf{X}^T\mathbf{X}$. (A discussion of the properties of this matrix, which is not the covariance matrix of a principal components analysis, can be found in the treatments of the singular value decomposition of rectangular matrices in Section 1.5, Section 2.2.1 and Section 3.4.1).

In Figures 4.3 the projections of points representing bediasites are square, those representing the inner and outer endmembers are circular, joined respectively by straight lines.

It is apparent from Figures 4.3 (a) and (b) that neither pair of endmembers are at the extremities of an approximately linear set of points defined by the bediasites. Nor do their inner and outer properties appear to have any geometrical meaning.

2) The conventional check on the on the validity of a derived endmember representation is to tabulate the coefficients of determination between the estimated and observed values of all the variables (after Miesch (1976b)). In fact, if a representation is good, then, for any variable, the pairs of estimated and observed values determine points which must lie close to a straight line through the origin, with slope one. It is sufficient in this case to follow convention. Estimated composition vectors were formed first by projecting the observed vectors orthogonally into the plane spanned by the two outer endmember vectors, and then scaling the position vector of each projection to form a mixture. That is, so that the coefficients of the two endmember vectors in the resultant linear combination summed to one (see Section 3.2.1). Coefficients of determination between the estimated and corresponding observed values of each of the variables, are set out in Table 4.4. Five out of 9 of the coefficients are less than 0.5.

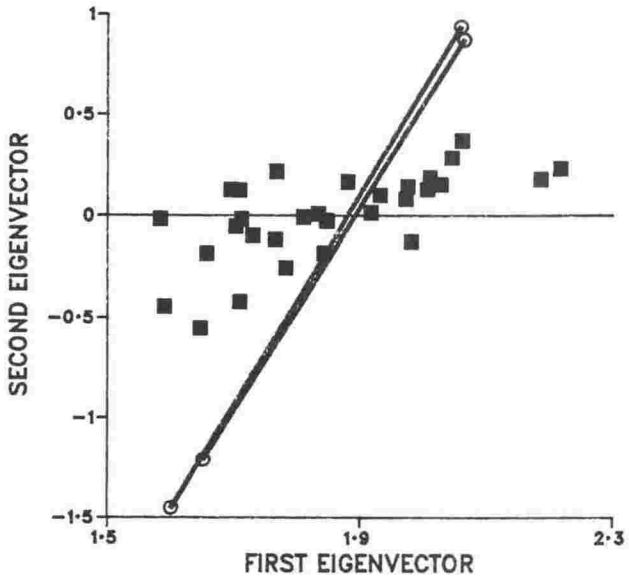
It is not possible on the basis of the uniformly low values of those coefficients to conclude that the two-endmember representation accounts for the given data.

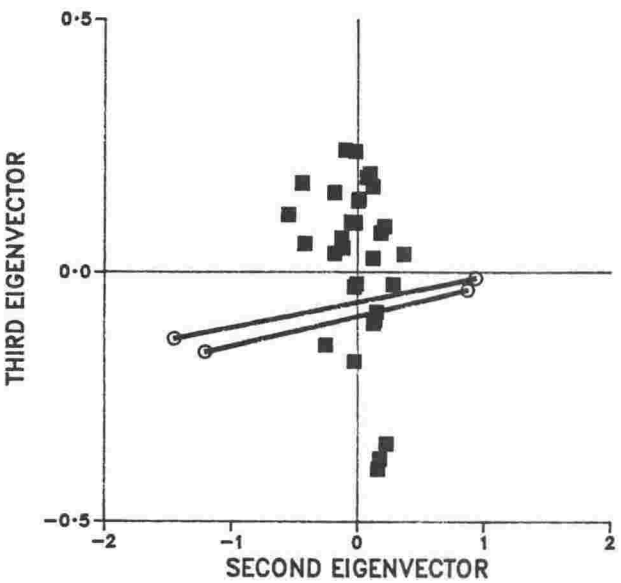
Table 4.4

Coefficients of Determination
Between Estimated and Observed Variables
in Woronow and Love's Two-Endmember
Representation of Bediasite Data

Si	0.43	Ti	0.71	Al	0.63
Fe	0.56	Mn	0.47	Mg	0.14
Ca	0.45	Na	0.43	K	0.76

Figure 4.3. The orthogonal projections of 31 bediasite data points and 2 pairs of inner and outer endmembers due to Woronow and Love (1988) onto each of two mutually orthogonal planes. Endmember pairs are joined by straight lines.





3) There is a deeper difficulty concerning the original (31×9) array $\mathbf{X}_0$ of this analysis. The singular value decomposition of the data, when column transformed as above, nevertheless yields a largest eigenvalue which accounts for 97.30% of the total for the 9 eigenvalues.

(These are the 9 non-zero eigenvalues of a symmetric matrix of the type $\mathbf{X}_0^T\mathbf{X}_0$ defined above, but not including the 4 inner and outer endmembers). Projecting the column transformed datapoints onto the unit hypersphere as for a Q-mode factor analysis (see Section 2.2.1) produces an almost identical result. Consequently even the column transformed data must be quite densely clustered about the eigenvector associated with the first eigenvalue.

Such a configuration also calls into question the procedure which resulted in the initial rejection of the hypothesis of complete subcompositional independence (Love and Woronow (1988)).

4.3.1 Further Comment

The preceding discussion illustrates the consequences of an inappropriate choice of the estimate space S, in this case straight lines in the positive orthant defined by the pair of estimated inner and outer endmembers. However, Woronow and Love (1988) made more fundamental errors before they set out to estimate their endmember compositions. In their abstract, Woronow and Love (*ibid*) asserted that 'the bediasites, being random samples of endmember mixtures, afford opportunities to establish a paradigm for endmember identification, determine the minimum number of lithologic/geochemical endmembers contributing to the bediasite compositions, and estimate the major-element chemistries of those endmembers'. Further on, in their section on statistical methods, they cautioned that 'logratioed data must be tested for multivariate normality, as the standard statistical procedures assume that underlying

distribution of the data'. A little later they stated that in the bediasite study 'the logratioed data passed the radius test for normality'.

The problem of choosing random geological samples has already been discussed in Section 3.1.3. In order to select random samples of 'endmember mixtures' as Woronow and Love claimed to have done, they would have had to have known or at least assumed the distribution of mixture coefficients on the sample space in order to define the geological equivalent of a probability sample (of samples). If what they really meant was that the selection process was based on a uniform probability distribution over the known collection(s) of bediasites, then that would not define a random sample of endmember mixtures at all. However, this is a general problem of geological data collection which has no clear solution because of the nature of mixing processes, particularly unknown mixing processes. It does not necessarily impair a mixture analysis unless an unwitting but substantial design bias in favour of one or more endmembers reduces the contributions of the remainder to the level of the errors.

What appears to be more difficult to understand is their assertion that the 'logratioed' data should be multivariate normal. Woronow and Love (*ibid*) did not define or even describe the model that they set out to test for. There is no mention for example of logratioed errors or residuals (see Chapter 5), but rather, of the properties of the covariance matrix of logratioed closed (constant sum) data. It can only be concluded therefore that the logratioed data referred to were derived from the (31×9) compositional data matrix $\mathbf{X}_0$. But, by equation (3.20), these particular logratios cannot follow a multivariate normal distribution unless the μ_{ij}/μ_{ip} are the components of a multivariate normal distribution, or are constant for i = 1,2,...,31. Since $\boldsymbol{\mu}_i$ is a composition vector all i, this latter condition would imply that $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \ldots = \boldsymbol{\mu}_{31}$, or that the data could be accounted for by one endmember. Confirmation of that possibility exists in the very high proportion (97.30%) of the total sum of squares for the column transformed data, achieved by the first eigenvalue.

4.4 LAKE TE ANAU SEDIMENTS

Lake Te Anau is one of 11 large glacial lakes formed on the eastern flanks of the Southern Alps of the South Island, New Zealand. With an area of 347 km^2, it is the largest lake in the South Island. It is also the 19th deepest in the world. The lake has three fiord arms which extend Northwest and West to the Southern Alps (see Figure 4.4). Until the present, almost no information on sediment input or trace element content of sediments existed for this lake.

In 1986, Operation Raleigh conducted a detailed sampling of sediments from Lake Te Anau, representing the first extensive survey of the trace element geochemistry of sediments within a single lake in New Zealand. In all, 108 locations were sampled and n = 102 analyses were made available for this study.

The compositions of a set of endmembers were determined using the sequence of procedures described in Section 3.4 and 3.5. The variable list for this analysis contained the major element oxides SiO_2, Al_2O_3, CaO, MgO, Na_2O, K_2O, TiO_2, Fe_2O_3, MnO, P_2O_5, LOI, and the minor elements V, Cr, Ba, Zn, Cu, Ni and Co, making p = 18. Both $CaCO_3$ and Corg (organic carbon) had been deleted from the list because their measurements were already included in those for CaO and LOI (loss on ignition). Since the list contained major element oxides and LOI (measured in percentages), and minor elements (measured in ppm), the latter were converted to percentages and the components of each sample corrected to sum to 100%. Prior to all singular value decompositions, iterative least squares partitioning and adjustment of extreme vectors, the observations on each variable had been divided by the maximium for that variable. That is, the compositional data matrix was column transformed to acheive similar weightings for each of the variables (see Section 3.4.4).

It was found that the sum of the first 5 eigenvalues obtained from the singular value decomposition of the column transformed data matrix accounted for 98.94% of the total sum of squares (equation (3.38)). When the datapoints (rows) of this matrix were orthogonally projected into the space spanned by the first 5 eigenvectors to determine the estimated mixture datapoints, only two coefficients of determination (r^2) between the observed values of the variables and their estimates were less than 0.5. Thus more than 50% of the variation in each of the remaining 16 variables was explained by the 5-dimensional approximation. Accordingly, a 5-endmember representation was constructed to account for the original compositional data. Five extreme vectors from the estimated mixture matrix were used to initialize the iterative algorithm, using the mean error vector coefficients (equations (3.46) and (3.48)), which was stopped when the the mean squared error (equation (3.50)) had fallen monotonically to 2.8×10^{-5}. The 5 endmember compositions and the coefficients of determination (r^2) for each element that were achieved by this representation are listed in Table 4.5. The maximum values for each element of the table are displayed in **bold face**.

In addition, each sediment was partitioned into the components of a mixture of the five endmembers. The proportional contribution of each endmember to the composition of a sediment is a measure of the abundance of that endmember at the location (sampling point) from which the sediment was taken. In Figure 4.4 a map of the lake has been shaded to show the regions in which each endmember was found to be the most abundant. Each region is defined by a collection of neighboring sampling points for which a single endmember was dominant. However, if all endmember concentrations for a sediment were found to be less than 30%, then the map was unshaded in the vicinity of its sampling point.

All 5 endmembers were characterised by relatively high Si and Al levels, but a single clay endmember was not isolated by the analysis. Diatoms which could be a possible source of silica, are relatively abundant in the sediments, and their distribution

is still being studied.

Descriptions of the five endmembers, identified by the Roman numerals I - V respectively, are as follows:

I. High in Cr, Mg and Ni, negligible P, Ba and Cu. This element assemblage indicates the presence of material derived from ferromagnesian rocks (that is, basic plus ultrabasic rocks, greenstones). Sediments dominant in endmember I were taken from The southern sector in the vicinity of the Waiau River (Figure 4.4). This part of the lake is surrounded by Pleistocene outwash gravels.

II. Highest in Si, Na and K, lowest in Ti, Fe and Zn, and negligible P, LOI, Cu, Ni and Co. This element assemblage indicates the presence of material derived from acidic rocks. Sediments dominant in endmember II occur near the heads of the three fiords as well as one sample taken at the northern head of the lake (Figure 4.4). The locations from which these samples were taken are all surrounded by metamorphic rocks of the wet Jacket and Bradshaw formations.

III. Highest in Ti, Fe, P, LOI, V, Ba, Zn ,Co and high in Ca. Lowest in Si, Al and negligible Cr, Cu, Ni. This element assemblage indicates the presence of organic carbon (LOI) plus titanomagnetite and igneous apatite-bearing rocks. Only three samples were dominant in endmember III. These were located at the head of the northern arm of the Middle Fiord and at the head of North Fiord (Figure 4.4), both regions that are surronded by metamorphic rocks of the wet Jacket and Bradshaw formations. Accordingly, sediments that were dominant in endmember II tended to be subdominant in endmember III.

IV. Highest Ca and Mg, and negligible K, Mn and Ni. This assemblage may reflect material rich in amphiboles. Sediments dominant in endmember IV occur principally at the head of the lake (Figure 4.4) which is surrounded by metamorphic rocks of the Bradshaw formation, Darran Diorite and upper Eocene sandstones. Two samples dominant in this endmember were taken from the South Fiord (Figure 4.4) virtually opposite an intrusion of Darran Diorite.

V. Highest Al, Mn and Cu, high Fe and negligible Na and Cr. This assemblage indicates the presence of minor amounts of adsorbed transition metals (Mn, Zn, Cu, Ni and Co). Samples dominant in endmember V were found in the three major deep basins of the lake (Figure 4.4).

There were a number of sediments having no dominant endmembers, that is, all mixture coefficients were less than 30%. These samples were found off the Eglinton River, which is the shallow area between two major basins, and the Middle Fiord (Figure 4.4) . These samples appear to be transitional between the samples from the heads of the lake and those from the deep basins.

This mixture analysis has discriminated between the the sediments from near the heads of the lake and those from the major deep basins. For samples taken from the heads of the lake, the local geology appears to have determined the compositions of the sediments, as might be expected. Samples high in endmembers II and III were formed in those parts of the lake with the same surrounding geology, but these two endmembers reflect different local sedimentation conditions. Endmember II accounts for lithogenous sedimentation and endmember III accounts for organic sedimentation. The high organic content of the sediments rich in endmember III (see LOI under III on Table 4.5) was derived from decayed vegetation at the headwaters of the lake (Figure 4.4). Samples

from the deep basins are high in endmember V and represent fine-grained composite material with a significant carbon content but low sedimentation rate. Adsorption of transition elements onto the fine-grained clay materials occured in these samples.

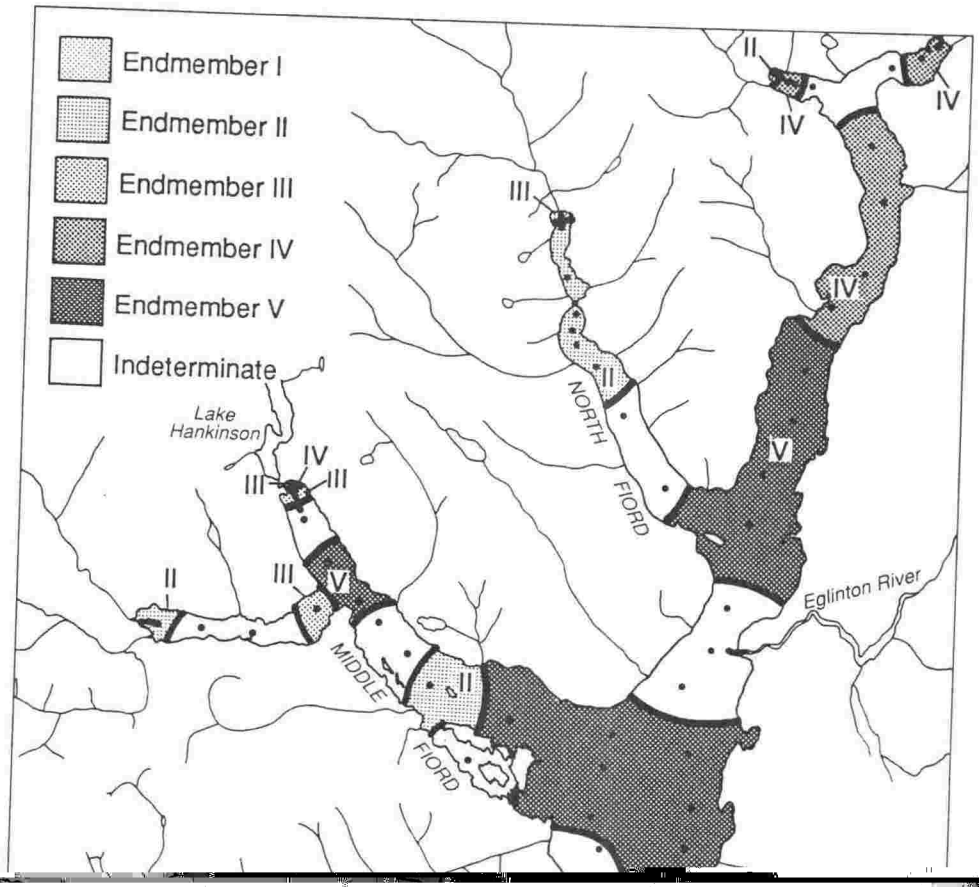
Imbrie and Van Andel (1964) remarked that, if the endmember map patterns are 'systematic with respect to known geological and hydrographical parameters, then the results may be accepted as both statistically and geologically significant' (see Chapter 2). They were describing contour maps formed from the values of each mixture coefficient rather than the single map of the distributions of dominant endmembers. Perhaps also, their claim for statistical significance was not quite appropriate. Nevertheless, the distributions of dominant endmembers displayed on Figure 4.4 are 'systematic with respect to known geological and hydrographical parameters', and would therefore seem to confirm a mixing process involving the 5 sources constructed by this analysis.

Table 4.5

Endmember Compositions (%) Iteratively Adjusted to Fit Partitioning by Least Squares, and Coefficients of Determination (r^2) Between Estimated and Observed Values for Lake Te Anau Sediment Data

Element	Endmember I	II	III	IV	V	r^2
SiO_2	61.08	**69.96**	27.24	46.56	51.04	0.83
Al_2O_3	16.02	16.50	10.65	15.33	**21.20**	0.69
CaO	2.47	3.07	3.71	**10.06**	0.91	0.88
MgO	5.89	1.16	2.29	**5.99**	3.83	0.93
Na_2O	1.87	**4.20**	1.06	3.67	0.00	0.78
K_2O	1.40	**2.78**	1.31	0.00	2.02	0.81
TiO_2	0.68	0.30	**1.78**	1.42	0.45	0.61
Fe_2O_3	7.63	1.95	**11.42**	9.05	10.82	0.82
MnO	0.19	0.02	0.26	0.00	**0.37**	0.56
P_2O_5	0.00	0.00	**0.89**	0.42	0.42	0.68
LOI	2.67	0.00	**39.29**	7.44	8.83	0.63
V	0.0144	0.0026	**0.0225**	0.0180	0.0189	0.72
Cr	**0.0384**	0.0025	0.0000	0.0091	0.0000	0.92
Ba	0.0000	0.0569	**0.0624**	0.0248	0.0477	0.40
Zn	0.0071	0.0035	**0.0181**	0.0059	0.0147	0.57
Cu	0.0000	0.0000	0.0000	0.0037	**0.0219**	0.98
Ni	**0.0247**	0.0000	0.0001	0.0003	0.0066	0.96
Co	0.0039	0.0007	**0.0058**	0.0027	0.0045	0.46

Figure 4.4. Lake Te Anau shaded to display regions in which endmembers I - V were found to be dominant. Regions which were found to have no dominant endmember are unshaded.



CHAPTER 5

APPROACHES TO TWO UNSOLVED PROBLEMS

SUMMARY

The presence of one or more missing values in a sample would normally force the exclusion of that sample from a mixture analysis. This is because algorithms constructed to process uniquely defined p-dimensional data cannot in general manipulate object vectors with undefined components. A possible solution to this difficulty in the case that a mixing process is believed to be present, is to exploit the overdetermined aspect of the mixture equations to impute values for those that are missing. This strategy is demonstrated for the well-researched database of Nazca Plate surface sediments.

Traditionally, the estimate for the number of endmembers has been assessed by mapping or by inspection of the coefficients of determination between the observed and estimated variables. Mapping entails the plotting on a map of the region from which the samples were taken, either the contours of the contributions of each endmember to each sample, or some other portrayal of the distribution of endmember abundances. Assessment by this method is too elaborate except for final confirmation and display. Alternatively, choosing a number of endmembers which results in suitably high coefficients of determination for all or most variables may account for elements which are not part of the conjectured mixing process. Even worse it may result in the identification of endmembers which do not in fact exist.

Another avenue for assessment lies in an examination of the distributions of certain logratios. The differences between corresponding logratio-transformed observed and estimated data form an array of residual logratios. A linear combination of these is formed for each sample which, under a random perturbation assumption should follow a univariate normal distribution. Whether or not this scalar is normal can be readily tested. It can also be examined graphically for such desirable qualities as symmetry when the test for normality may be too severe. This procedure is employed to assess the decompositions of the United States Geological Survey Mid-Pacific cobalt-rich manganese crust data and the Nazca Plate surface sediment data.

5.1 MISSING VALUES

Missing values are a common occurence in geochemical data. For example, different laboratories do not always analyse for identical lists of elements in the collections of samples which eventually form a single database. Quite often trace elements are present in a sample but in concentrations below the analytical detection level, and each of these will be recorded as an upper limit or simply as non-zero which is 'missing'. Mixture algorithms of the type described in this work can not process object vectors with undefined components. But, since the non-missing components in a sample contain valid information on any underlying mixing process that may account for the data as a whole, it would seem desirable to develop a method for imputing values for those that are missing which will permit the information in the non-missing components to be extracted.

Suppose that, without loss of generality, the dataset $\mathbf{X}$ (n×p) is partitioned so that the first r samples have missing values on the [q+1]-th to p-th variables (q < p). These missing values are all located in the top right (r×[p-q]) submatrix $\mathbf{X}_{12}$, where $\mathbf{X}$ is partitioned as in equation (5.1) below, the order of $\mathbf{X}_{11}$ being (r×q)

.

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_{11} & \mathbf{X}_{12} \\ \mathbf{X}_{21} & \mathbf{X}_{22} \end{bmatrix} \tag{5.1}$$

The array $\mathbf{X}'$ of estimated mixtures associated with $\mathbf{X}$, irrespective of the presence of missing values, is given by

$$\mathbf{X}' = \mathbf{L}\mathbf{B} \tag{5.2}$$

from equation (3.5). Partitioning equation (5.2) in the same way as equation (5.1),

$$\begin{bmatrix} \mathbf{X}'_{11} & \mathbf{X}'_{12} \\ \mathbf{X}'_{21} & \mathbf{X}'_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{11} \\ \mathbf{L}_{21} \end{bmatrix} \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \end{bmatrix} \tag{5.3}$$

where the orders of $\mathbf{X}'_{11}$, $\mathbf{L}_{11}$ and $\mathbf{B}_{11}$ are (r×q), (r×k) and (k×q) respectively, and $k \leq q < p$.

Three interrelated matrix equations can be extracted from equation (5.3). They are,

(i) from the botton (n-r) rows,

$$\begin{bmatrix} \mathbf{X}'_{21} & \mathbf{X}'_{22} \end{bmatrix} = \mathbf{L}_{21} \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \end{bmatrix} \tag{5.4}$$

(ii) from the first q columns,

$$\begin{bmatrix} \mathbf{X}'_{11} \\ \mathbf{X}'_{21} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{11} \\ \mathbf{L}_{21} \end{bmatrix} \mathbf{B}_{11} \tag{5.5}$$

(iii) from the top right submatrix corresponding to the block of missing values,

$$\mathbf{X}'_{12} = \mathbf{L}_{11}\mathbf{B}_{12} \tag{5.6}$$

These three equations follow from the approximate decomposition of (n×p) **X**. Arguing in reverse, it is anticipated that, assuming that a mixing process is present, the separate computations of the estimates (5.4) and (5.5) in that order, will allow the derivation of estimate (5.6). The singular value decompositions of $[\mathbf{X}_{21}\ \mathbf{X}_{22}]$ and $\begin{bmatrix} \mathbf{X}_{11} \\ \mathbf{X}_{21} \end{bmatrix}$ must establish a common approximate rank k, the estimated number of endmembers, and lead to similar estimates for $\mathbf{X}_{21}$.

Following the methods described in Section 3.4, a possible procedure for accomplishing this, is first, to project the rows of $[\mathbf{X}_{21}\ \mathbf{X}_{22}]$ orthogonally into the best

fitting k-dimensional subspace of p-space, forming $[\mathbf{X}'_{21}\ \mathbf{X}'_{22}]$. Then, the matrices $\mathbf{L}_{21}$ and $[\mathbf{B}_{11}\ \mathbf{B}_{12}]$ can be constructed satisfying equation (5.4). With the (k×q) submatrix $\mathbf{B}_{11}$ obtained in this way, it is possible to solve the overdetermined system $\mathbf{X}_{11} \approx \mathbf{L}_{11}\mathbf{B}_{11}$ for (r×k) $\mathbf{L}_{11}$. In practice, a safer estimate is probably obtained by solving the overdetermined system corresponding to equation (5.5) in which the left hand side is replaced by the first q columns of $\mathbf{X}$. In any event, the matrix $\mathbf{X}'_{12}$ of estimated mixtures corresponding to the submatrix of missing values can then be obtained by equation (5.6) since $\mathbf{L}_{11}$ and $\mathbf{B}_{12}$ have both been constructed. The matrix $\mathbf{X}'_{12}$ then replaces $\mathbf{X}_{12}$ in $\mathbf{X}$ so that a mixture analysis can be undertaken on all n samples.

It is not necessary in the computations that (k×p) $\mathbf{B} = [\mathbf{B}_{11}\ \mathbf{B}_{12}]$ be a matrix of proper endmembers. Any k rows of $[\mathbf{X}'_{21}\ \mathbf{X}'_{22}]$ will suffice. Indeed, bearing in mind the approximate nature of the representation, there may be merit in initially setting k = q. Since, if (q×p) $[\mathbf{X}'_{31}\ \mathbf{X}'_{32}]$ consists of q rows of $[\mathbf{X}'_{21}\ \mathbf{X}'_{22}]$ (provided q < [n-r] of course), then ultimately,

$$\left[\mathbf{X}'_{31}\ \ \mathbf{X}'_{32}\right] = \mathbf{L}_{31}\left[\mathbf{B}_{11}\ \ \mathbf{B}_{12}\right] \tag{5.7}$$

where $\mathbf{L}_{31}$ is of order (q×k). So a first approximation to equations of the type (5.4) to (5.6) can be made by deliberately overspecifying the system. That is, the singular value decomposition referred to above may be employed to identify the best-fitting q-dimensional space and the orthogonal projections of the raw data into that space corresponding to the left hand side of equation (5.4). (That will be possible provided the exact rank of $[\mathbf{X}_{21}\ \mathbf{X}_{22}]$ is equal to p. If it were less than p, then an exact mixture analysis would be carried out on $[\mathbf{X}_{21}\ \mathbf{X}_{22}]$ anyway). The right hand side of equation (5.5) may also be obtained by least squares (as in Section 3.2.1) but in terms of (q×q) $\mathbf{X}'_{31}$ instead of $\mathbf{B}_{11}$. Finally, $\mathbf{X}'_{12}$ would be obtained in terms of $\mathbf{X}'_{32}$.

Following the substitution of $\mathbf{X'}_{12}$ for $\mathbf{X}_{12}$ in the raw data $\mathbf{X}$ (n×p), the estimated endmember solutions (k×p) $[\mathbf{B}_{11}\ \mathbf{B}_{12}]$ would be constructed on a second pass through $\mathbf{X}$.

5.1.1 Nazca Plate Surface Sediments

Dymond (1981) reported that sediment samples for this study were selected primarily from cores recovered during cruises conducted by the Oregon State University and Hawaii Institute of Geophysics as part of the Nazca Plate project. Additional samples were obtained from the core collections of the Lamont-Doherty Geological Observatory and the Scripps Institution of Oceanography. Nearly all samples were taken from the 5 to 10 cm level of gravity cores. The various maps that were reproduced in Dymond's paper (*ibid*) indicated that the region from which the samples were taken lies to the West of the Peru-Chile Trench, from about 80°W to 120°W, and from the equator down to 40°S. A total of 425 analyzed samples were listed on microfiche which was the source of the data for this work. Two rows on the list were illegibly smudged, and were discarded. Thus there were available for this study a total of 423 samples each analyzed for the abundances of the 8 elements Al (%), Si (%), Mn (%), Fe (%), Ni (ppm), Cu (ppm), Zn (ppm) and Ba (%).

Due to contamination during storage, the values of zinc for 50 Lamont-Doherty samples were not recorded. Since 50 samples formed a relatively large proportion of the total dataset, it was these missing values which were imputed.

For all singular value decompositions, all calculations of the loading matrices of mixture coefficients (by least squares) and all measurements of the angles between the observed and estimated object vectors which follow, the values of each variable were first divided by its observed maximum (the column transformation of equation (3.51))

following the formation of compositions (sums to 100%). The final operation before reporting the results at any stage of an analysis was the inverse of the first column transformation.

Of the 423 samples available, 373 contained the abundances on all 8 elements. The trace elements were transformed into percentages, and, for the imputation operation, a 'fill-up value' (Aitchison (1986)) or remainder term as in equation (3.3), was included to complete the sum to 100%. These 373 partial compositions with no missing values thus constituted a (373×9) dataset with constant row-sums equal to 100%. The singular value decomposition algorithm was used to construct (373×9) $[\mathbf{X'}_{21}\ \mathbf{X'}_{22}]$, from which the (5×9) submatrix $[\mathbf{X'}_{31}\ \mathbf{X'}_{32}]$ was selected. Then a representation of the form (5.4) was obtained but in terms of (5×9) $[\mathbf{X'}_{31}\ \mathbf{X'}_{32}]$ rather than (5×9) $[\mathbf{B}_{11}\ \mathbf{B}_{12}]$. The least squares partitioning algorithm was used to construct the (373×5) loading matrix corresponding to $\mathbf{L}_{21}$ for this representation.

Zinc was then dropped from all 423 samples and replaced by a 'fill-up value' creating a second dataset of order (423×8) also with fixed row-sums equal to 100%. (It was shown in Section 3.1.2 that inclusion of the 'fill-up' term should not alter the rank of the estimates nor the loading matrix). Representation (5.5) was constructed by the least squares partitioning algorithm to determine (50×5) $\mathbf{L}_{11}$ and the redundant (373×5) $\mathbf{L}_{21}$ (for checking), using the previously identified submatrix (5×8) $\mathbf{X'}_{31}$ in place of $\mathbf{B}_{11}$. Hence a (50×1) vector of estimates for the missing values followed by substitution in equation (5.6). This made a third dataset of order (423×9) available for reanalysis.

The success of the imputation operation can be assessed by the scanning the angular errors for the 50 samples before and after the imputation. The column transformed (423×8) dataset was projected into the best fitting 5-dimensional subspace, and the angular deviations between the 50 pairs of observed and estimated object vectors were recorded as in table 5.1 column I. Similarly, the column transformed (423×9)

dataset was projected into the best fitting 5-dimensional subspace, and the angular deviations between the 50 pairs of observed and estimated object vectors were also recorded as in table 5.1 column II. Scanning across rows, it is evident that the angles in Column II tend to be somewhat smaller than the corresponding angles in Column I. Indeed the mean angular deviations for Column I and Column II are 5.24° and 5.07° respectively. That is, following the imputation, an observed vector tends to be somewhat closer to its estimate in the 5-dimensional space than before imputation.

An unexpected result of this study was the recognition that the mean angular deviation of the 50 samples from the Lamont-Doherty cores was significantly higher than that for the remaining samples. The overall mean angular deviation for the column transformed (423×8) dataset (from which zinc had been dropped) was 4.27° compared to 5.24° for the subset of 50 Lamont-Doherty samples. When all 423 angular deviations were transformed into rank order statistics and then partitioned into those from the Lamont-Doherty Observatory and those that were not, the mean rank of the former was 268.90 and of the latter was 204.37. Under an *a priori* assumption that these sets of deviations were random samples from the same distribution, the expected value for both mean ranks would have been 212. The departures from this were highly significant. An approximate chi-square statistic with 1 degree of freedom was found to be 12.28 (Mann-Whitney) with tail-end probability 0.0005.

All 423 samples contributed to the determination of the first 5 eigenvectors in 8-space (zinc having been excluded). But the 50 Lamont-Doherty samples were on average more remote from the space spanned by those eigenvectors than the remaining samples. The average magnitudes of the angular deviations do not appear so different as to suggest distinct mixing processes. It is possible that analyses from the separate laboratories were relatively biassed in some way.

Table 5.1

Angular Deviations Between Observed and Estimated Object Vectors for Samples with Missing Values Before Imputation I and After Imputation II

I	II	I	II	I	II
2.0931	2.0891	3.6701	3.7212	3.3671	3.4123
8.1990	7.5835	0.8491	0.8833	4.4274	4.3089
5.9426	5.6692	4.8758	4.7678	5.4740	5.3032
7.0654	6.6506	1.7905	1.8818	3.7512	3.5020
10.5604	10.3855	5.4181	5.3617	1.7714	2.0421
7.5001	7.5141	1.5678	1.5688	1.5619	1.6454
5.9448	5.6592	10.8046	10.8514	6.8641	6.3743
5.8063	5.5537	4.7376	4.5971	7.6982	7.1464
18.3168	16.6355	8.1353	7.8516	4.9942	4.7061
6.0612	6.0148	3.5919	3.5805	5.6347	5.3427
4.9434	4.8476	2.8937	2.7458	0.7457	0.7693
5.1393	4.9965	4.2565	4.1274	1.4567	1.4635
7.3840	7.2590	8.5833	8.6493	1.8217	1.8217
6.4419	6.1510	7.3922	6.7700	1.9096	1.8905
9.6260	9.0154	7.6267	7.4957	3.4097	3.4200
4.9808	4.7869	4.3296	4.1831	2.5374	2.6239
5.7115	5.5771	2.4708	2.3919		

Dymond's description (*ibid*) of the normative analysis and partitioning by linear programming of the data into Hydrothermal, Biogenic, Detrital, Hydrogenous (Authigenic) and Dissolution residue components (endmembers) has already been described in Section 2.3. His paper has been widely cited in view of both his mode of analysis and the model he constructed for explaining the formation of marine sediments (see for example, Leinen and Pisias (1984), Walter and Stoffers (1985), Leinen (1987), Owen (1987), Chen and Owen (1989), Dean, Gardener and Parduhn (1989) and Nath, Rao and Becker (1989)). With 423 samples now available, the opportunity exists to

perform an independent mixture analysis on them without any prior assumptions concerning the compositions of the endmembers.

As was described in Section 2.3, the basis of Dymond's account was the table of 'elemental ratio coefficients of the five components used in the normative analysis', which were specified *a priori*. It is a simple matter to transform such a table into a collection of (sub)compositions **B** (k×p) which are in this case the conjectured endmembers for the data. If $\mathbf{B}_N$ (k×p) is the array of 'elemental ratio coefficients' as defined by equations (2.15) and (2.16) and $b_{i\alpha}$ is the concentration of the normalizing element in the i-th endmember, then

$$b_{Nij} = b_{ij} / b_{i\alpha} \tag{5.8}$$

for j = 1,2,...,p. The row-sums of **B** are each 1 so summing over the p components on either side of equation (5.8),

$$\sum_{\beta=1}^{p} b_{Ni\beta} = 1 / b_{i\alpha} \tag{5.9}$$

Substituting for $b_{i\alpha}$ in equation (5.8) from equation (5.9),

$$b_{ij} = b_{Nij} / \sum_{\beta=1}^{p} b_{Ni\beta} \tag{5.10}$$

for j = 1,2,...,p. In the case of Dymond (1981, Table 3) the 5 components were transformed as in equation (5.10) to form subcompositions (since there were only 8 elements and no residue). Each elemental abundance was recorded as a percentage as set out in Table 5.2 (a). Hence, Table 5.2 (a) contains the theoretical subcompositions of endmembers which have been derived as a direct consequence of Dymond's *a priori* assumptions. There are two possible methods for making an appraisal of these assumptions using the procedures described in Chapter 3. The first would be to project

the data (as a subcompositional dataset with 8 variables) into the space spanned by the vectors in Table 5.2 (a) and then to examine the the signs of the mixture coefficients and the magnitudes of the angular errors. The second would be to independently construct five endmember estimates from the data and compare these with the vectors in Table 2 (a). The latter method has been followed here.

Four separate mixture analyses were conducted. The first two on the (373×8) and (423×8) datasets of subcompositions respectively. The third and fourth on the (373×9) and (423×9) datasets of partial compositions. In each case, the data matrices were column transformed (equation (3.51)). Singular value decompositions attributed between 99.0% to 99.2% of the cumulative sums of squares to the first five eigenvalues in all 4 cases. The mean angular errors were of the order of 3^{o} (arccos 0.999) for the subcompositions and 4^{o} (arccos 0.998) for the partial compositions. For the iteration procedures, the two forms of the matrix **G** of error vector coefficients defined by equations (3.48) and (3.49) were chosen consecutively for each analysis, making a total of 8 sets of estimates. Iteration cycles were stopped when the mean squared error (equation (3.50)) was in the range 1.1×10^{-4} to 1.6×10^{-4} which seemed to be the best that could be achieved. In every case the mean squared error fell monotonically until the procedure was stopped.

It is not proposed to reproduce 8 tables of estimated endmembers here. All 8 contained 3 subcompositions that were remarkably similar to and therefore readily identifiable with the Detrital, Hydrothermal and Biogenic components of Table 5.2 (a). There was however some diversity in the estimates of the remaining two. Further, where the first 3 were always returned by the iterative algorithm when the initializing extremes were altered, the last 2 were by no means so stable. Table 5.2 (b) sets out as subcompositions the estimates obtained from the (423×9) dataset of partial compositions, applying the error vector coefficients defined by equation (3.49) for a mean squared error of 1.2×10^{-4}.

In geochemical terms, the component by component similarity between the first columns of Tables 5.2 (a) and (b) is remarkable, bearing in mind that Al, Si, Mn, Fe and Ba are major, while Ni, Cu and Zn are trace elements. Dymond's composition for this component was taken from 'summary analyses of igneous and sedimentary rock'.

The estimated endmember in the second column of Table 5.2 (b) is more extreme than the hydrothermal component in Table 5.2 (a). The elements Al, Si, Ni and Ba were driven down to zero by the iterative algorithm, while holding small values in Dymond's subcomposition. Iron, on the other hand, was somewhat higher than Dymond's value. Corresponding values for Mn, Cu, and Zn are in strong agreement and, taken together, it is clear that the iterative algorithm reconstructed the hydrothermal component.

Comparing the third columns of Tables 5.2 (a) and (b), what stands out is the similarity of the very high values for Si, although the last few decimal places above 99% meant the presence or absence of the other elements in the subcompositions. Again the iterative estimate was the more extreme but the nearly equal values for Ba are possibly notable. Dymond stated that the biogenic source is composed of predominantly biogenic opal (an amorphous form of hydrated silicon dioxide) and refractory organic matter.

For each of the two sets of 5 endmembers, the fourth columns of Table 5.2 (a) and (b) agree in being the dominant sources of Mn and Ni, as well as being high in Cu and Fe. But the values of the element concentrations hardly correspond at all. A similar pattern emerged for the last pair of endmembers. The fifth columns of Table 5.2 (a) and (b) display very similar Ba, an element with strong biogenic associations. Dymond chose it as the index element for this endmember which was supposed to consist of relatively insoluble elements of carbonate and siliceous organisms. Otherwise, the two components display high Al, Fe, Cu and Zn but not in comparable concentrations.

Table 5.2 (a)

Endmember Subcompositions (%) for Nazca Plate Data

Derived From Dymond (1981, Table 3)

Element	Detrital	Hydrothermal	Biogenic	Authigenic	Dissolution residue
Al	21.132	0.42	0.199	2.84	26.55
Si	63.396	9.04	99.484	8.52	0.00
Mn	0.338	20.17	0.002	56.82	0.37
Fe	14.793	69.54	0.099	28.41	18.58
Ni	0.032	0.06	0.004	1.89	0.35
Cu	0.025	0.29	0.005	0.95	0.85
Zn	0.030	0.13	0.008	0.19	0.21
Ba	0.254	0.35	0.199	0.38	53.09

Table 5.2 (b)

Endmember Subcompositions (%) for Nazca Plate Data

Computed by Least Squares Partitioning

Al	23.743	0.00	0.000	0.00	15.13
Si	64.084	0.00	99.784	0.00	0.00
Mn	0.000	20.15	0.000	46.18	4.06
Fe	12.104	79.42	0.000	51.78	27.36
Ni	0.022	0.00	0.000	1.41	0.26
Cu	0.017	0.28	0.000	0.60	1.11
Zn	0.030	0.15	0.015	0.03	0.45
Ba	0.000	0.00	0.201	0.00	51.63

The conclusion to be drawn from this analysis is that there is quite strong evidence in favour of Dymond's first three specified endmembers but the contributions, if any, to the bulk of the samples from his last two are too slight to allow their stable estimation. That raises the issue of the valid estimation of the number of endmembers, which will be examined in the next section.

The transformation of element ratios into the components of partial compositions naturally requires that the concentrations of the normalizing elements be known. Dymond supplied these concentrations which, he stated, were taken from the same literature sources that were used to obtain the elemental ratios found in Dymond (1981, Table 3). He assumed a value for (1) the concentration of Al in pure detritus to be 8.4%, (2) the concentration of Fe in pure hydrothermal material to be 34.8%, (3) the concentration of Si in pure biogenic opal to be 36.0%, (4) the concentration of Ni in hydrogenous (authigenic) material to be 1.0%, and (5) the concentration of Ba in the dissolution residue to be 27.0%.

It was noted above that the estimated endmember subcompositions set out in Table 5.2 (b) were derived from the estimated endmember partial compositions extracted iteratively from the (423×9) dataset. These partial compositions naturally contained the estimates of the concentrations of the specific 8 elements for each full composition of a 5-endmember mixing process. The estimated concentrations for the 5 normalizing elements in particular, together with Dymond's choices (from above) in parentheses, were,

(1) Al 9.8% (8.4%), (2) Fe 38.8% (34.8%), (3) Si 31.2% (36.0%),
(4) Ni 0.9% (1.0%), (5) Ba 22.2% (27.0%)

Considering each pair of concentrations at a time, there is an evident comparability between the first figure, constructed from the data, and the second, taken from the

literature. These figures are estimates of the concentrations of the normalizing elements in each of the conjectured sources in which they are extreme. In geochemical terms, there is no evidence in these figures to refute Dymond's choices.

Leinen and Pisias (1984) employing a Q-mode factor method (see Chapter 2) also analyzed the Nazca Plate sediment data. Apart from stating that the dataset contained 423 samples for which the concentrations of Al, Si, Fe, Mn, Cu, Ni, Zn, and Ba had been determined, they made no mention of missing values or of the number of samples that were included in their analysis. They converted the element concentrations of the raw data to oxides then formed subcompositions. In order to compare their terminal solution with Dymond (1981, Table 3), they had to recalculate their estimated endmember oxide subcompositions as element ratios. These ratios appear in Leinen and Pisias (1984, Table 3). By employing equation (5.10), the columns of this table have been transformed into 5 endmember element subcompositions which are displayed in Table 5.3.

Table 5.3

Endmember Subcompositions (%) for Nazca Plate Data

Derived From Leinen and Pisias (1984, Table 3)

Element	Detrital	Hydrothermal	Biogenic	Authigenic	Dissolution residue
Al	25.157	0.37	0.751	1.63	15.94
Si	49.987	1.42	68.264	73.36	0.00
Mn	4.830	22.81	4.505	20.95	17.34
Fe	19.522	74.78	23.346	0.00	34.74
Ni	0.277	0.16	0.000	0.82	0.55
Cu	0.146	0.32	0.130	0.39	0.67
Zn	0.050	0.13	0.068	0.05	0.21
Ba	0.030	0.01	2.935	2.81	30.53

Apart from their hydrothermal endmember, it is difficult to understand the claims made by Leinen and Pisias (1984) for high level of concordance with Dymond's components. Working from their table of 'endmember ratios', they concluded that the 'distribution patterns of detritus from the two techniques are virtually identical', for hydrothermal sediment the 'ratios of Al, Mn, Cu and Zn to Fe are virtually identical, for dissolution residue 'the composition of the factor analysis endmember is remarkably similar to the linear programming endmember except for Fe and Mn', and for the authigenic sediment they conceded that the two set of results differed most. Comparing the corresponding subcompositions of Tables 5.2 (a) and Table 5.3, it is evident that conclusions of great similarity between the results of the two techniques (that is, Q-mode factor and normative analysis) are a little exaggerated.

5.2 TESTING ENDMEMBER HYPOTHESES

This concluding section not only introduces an alternative approach to the problem of assessing the number of endmembers, but also brings together many of the most important ideas covered in the previous two chapters. Where it is necessary, those ideas are revised in summary form.

Mixture analysis, including normative analysis, partitioning by linear programming or principally Q-mode factor analysis, has become a well-established quantitative method over the last two decades (for a discussion and comparison of these three techniques, see Leinen (1987)). Its relevance to geochemistry is that it can be used to identify the systematic components of variation in large compositional datasets.

An illustration of a perfect mixing process is provided in the introduction to this thesis. In summary, suppose three rivers each bear sedimentary materials of fixed (1×p) compositions $\boldsymbol{\beta}_1$, $\boldsymbol{\beta}_2$ and $\boldsymbol{\beta}_3$ respectively, into a lake. These 3 source materials are

called endmembers, and $p > 3$ is the number of elements whose abundances form a composition. Assuming mixing takes place without contamination, a sample of sediment from the lake floor will have a (1×p) composition vector **x** given by,

$$\mathbf{x} = \lambda_1\boldsymbol{\beta}_1 + \lambda_2\boldsymbol{\beta}_2 + \lambda_3\boldsymbol{\beta}_3 \tag{5.11}$$

where $$\lambda_1 + \lambda_2 + \lambda_3 = 1 \tag{5.12}$$

and $$\lambda_1, \lambda_2, \lambda_3 \geq 0 \tag{5.13}$$

Equations (5.11), (5.12) and (5.13) identify **x** as a convex combination of the endmembers, and therefore the position vector of a point in the interior of a plane triangle in p-space. The position vectors of the vertices of the triangle are the 'extreme' compositions $\boldsymbol{\beta}_1$, $\boldsymbol{\beta}_2$ and $\boldsymbol{\beta}_3$.

In matrix form,

$$\mathbf{x} = \begin{bmatrix}\lambda_1, \lambda_2, \lambda_3\end{bmatrix}\begin{bmatrix}\boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \\ \boldsymbol{\beta}_3\end{bmatrix} = \boldsymbol{\lambda\beta} \tag{5.14}$$

where $\boldsymbol{\lambda}$ is (1×3) and $\boldsymbol{\beta}$ is (3×p). If n samples are taken from the lake floor, their compositions $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ all (1×p), constitute an (n×p) array **X** of exact rank 3 given by

$$\mathbf{X} = \boldsymbol{\Lambda\beta} \tag{5.15}$$

where (n×3) $\boldsymbol{\Lambda}$ is the matrix of mixture coefficients.

In reality, even if a mixing process has been at work, nature never delivers compositional data matrices of exact low rank like 3 in the illustration above. It is

therefore assumed that nature created

$$\mathbf{X} = \mathbf{\Lambda\beta} + \boldsymbol{\varepsilon} \tag{5.16}$$

In this model $\mathbf{X}$, is (n×p), $\mathbf{\Lambda}$ is (n×κ) and $\boldsymbol{\beta}$ is (κ×p) where κ is the number of true endmembers. Error matrix $\boldsymbol{\varepsilon}$ is of course (n×p) and represents the non-systematic contribution to the data.

Since matrices $\mathbf{\Lambda}$, $\boldsymbol{\beta}$ and $\boldsymbol{\varepsilon}$ are usually unknown, the estimated mixing model is,

$$\mathbf{X} = \mathbf{LB} + \mathbf{E} \tag{5.17}$$

In this representation, $\mathbf{X}$ is the same as in equation (5.16). The matrix of estimated mixture coefficients $\mathbf{L}$ is of order (n×k) and the matrix of estimated endmembers $\mathbf{B}$ is (k×p) where k is the estimated number of endmembers. The matrix of residuals is $\mathbf{E}$ (n×p).

An important matrix for the purposes of interpretation is (n×p) $\mathbf{X}'$ given by

$$\mathbf{X}' = \mathbf{LB} \tag{5.18}$$

The matrices $\mathbf{X}$, $\mathbf{X}'$, $\mathbf{\Lambda}$, $\mathbf{L}$, $\boldsymbol{\beta}$ and $\mathbf{B}$ have two properties in common. First, every matrix element must be non-negative and second, each row-sum must be 1 (or 100%). These two conditions define a composition (Aitchison (1986)) which therefore applies to every row in each matrix. Further, since the rows of (n×p) $\mathbf{\Lambda\beta}$ define points within the convex hull of the points whose position vectors are the rows of $\boldsymbol{\beta}$, equation (5.16) represents a convex model for the particular random experiment under study.

It follows from the constant row-sum assumption that the row-sums of $\boldsymbol{\varepsilon}$ and **E** must each be zero.

The matrix **X'** is the estimate of the matrix of true mixtures $\boldsymbol{\Lambda\beta}$. It is of exact rank k. The original data matrix **X** is of 'approximate rank' k if each of its rows is well-approximated by a linear combination of any k linearly independent (1×p) vectors, and that is essentially a subjective concept. Nevertheless, in view of the introductory illustration and from equations (5.16), (5.17) and (5.18), it is clear that if **X** were determined by a κ-component mixing process contaminated by small non-systematic errors, then the chosen approximate rank of **X** should be close to κ. That raises an interesting question: which then would be the more serious error, to choose k less than κ, the unknown true number of endmembers, or to choose k greater than κ ?

In the case that all variables have equal weight (by transformation to fractional ranges or otherwise) and endmember estimates are chosen in order of their remoteness from each other, the answer to the question is that to choose $k > \kappa$ is the more serious error. This is because,

(i) $k > \kappa$ implies the identification of source components which do not actually exist,

(ii) the presence of such false components (as rows) in **B** may result in non-trivial concomitant estimated mixture coefficients (elements of **L** in equation (5.17)) for samples that are near such components and,

(iii) elements may be accounted for in the estimate which do not feature in the true mixing process.

(The geometrical view is that the estimated endmember compositions are the position vectors of the k vertices (extremes) of a convex polytope which must contain all the estimated data points (rows of **X'**). If $k > \kappa$ then extra vertices have been added to

include in the estimated polytope, departures from the true polytope due to errors).

In the mixing model defined by equation (5.16), the rows of β span $\mathcal{S}$, a κ-dimensional subspace of the positive p-orthant. In the estimated model, equation (5.17), the rows of **B** span S a k-dimensional subspace of the positive p-orthant. The optimistic stance is that S will be located close to $\mathcal{S}$. For example, if there were just 2 true endmembers and hence the data resembled a 'fuzzy' line then S would be a line through the 'fuzz'. Choosing k correctly equal to 2 in this example would not of itself yield a a satisfactory solution if the line S were located through some region of p-space remote from $\mathcal{S}$ (see Section 3.1, Figure 3.1).

Roughly summarizing Section 3.4, an analysis of mixtures can be conducted according to the following 5 steps.

1. Choose k-space S (for example, by the singular value decomposition of (n×p) **X** or any non-singular transformation of **X**).
2. Project the rows of **X** orthogonally (by least squares) into S to form (n×p) **X'**
3. Test the validity of the choice of S.
   (i) by mapping.
   (ii) by inspection of the p coefficients of determination (r^2) between corresponding pairs of observed and estimated variables, that is between corresponding columns of (n×p) **X** and (n×p) **X'**.
   (iii) *by examining the residuals formed following the transformation of the observed and estimated data to logratios.*
4. Locate initial extremes (k×p) $\mathbf{B}_0$ and compute the mixture coefficients (n×k) $\mathbf{L}_0$ by least squares.
5. Iteratively construct the terminal solutions (k×p) **B** and (n×k) **L**, exploiting the magnitudes of least squares regression coefficients.

Steps 4 and 5 summarize the iterative construction of the estimate equation (5.17) which is described in Section 3.4.3 and Section 3.4.5. It suffices to say that, contrary to statements that have appeared in the literature, the least squares method constructs the 'best' estimate when it is required to partition a given sample into specified endmembers. Further, the possible occurrence of negative values for some regression coefficients indicates not only that at least one endmember estimate is not extreme enough, but also that the magnitudes of the remaining positive coefficients determine the adjustments to be made to the non-extreme endmembers.

Steps 1 through 5 outline an integrated approach to a mixture analysis the actual detail of which is substantial. It is desirable that an assessment of the estimate of the dimensionality of the data be made at the earliest stage possible, which in this strategy is at Step 3.

Mapping, Step 3 (i), usually takes the form of a contour plot of the columns of (n×k) **L** or some other portrayal of the distributions of the estimated endmembers. Naturally these are displayed on a map of the region from which the samples were taken and are quite elaborate to prepare. Mapping is actually out of sequence in this scheme because it requires a terminal solution for equation (5.17). So, it is usually used for the final confirmation of the geographical continuity of endmember abundances and, of course, for descriptive purposes. It was initially demonstrated by Imbrie and Van Andel (1964).

Inspection of the p coefficients of determination, Step 3 (ii), was recommended by Miesch (1976b) and remains a most severe and relatively quick test of the validity of a decomposition of the form equation (5.18). The problem seems to be that concentrating on the precision of all or most the estimated variables is likely to force elements which may not belong to a natural mixing process into the endeavour to model it. Rather like regression models with unknown numbers of explanatories, this

enhances the risk of overspecification which in a mixture analysis results in the inclusion of source components in the physical model which do not in fact exist.

The formation of logratios, Step 3 (iii), highlighted in italics, implies remodelling the residuals. The difference of the logratios of corresponding components of $\mathbf{X}$ and $\mathbf{X}'$ is a logratio residual whose behaviour may be predicted approximately provided that $\mathbf{X}'$ is indeed close to (unknown) $\mathbf{\Lambda\beta}$, and the values of the logratio residuals have been determined by chance mechanisms which permit the valid application of the multivariate central limit theorem.

Aitchison (1986) discussed the situation where a composition evolves over time into another composition. The latter composition is called a perturbed composition. It is possible to calculate a perturbing vector whose components scale the components of the original vector. A summary follows of the basic algebra, which was even more briefly covered in Section 3.1.3.

$\boldsymbol{\mu}$ (1×p) is a primeval composition vector, that is $\sum_{j=1}^{p} \mu_j = 1$, where $\mu_j \geq 0$ all j.

$\boldsymbol{\rho}$ (1×p) is a compositional 'perturbing' vector. The perturbation of $\boldsymbol{\mu}$ is defined to be

$$\boldsymbol{\mu} \circ \boldsymbol{\rho} = \left[\mu_1\rho_1,\ \mu_2\rho_2, \ldots, \mu_p\rho_p\right] / D(\boldsymbol{\mu},\boldsymbol{\rho}), \quad D(\boldsymbol{\mu},\boldsymbol{\rho}) = \sum_{j=1}^{p} \mu_j\rho_j \qquad (5.19)$$

For any constant A,

$$(A\boldsymbol{\mu}) \circ \boldsymbol{\rho} = \boldsymbol{\mu} \circ (A\boldsymbol{\rho}) = \boldsymbol{\mu} \circ \boldsymbol{\rho} \qquad (5.20)$$

If $\boldsymbol{\tau}$ is a second perturbing vector then it follows from equations (5.19) and (5.20) that,

$$\boldsymbol{\mu} \circ \boldsymbol{\rho} \circ \boldsymbol{\tau} = (\boldsymbol{\mu} \circ \boldsymbol{\rho}) \circ \boldsymbol{\tau} = \left[\mu_1\rho_1\tau_1,\ \mu_2\rho_2\tau_2, \ldots, \mu_p\rho_p\tau_p\right] / D(\boldsymbol{\mu},\boldsymbol{\rho},\boldsymbol{\tau}) \qquad (5.21)$$

where $D(\boldsymbol{\mu},\boldsymbol{\rho},\boldsymbol{\tau}) = \sum_{j=1}^{p} \mu_j\rho_j\tau_j$.

It is quite straightforward to show that the resultant of a sequence of perturbations is a single perturbation $\boldsymbol{\pi}$ whose components are respectively the products of corresponding components of the perturbations in the sequence, scaled by a common factor to maintain their sum to 1. In the case of two perturbations and by equations (5.19) and (5.21), a resultant composition $\mathbf{x}$ would be given by,

$$\mathbf{x} = \boldsymbol{\mu}\circ\boldsymbol{\rho}\circ\boldsymbol{\tau} = \boldsymbol{\mu}\circ(\boldsymbol{\rho}\circ\boldsymbol{\tau}) = \boldsymbol{\mu}\circ\boldsymbol{\pi} = \left[\mu_1\pi_1, \mu_2\pi_2, \ldots, \mu_p\pi_p\right] / D(\boldsymbol{\mu},\boldsymbol{\pi}) \quad (5.22)$$

where $\pi_j = \rho_j\tau_j$ and $D(\boldsymbol{\mu},\boldsymbol{\pi}) = \Sigma\mu_j\pi_j$.

Cancelling the denominator in equation (5.22),

$$x_j/x_p = \mu_j\pi_j/\mu_p\pi_p = \mu_j\rho_j\tau_j/\mu_p\rho_p\tau_p$$

It follows that the logratios formed from a sequence of perturbations are additive, for in the case of two,

$$\begin{aligned}\log(x_j/x_p) &= \log(\mu_j/\mu_p) + \log(\pi_j/\pi_p) \\ &= \log(\mu_j/\mu_p) + \log(\rho_j/\rho_p) + \log(\tau_j/\tau_p) \qquad j = 1,2,\ldots,p\text{-}1 \quad (5.23)\end{aligned}$$

provided $x_i, \mu_i, \rho_i, \tau_i > 0$, $i = 1,2,\ldots,p$. Generalizing equation (5.23), if it is assumed that a resultant perturbation is made up of many minor perturbations then, under certain regularity conditions $y_j = \log(\pi_j/\pi_p)$, $\pi_p > 0$, $j = 1,2,\ldots p\text{-}1$, will jointly follow a multivariate normal distribution.

Let $\mathbf{x}_i$ be the i-th row of (n×p) $\mathbf{X}$. Assume that $\mathbf{x}_i$ has been perturbed from a primeval perfect mixture $\boldsymbol{\mu}_i = \boldsymbol{\lambda}_i\boldsymbol{\beta}$ by a resultant perturbation $\boldsymbol{\pi}_i$ of many independent perturbations. Then,

$$\log(x_{ij}/x_{ip}) = \log(\mu_{ij}/\mu_{ip}) + \log(\pi_{ij}/\pi_{ip}) \qquad (5.24)$$

provided $x_{ij}, \mu_{ij}, \pi_{ij} > 0$, $i = 1,2,...n$, $j = 1,2,...p-1$. Substituting scalars y_{ij}, m_{ij} and z_{ij} for each logratio in equation (5.24),

$$y_{ij} = m_{ij} + z_{ij}$$

These terms are the components of respectively of the vectors $\mathbf{y}_i$, $\mathbf{m}_i$ and $\mathbf{z}_i$ Hence,

$$\mathbf{y}_i = \mathbf{m}_i + \mathbf{z}_i \qquad (5.25)$$

and this takes the familiar form,

$$\text{Response} = \text{Signal} + \text{Noise}$$

In the light of the remarks made above, it may be anticipated that $\mathbf{z}_i \sim N_{p-1}(\mathbf{0}, \boldsymbol{\Sigma})$, $i = 1,2,...,n$. The n row vectors $\mathbf{y}_i$ of equation (5.25) constitute an $(n \times p)$ array $\mathbf{Y}$ given by

$$\mathbf{Y} = \mathbf{M} + \mathbf{Z} \qquad (5.26)$$

all of these being $n \times (p-1)$ matrices. If the stochastic model is correct, $\mathbf{Z}$ should resemble a multivariate random sample from the $N_{p-1}(\mathbf{0}, \boldsymbol{\Sigma})$ distribution.

It should be evident at once that, since $\boldsymbol{\Lambda}$ and $\boldsymbol{\beta}$ are unknown, there are no observations available on $\log(\pi_{ij}/\pi_{ip})$ by equation (5.24), and therefore no observations on z_{ij}. If it is further assumed that $\mathbf{X}'$ is a good approximation to $\boldsymbol{\Lambda\beta}$, then μ_{ij} may be replaced by x'_{ij} for all i,j in equation (5.24), and so $z_{ij} = \log(\pi_{ij}/\pi_{ip})$ becomes the (i,j)th logratio residual.

Just as it is always sound practice to examine the distribution of the residuals in a regression analysis, it is proposed here that the distribution of logratio residuals, as described above, may be employed to assess the plausibility of an endmember analysis.

If a random vector follows a multivariate normal distribution, then the scalar formed by any linear combination of its components must follow a univariate normal distribution. The strict application of this condition is too severe for geochemical compositional data. There are frequently zeros, repeated values and outliers among the observations for any one variable. Any variable which is not part of a mixing process may not be expected to have normally distributed logratio residuals. If just one marginal distribution is non-normal, then the joint distribution of the whole collection is not multivariate normal. (Consider a linear combination of zeros for all but the abberant component). Therefore, a relatively forgiving statistic was formed as follows. After making the logratio transformation, each of the (p-1) components of the n residual vectors was standardized (to zero mean and unit variance). For each sample, these (p-1) standardized variables were summed and their sum was standardized to form a statistic which will be denoted here by $\mathcal{T}$. The value of $\mathcal{T}$ for the i-th sample would be calculated as follows,

$$\mathcal{Z}_i = \sum_{j=1}^{p-1} (z_{ij} - \bar{z}_j) / s_j$$

$$\mathcal{T}_i = (\mathcal{Z}_i - \overline{\mathcal{Z}}) / s_{\mathcal{Z}} \qquad (5.27)$$

where $s_{\cdot}$ is the appropriate sample standard deviation for samples of size n.

The univariate central limit theorem would in the absence of strong intercorrelations predispose $\mathcal{T}$ to follow a univariate normal distribution which, under the ruling assumption, must otherwise be normal. The normality of $\mathcal{T}$, can of course, be tested by the chi-square goodness of fit test which has the advantage of being immune to outliers.

5.2.1 Mid-Pacific Cobalt-Rich Manganese Crusts

This first illustration of the use of logratio residuals is based on the mixture analysis which was described in Section 4.2. Four endmember compositions were estimated for the 275 samples from a Mid-Pacific subset of the United States Geological Survey world ocean ferromanganese crust database (Lane *et al.* (1986)). The endmembers were identified as (1) Silicate (clay), (2) Cobalt-rich manganese oxide, (3) Biogenic phosphate, (4) Hydrogenous (Authigenic) and their estimated compositions are displayed in Table 4.3.

The (275×22) raw data matrix of this analysis was column transformed (equation (3.51), then a singular value decomposition was performed on the transformed array . Hence, the 22 major and trace elements had equal weights in the analysis. The spaces spanned by k = 2,3,...,10 eigenvectors were taken respectively for choices of S. Setting k = 2,3,...,10 in turn, 9 forms of the (275×22) approximation $\mathbf{X}^{'c}$ were computed whose precision increased with k.

The proportional cumulative sum of the eigenvalues (equation (3.39)) associated with each value of k = 2,3,...,7 is expressed as a percentage variability in the first row at the top of Table 5.4. The chi-square values in the next row of Table 5.4, are the values taken by the goodness of fit statistic with 4 degrees of freedom for the frequency distributions of $\mathcal{T}$. These frequency distributions are depicted as histograms on Figure 5.1. The coefficients of determination (r^2) between the observed and estimated variables for each of these numbers of endmembers are in the body of Table 5.4.

The rule is that any value for $r^2 < 0.5$ (or a Pearson correlation < 0.7) indicates an inadequate estimate. Such values are printed in **boldface** on the table and a ranking scheme keeps keeps them at the bottom of each column.

From the first column of Table 5.4, it can be seen that 9 elements out of 22 are accounted for by k = 2 dimensions. It is noteworthy that, of these 9 elements (or oxides), two (SiO_2 and Al_2O_3) were dominant in a clay endmember, and four (MnO_2, Mo, NiO and Co_3O_4) were dominant in a cobalt-rich manganese oxide endmember (see Table 4.3). Each endmember had been constructed, in the course of a 4 endmember mixture analysis, by the iterative procedures described in Section 3.4.

The inclusion of a third dimension then produced a remarkable increase in the values of r^2 for CaO, P_2O_5 and CO_2, which can be read from the first and second columns of Table 5.4. Further, the iterative algorithm had allocated these 3 oxides together with Sr, almost exclusively to the 3rd endmember which was identified as 'biogenic phosphate' (see Table 4.3).

According to the rule, the addition of a 4th dimension did not account for any more variables. The r^2 value for Ce, the largest contender, rose from 0.38 to 0.47 which is still less than 0.5 (see Table 5.4). It did, however, improve the precision of the estimates for the 15 variables accounted for by 3 endmembers. It also led to the construction of a 4th endmember, identified as hydrogenous, which was the principal source of Fe_2O_3 and the trace elements As, Ce, Pb and V.

The 4 estimated endmembers appeared geochemically viable and, as can be seen from Table 5.4, the inclusion of more dimensions would have added one or two elements at a time at the cost of rapidly dwindling parsimony. On these grounds the 4 endmembers of Table 4.3 are believed to provide a satisfactory account of the data.

The construction of the summary statistic $\mathfrak{T}$ required that samples with zero values be excluded so that the logratios were defined. Differing but small numbers of samples were dropped automatically by the algorithm for each value of k that was examined. Figure 5.1 displays 6 histograms. The first at the top left is of the

probabilities expressed as percentages for the standard normal distribution. The remaining 5 are relative frequency histograms, also in percentages, for the frequency distributions of $\mathfrak{T}$. This statistic was constructed from the logratio residuals arising from representations based on k = 2,3,...,6 endmembers as indicated on Figure 5.1. The histogram which most nearly resembles the standard normal is that for 3 endmembers. The chi-square value for this frequency distribution was 8.63 (Table 5.4) which with 4 degrees of freedom is not significant. Since the chi-square values associated with all other values of k are highly significant, it is therefore tempting to conclude that the data results from a mixing process involving just 3 sources.

Apart from the very satisfactory way in which the iterative algorithm located the extreme compositions for the 4 endmember representation already mentioned, the next illustration will show that caution is needed in the interpretation of the distributions of $\mathfrak{T}$. One further effect was observed which is quite important. Although not all shown here, the distributions of $\mathfrak{T}$ were in fact constructed for k = 2,3,...,10. For $k \geq 5$, the kurtosis of the distributions increased montonically and sharply as did the corresponding chi-square values. This effect was due to the increasing precision of the estimates producing disproportionately many logratio residuals near to zero on the standardized scale. Such an effect is an obvious consequence of over-specification, whether it is of the explanatories in a regression analysis or of the endmembers in a mixture analysis.

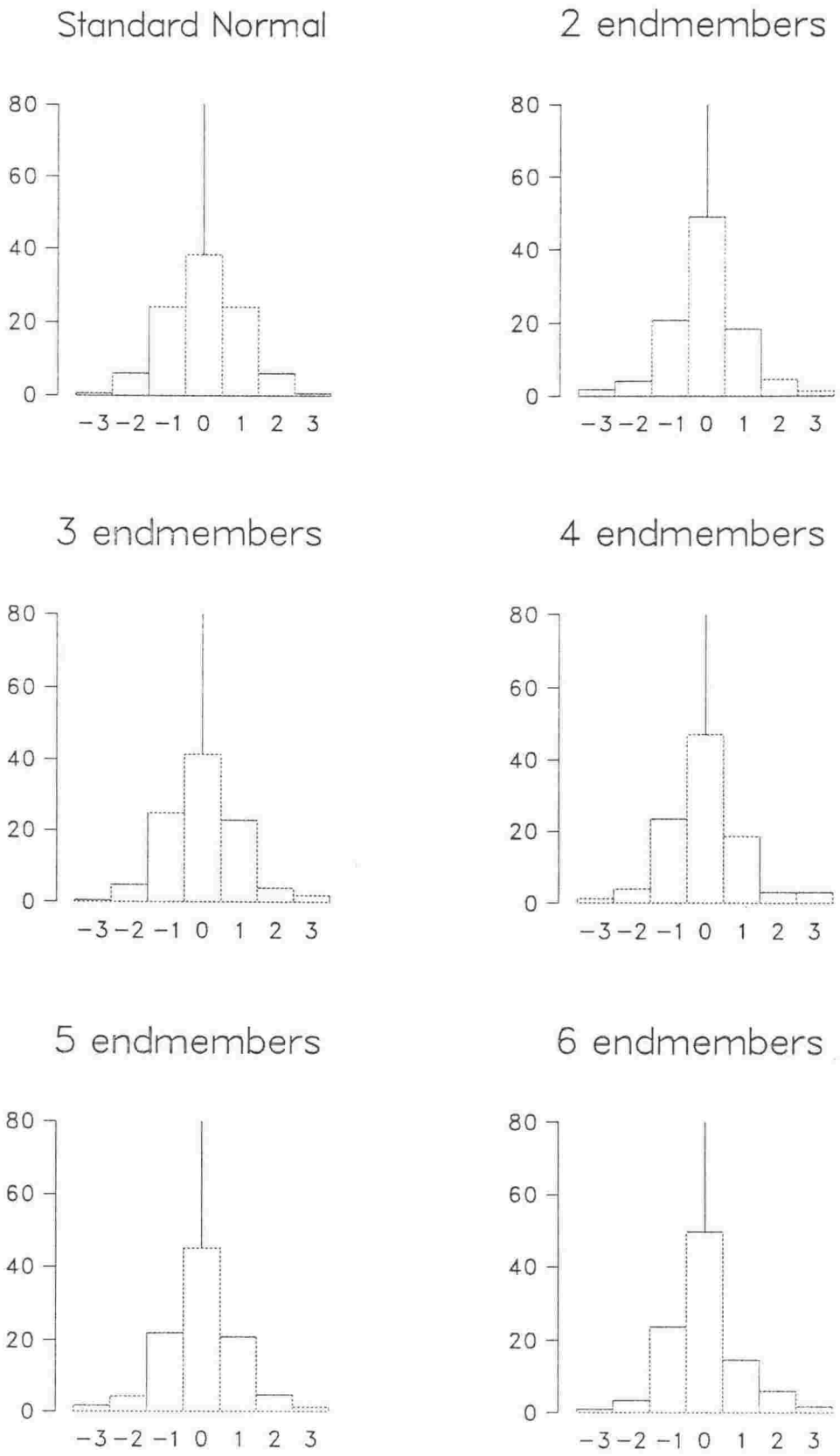
Table 5.4.

Coefficients of Determination (r^2) Between Estimated and Observed Variables for Mid-Pacific Data.

(Note: Values for $r^2 < 0.5$ are printed in boldface)

Endmembers	2	3	4	5	6	7
Variability	94.8%	96.2%	97.2%	97.8%	98.3%	98.6%
Chisquare	20.11	8.63	36.26	12.39	23.8	37.92
SiO_2	0.93	0.93	0.93	0.95	0.97	0.98
MnO_2	0.85	0.87	0.96	0.96	0.98	0.98
Al_2O_3	0.83	0.85	0.86	0.90	0.94	0.94
Mo	0.79	0.80	0.80	0.82	0.83	0.89
Sr	0.65	0.67	0.78	0.79	0.80	0.80
V	0.54	0.62	0.69	0.75	0.76	0.80
NiO	0.51	0.62	0.82	0.84	0.85	0.86
Pb	0.52	0.68	0.71	0.73	0.90	0.90
Co_3O_4	0.50	0.52	0.75	0.76	0.78	0.93
CaO	**0.02**	0.73	0.96	0.96	0.97	0.98
Fe_2O_3	**0.05**	0.72	0.82	0.82	0.93	0.94
P_2O_5	**0.02**	0.64	0.91	0.91	0.91	0.92
As	**0.33**	0.59	0.64	0.87	0.88	0.89
CO_2	**0.00**	0.55	0.72	0.73	0.73	0.77
K_2O	**0.47**	0.50	0.62	0.62	0.72	0.75
Ce	**0.23**	**0.38**	**0.47**	0.75	0.94	0.95
TiO_2	**0.27**	**0.44**	**0.44**	0.53	0.53	0.69
Zn	**0.21**	**0.27**	**0.37**	**0.47**	0.58	0.61
CuO	**0.05**	**0.06**	**0.06**	**0.33**	**0.37**	0.62
MgO	**0.11**	**0.13**	**0.25**	**0.36**	**0.37**	**0.37**
Na_2O	**0.00**	**0.00**	**0.22**	**0.26**	**0.38**	**0.39**
H_2O	**0.01**	**0.10**	**0.11**	**0.12**	**0.14**	**0.19**

Figure 5.1. Histograms of the standardized sums of logratioed residuals for Mid-Pacific data.



5.1.2 Nazca Plate Surface Sediments

The second illustration is based on an analysis of 425 surface sediments from the Nazca plate which have been made available on microfiche by Dymond (1981). For these data, Dymond (*ibid*) tabulated *a priori* the elemental ratio coefficients of five components (endmembers) he used in a normative analysis. These were identified as (1) Detrital, (2) Hydrothermal, (3) Biogenic, (4) Hydrogenous (Authigenic), (5) Dissolution residue (see Sections 2.3 and 5.1) .

The data were analysed in exactly the same way as the Mid-Pacific data above. But first a five endmember representation was constructed to be compared with Dymond's table of elemental ratio coefficients. It was found (Section 5.1) that the first three iteratively constructed estimates were very close to Dymond's components (1), (2) and (3). However the 4th and 5th constructs were similar to components (4) and (5) only in possessing the same dominant elements, and not in the actual magnitudes of those elements (*cf.* Leinen and Pisias (1984)). By varying the initializing components of the iterative routine, the first three estimates were found to be stable while the 4th and 5th were not. So it was with this information at hand that the logratio residuals were examined.

Table 5.5 and Figure 5.2 are to be interpreted in the same way as Table 5.4 and Figure 5.1 respectively. From Table 5.5, it will be seen that there were only 8 variables present and 6 of these were accounted for by 2 endmembers. The associated chi-square value (Table 5.5) was 71.14 which with 4 degrees of freedom refuted the underlying assumptions. The poor fit is evident in the asymmetry of the histogram at the top right of Figure 5.2. If a mixing process did account for the data, then it would seem possible that a 2-endmember representation forced the transfer of part of the systematic effect into the residuals. The inclusion of a 3rd endmember lifted r^2 for Ni from 0.24 to 0.88 and dropped the chi-square value from 71.14 to 31.51, the minimum (Table 5.5). The

corresponding histogram (Figure 5.2) adopted a symmetric shape but its kurtosis also denied the normal hypothesis. However, regression analysts might argue that its symmetry is the important development because an absence of symmetry tends to deny randomness in the errors.

Increasing the number of endmembers accounted for Ba when $k \geq 4$ but, from the appearance of the histograms, resulted in unacceptable kurtosis. A judgement supported by the gross chi-square values. Just as for the distributions formed by increasing the values of k in the study of the Mid-Pacific data, this effect was due to the disproportionately many logratio residuals near to zero on the standardized scale.

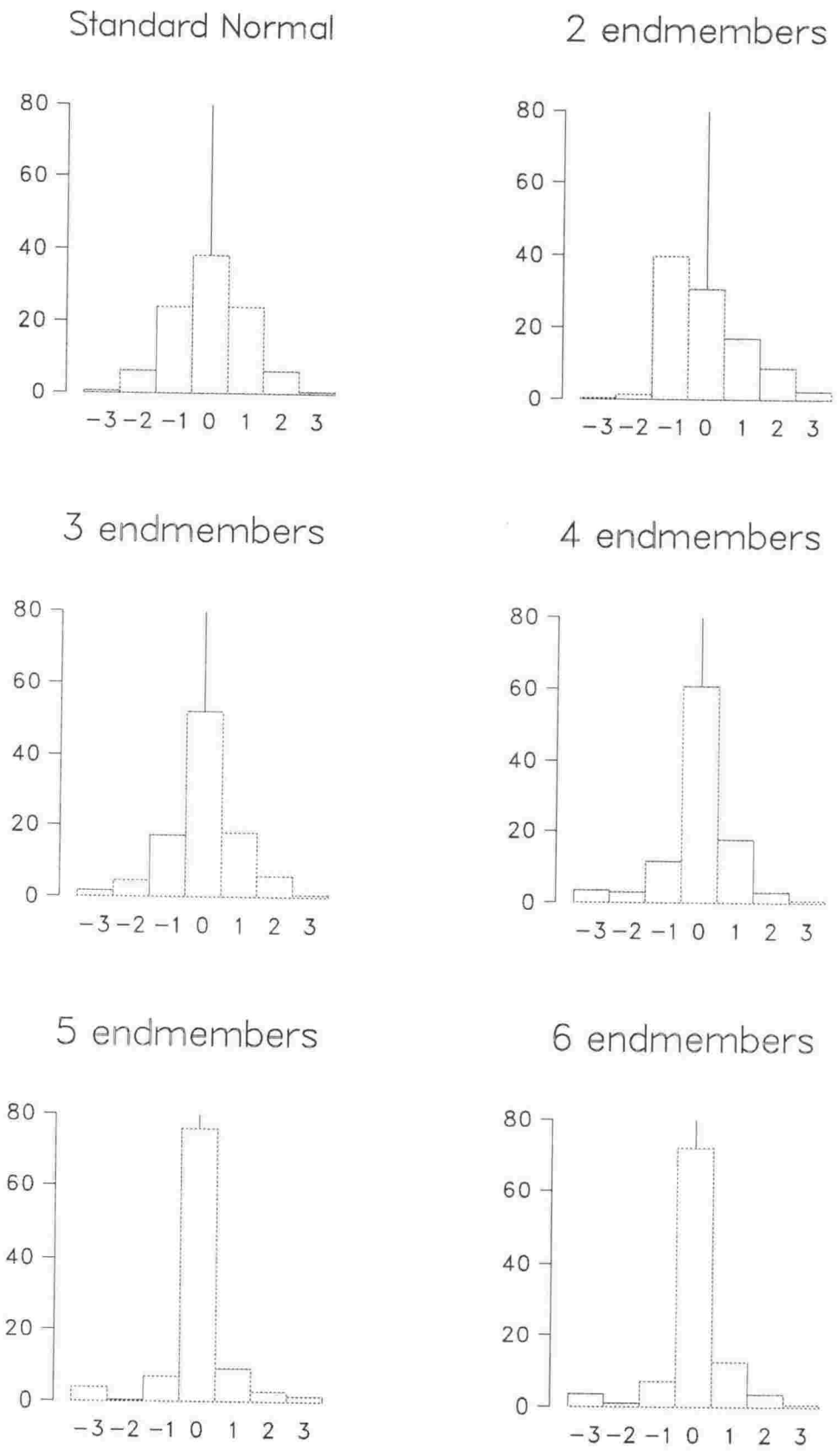
On the basis of the evidence presented here, the normal model is not sustainable. However, if a mixing process were responsible for the data, then 2 endmembers would be too few and 3 the maximum for 'best behaved' residuals.

Table 5.5

Coefficients of determination (r^2) between estimated and observed variables for Nazca Plate data
(Note: Values for $r^2 < 0.5$ are printed in boldface).

Endmembers	2	3	4	5	6	7
Variability	92.8%	96.0%	98.4%	99.2%	99.5%	99.8%
Chisquare	71.14	31.51	109.45	243.47	194.46	193.09
Fe	0.91	0.92	0.99	0.99	0.99	0.99
Si	0.89	0.90	0.99	1.00	1.00	1.00
Mn	0.88	0.89	0.89	0.90	0.91	0.92
Cu	0.70	0.85	0.88	0.91	0.92	0.99
Al	0.65	0.84	0.99	1.00	1.00	1.00
Zn	0.50	0.51	0.53	0.56	0.98	0.98
Ni	**0.24**	0.88	0.90	0.99	0.99	0.99
Ba	**0.03**	**0.35**	0.69	0.95	0.97	1.00

Figure 5.2. Histograms of the standardized sums of logratioed residuals for Nazca Plate data.



In conclusion, the choice of of the number of endmembers in a mixture analysis is an estimate of the true number, and that should be tested. Relying in particular on the coefficient of determination to measure the precision of the estimates for all or most of the variables is to run the risk of over-specification. That is, the identification of endmembers that do not exist and the inclusion of elements into the solution that were not part of the natural mixing process. This is a direct consequence of reckoning errors into the systematic part of the solution.

There are plausible theoretical grounds for anticipating normal distributions among the logratio residuals of those variables that do belong to a mixing process and for which the estimated mixtures are close to the true mixtures. The testing of the logratio residuals for each variable separately is a possibility, although the rejection of the normal distribution hypothesis may not necessarily imperil a mixture hypothesis. A single summary statistic can indicate directly when the systematic part of the estimated model has been included in the errors, and when over-specification seems likely. This statistic could possibly be improved by including in it only those variables that are currently satisfactorily estimated.

REFERENCES

Anderson, T.W. (1958) *An Introduction to Multivariate Statistical Analysis.* John Wiley and Sons, New York, 374 pp.

Anderson, T.W. (1963) *Asymptotic theory for principal-component analysis.* Annals of Mathematical Statistics, Vol. 34, pp. 122-148.

Aitchison, J. (1986) *The Statistical Analysis of Compositional Data.* Chapman and Hall, London and New York, 416 pp.

Bazaraa, M.S. and Shetty, C.M. (1979) *Nonlinear Programming.* John Wiley and Sons, New York, 560 pp.

Cattell, R.B. (1952) *Factor Analysis.* Harper and Bros, New York, 462 pp.

Chen, J.C. and Owen, R.M. (1989) *The hydrothermal component in ferromanganese nodules from the southeast Pacific Ocean.* Geochimica et Cosmochimica Acta, Vol. 53, 1299-1305.

Churchman, G.J. Hunt, J.L., Glasby, G.P., Renner, R.M. and Griffiths, G.A. (1988) *Input of River-derived Sediment to the New Zealand Continental Shelf: II Mineralogy and Composition.* Estuarine, Coastal and Shelf Science, Vol. 27, pp.397-411.

Clarke, T.L. (1978) *An Oblique Factor Analysis Solution for the Analysis of Mixtures.* Mathematical Geology, Vol. 10, No. 2, pp. 225-241.

De Carlo, E.H., McMurtry, G.M. and Kim, H.K. (1987) *Geochemistry of ferromanganese crusts from the Hawaiian Archipelago-I. Northern survey areas.* Deep-Sea Research, Vol. 34, No. 3, pp. 441-467.

Dean, W.E., Gardner, J.V. and Parduhn, N.L. (1989) *Influence of Shimada Seamount on sediment composition in the eastern tropical North Pacific.* Geochimica et Cosmochimica Acta, Vol. 53, pp. 1523-1536.

Dwyer, P.S. (1939) *The contribution of an orthogonal multiple factor solution to multiple correlation.* Psychometrika, Vol. 4, pp. 163-171.

Dymond, J. (1981) *Geochemistry of Nazca plate surface sediments : An evaluation of hydrothermal, biogenic, detrital, and hydrogenous sources.* Geological Society of America Memoir 154, pp. 133-173.

Dymond, J., Lyle, M., Finney, B., Piper, D.Z., Murphy, K., Conard, R. and Pisias, N. (1984) *Ferromanganese nodules from MANOP sites H, S, and R - Control of mineralogical and chemical composition by multiple accretionary processes.* Geochimica et Cosmochimica Acta, Vol. 48, Pergamon Press, pp. 931-949.

Everitt, B.S. (1984) *An Introduction to Latent Variable Models.* Chapman and Hall, London and New York, 107 pp.

Full, W.E. and Ehrlich, R. (1986) *Comment on "An objective technique for determining end-member compositions and for partitioning sediments according to their sources".* Geochimica et Cosmochimica Acta, Vol. 50, Pergamon Journals Ltd., p. 1303.

Full, W.E., Ehrlich, R. and Bezdek, J.C. (1982) *Fuzzy QMODEL - A New Approach for Linear Unmixing.* Mathematical Geology, Vol. 14, No. 3, pp. 259-270.

Full, W.E., Ehrlich, R. and Klovan, J.E. (1981) *EXTENDED QMODEL - Objective Definition of External End Members in the Analysis of Mixtures.* Mathematical Geology, Vol. 13, No. 4, pp. 331-344.

Glasby, G.P., Hunt, J.L. and Renner, R.M. (1985) *Trace Element Analyses of Marine Sediments from the Southwest Pacific.* New Zealand Soil Bureau Scientific Report 53, Department of Scientific and Industrial Research, New Zealand, 62 pages.

Glasby, G.P., Stoffers, P., Walter, P., Davis, K.R. and Renner, R.M. (1988) *Heavy-metal pollution in Manukau and Waitemata Harbours, New Zealand.* New Zealand Journal of Marine and Freshwater Research, Vol. 22, pp.595-611.

Hadley, G. (1962) *Linear Programming.* Addison-Wesley Publishing Company Inc., Reading, Massachusetts, 520 pp.

Harman, H.H. (1967) *Modern Factor Analysis* (2nd edition). University of Chicago Press, Chicago, 474 pp.

Heath, G.R. and Dymond, J. (1977) *Genesis and transformation of metalliferous sediments from the East Pacific Rise, Bauer Deep, and Central Basin, northwest Nazca plate.* Geological Society of America Bulletin, Vol. 88, pp. 723-733.

Hotelling, H. (1933) *Analysis of a complex of statistical variables into principal components.* Journal of Educational Psychology, Vol. 24, pp. 417-441, 498-520.

Imbrie, J. (1963) *Factor and Vector Analysis Programs for Analyzing Geologic Data.* Technical Report No. 6 of ONR Task No. 389-135, Contract Nonr 1228(26), Office of Naval Research, Geography Branch, 83 pp.

Imbrie, J. and Purdy, E.G. (1962) *Classification of modern Bahamian carbonate sediments.* W.E. Ham (*Editor*), American Association of Petroleum Geologists, Mem. 1, pp. 253-272.

Imbrie, J. and Van Andel, T.H. (1964) *Vector Analysis of Heavy-Mineral Data.* Geological Society of America Bulletin, Vol. 75, pp. 1131-1156.

Jöreskog, K.G., Klovan, J.E. and Reyment, R.A. (1976) *Methods in Geomathematics 1 Geological Factor Analysis.* Elsevier Scientific Publishing Company, Amsterdam, 178 pp.

Johnson, A.J. and Wichern, D.W. (1988) *Applied Multivariate Statistical Analysis* 2nd Ed. Prentice Hall, Englewood Cliffs, New Jersey, 607 pp.

Kaiser, H.F. (1958) *The varimax criterion for analytic rotation in factor analysis.* Psychometrika, Vol. 23, pp. 187-200.

Kempthorne, O. (1952) *The design and analysis of experiments.* John Wiley and Sons, New York, 631 pp.

Klovan, J.E. (1966) *The Use of Factor Analysis in Determining Depositional Environments From Grain-size Distributions.* Journal of Sedimentary Petrology, Vol. 36, No. 1, pp. 115-125.

Klovan, J.E. and Imbrie, J. (1971) *An Algorithm and FORTRAN-IV Program for Large-Scale Q-Mode Factor Analysis and Calculation of Factor Scores.* Mathematical Geology, Vol. 3, No. 1, pp. 61-76.

Klovan, J.E. and Miesch, A.T. (1976) *Extended CABFAC and QMODEL computer programs for Q-mode factor analysis of compositional data.* Computers and Geosciences, Vol. 1, No. 3, Pergamon Press, pp. 161-178.

Kunzendorf, H., Gwozdz, R., Glasby, G.P., Stoffers, P. and Renner, R.M. (1989) *The distribution of rare earth elements in manganese micronodules and sediments from the equatorial and southwest Pacific.* Applied Geochemistry, Vol. 4, pp.183-193.

Lawley, D.N. and Maxwell, A.E. (1971) *Factor Analysis as a Statistical Method.* Butterworths, London, 153 pp.

Lane, C.M., Manheim, F.T., Hathaway, J.C. and Ling, T.H. (1986) *Station Maps of The World Ocean - Ferromanganese - Crust data base.* Department of The Interior U.S. Geological Survey. Miscellaneous Field Studies Map, Map MF-1869. U.S. Geological Survey.

Leinen, M. (1987) *The origin of paleochemical signature in North Pacific pelagic clays : Partitioning experiments.* Geochimica et Cosmochimica Acta, Vol. 51, Pergamon Journals Ltd., pp. 305-319.

Leinen, M. and Pisias, N. (1984) *An objective technique for determining end-member compositions and for partitioning sediments according to their sources.* Geochimica et Cosmochimica Acta, Vol. 48, Pergamon Press, pp. 47-62.

Miesch, A.T. (1976a) *Q-mode Factor Analysis of Compositional Data.* Computers and Geosciences, Vol. 1, No. 3, Pergamon Press, pp. 147-159.

Miesch, A.T. (1976b) *Q-mode Factor Analysis of Geochemical and Petrologic Data Matrices with Constant Row Sums.* Statistical Studies in Field Geochemistry, Geological Survey Professional Paper 574-G, U.S. Government Printing Office, Washington, 47 pp.

Miesch, A.T. (1980) *Scaling Variables and Interpretation of Eigenvalues in Principal Component Analysis of Geologic Data.* Mathematical Geology, Vol. 12, No. 6, pp. 523-538.

Morrison, D.F. (1976) *Multivariate Statistical Methods* 2nd Ed. McGraw-Hill Kogakusha Ltd, International Student Edition, Tokyo, 415 pp.

Narula, S.C. and Wellington, J.F. (1977) *Multiple Linear Regression with Minimum Sum of Absolute Errors.* Applied Statistics, Journal of The Royal Statistical Society (Series C), Vol. 26, pp. 106-111.

Nath, B.N., Rao, V.P. and Becker, K.P. (1989) *Geochemical evidence of terrigenous influence in deep-sea sediments up to 8°S in the Central Indian Basin.* Marine Geology, Vol. 87, pp. 301-313.

Owen, R.M. (1987) *Geostatistical Problems in Marine Placer Exploration.* P.G. Teleki, M.R. Dobson, J.R. Moore and V. von Stackelberg, *Marine Minerals.* D. Reidel Publishing Company, Dordrecht, pp. 533-540.

Rao, C.R. (1973) *Linear Statistical Inference and Its Applications,* 2nd edition. John Wiley and Sons, New York, 625 pp.

Renner, R.M. (1982) *Sediment Analysis : A Q-mode Approach.* The New Zealand Statistician, Vol. 17, No. 2, pp. 12-17.

Renner, R.M. (1988) *On the resolution of compositional datasets into convex combinations of extreme vectors.* Technical Report No. 88/02, Institute of Statistics and Operations Research, Victoria University of Wellington, Wellington, New Zealand, 40 pp.

Renner, R.M. (1989) *Comment on " Bediasite source materials: A solution to an endmember mixing problem exploiting closed data" by A. Woronow and K.M. Love.* Geochimica et Cosmochimica Acta, Vol. 53, pp. 1669-1670.

Renner, R.M., Glasby, G.P., Manheim, F.J. and Lane-Bostwick, C.M. (1990) *A partitioning process for geochemical datsets.* Proceedings of the Colloquium on "Statistical Applications in The Earth Sciences". Geological Survey of Canada, Ottawa. (in press).

SAS Institute Inc.(1985) *SAS User's Guide: Basics, Version 5 Edition.* Cary, NC: SAS Institute Inc., 1290 pp.

Seber, G.A.F. (1984) *Multivariate Observations.* John Wiley and Sons, New York, 686 pp.

Thurstone, L.L. (1947) *Multiple Factor Analysis.* University of Chicago Press, Chicago, 535 pp.

Tryon, R.C. and Bailey, D.C. (1970) *Cluster Analysis.* McGraw-Hill.

Walter, P. and Stoffers, P. (1985) *Chemical characteristics of metalliferous sediments from eight areas on the Galapagos Rift and East Pacific Rise between 2°N and 42°S.* Marine Geology, Vol. 65, pp. 271-287.

APPENDIX

```
C*************************** SVD0 FORTRAN ****************************
C
C   NOTES:  This program reads the arguments for the singular value
C           decomposition program SVD1 FORTRAN.
C
      CHARACTER*1 REPLY
      COMMON /LABEL1/NVAR,NEND,ISCALE,IUNIT,IPE,ICONT,ICON
C*********************************************************************
C                              MAIN program                          *
C*********************************************************************
C                                   Read required constants from terminal
      CALL LOAD
C                                        Write required constants to disk
      CALL WRITE
C
      STOP
C****                          End of MAIN program                  *****
      END
C
C*********************************************************************
C*                            Procedure LOAD  (load raw data)        *
C*********************************************************************
      SUBROUTINE LOAD
      CHARACTER*1 REPLY
      COMMON /LABEL1/NVAR,NEND,ISCALE,IUNIT,IPE,ICONT,ICON
    1 CONTINUE
      WRITE(6,2)
    2 FORMAT(2(/),5X,'Enter the number of variables (at most 40)')
      READ(5,*) NVAR
      WRITE(6,3)
    3 FORMAT(/,5X,'Enter the number of end-members (at most 10)')
      READ(5,*) NEND
    4 CONTINUE
      WRITE(6,5)
    5 FORMAT(/,5X,'Key in 0 for no scaling of variables,'/
     @         5X,'       1 for division by observed maximum,'/
     @         5X,'       2 for fractional ranges')
      READ(5,*) ISCALE
      IF (ISCALE.NE.0.AND.ISCALE.NE.1.AND.ISCALE.NE.2) GO TO 4
    6 CONTINUE
      WRITE(6,7)
    7 FORMAT(/,5X,'Row normalize (objects into unit vectors) ?  y/n')
      READ(6,8) REPLY
    8 FORMAT(A1)
      IF (REPLY.NE.'Y'.AND.REPLY.NE.'y'.AND.
     @    REPLY.NE.'N'.AND.REPLY.NE.'n') GO TO 6
      IUNIT = 1
      IF (REPLY.EQ.'N'.OR.REPLY.EQ.'n') IUNIT = 0
    9 CONTINUE
      WRITE(6,10)
   10 FORMAT(/,5X,'Does the data sum to 100% ?  y/n')
      READ(6,11) REPLY
```

```
   11 FORMAT(A1)
      IF (REPLY.NE.'Y'.AND.REPLY.NE.'y'.AND.
     @    REPLY.NE.'N'.AND.REPLY.NE.'n') GO TO 9
      IPE = 1
      IF (REPLY.EQ.'N'.OR.REPLY.EQ.'n') IPE = 0
   12 CONTINUE
      WRITE(6,13)
   13 FORMAT(/,5X,'Output loadings, ranks, angles and estimates?  y/n')
      READ(6,14) REPLY
   14 FORMAT(A1)
      IF (REPLY.NE.'Y'.AND.REPLY.NE.'y'.AND.
     @    REPLY.NE.'N'.AND.REPLY.NE.'n') GO TO 12
      ICONT = 1
      IF (REPLY.EQ.'N'.OR.REPLY.EQ.'n') ICONT = 0
   15 CONTINUE
      WRITE(6,16)
   16 FORMAT(/,5X,'Output non-negative estimates only (CONLSQ)?  y/n')
      READ(6,17) REPLY
   17 FORMAT(A1)
      IF (REPLY.NE.'Y'.AND.REPLY.NE.'y'.AND.
     @    REPLY.NE.'N'.AND.REPLY.NE.'n') GO TO 15
      ICON = 1
      IF (REPLY.EQ.'N'.OR.REPLY.EQ.'n') ICON = 0
      WRITE(6,18)
   18 FORMAT(/,5X,'All correct ?   y/n ')
      READ(5,19) REPLY
   19 FORMAT(A1)
      IF (REPLY.NE.'Y'.AND.REPLY.NE.'y') GO TO 1
      RETURN
C****                      End of procedure LOAD                      *****
      END
C
C*************************************************************************
C*                          Procedure WRITE (Write constants to disk)   *
C*************************************************************************
      SUBROUTINE WRITE
      COMMON /LABEL1/NVAR,NEND,ISCALE,IUNIT,IPE,ICONT,ICON
      WRITE(11,1) NVAR,NEND,ISCALE,IUNIT,IPE,ICONT,ICON
    1 FORMAT(1X,7I5)
      RETURN
C****                      End of procedure WRITE                     *****
      END
```

```
C*************************** SVD1 FORTRAN ****************************
C
C NOTES: (1) This is a singular value decomposition algorithm.
C        (2) Input raw data must be in freefield.
C        (3) Summary information is written to disk ddname 13,
C            estimated A ddname 11, recovered A ddname 17, the
C            loading matrix ddname 15, the eigenvectors (columns)
C            ddname 19.
C
      REAL*16 A(800,40),AE(800,40),D(800,10),RLNGTH(800),
     @        RANGE(40),C(40,40),V(40,40),SSQS,ZERO,ONE,TEST,
     @        WE(800,40)
C
      COMMON  /LABEL1/ZERO,ONE,TEST,NOBJ,NVAR,NEND,MAL,ISCALE,IUNIT,IPE
     @               ,ICONT,ICON,IFAULT
     @        /LABEL2/A
     @        /LABEL3/RANGE
     @        /LABEL4/C,V
     @        /LABEL5/D
     @        /LABEL6/RLNGTH
C****                                                  Define constants
      ZERO = 0.0Q+00
      ONE  = 1.0Q+00
      TEST = 1.0Q-10
C*********************************************************************
C                              MAIN program                          *
C*********************************************************************
C                                                  Load raw data from disk
      CALL LOAD
C                                      Rescale columns into fractional ranges
      CALL SCALE
C                                              Rescale rows into unit vectors
      CALL UNIT
C                                 Form the matrix product C = A transpose by A
      CALL SYMM
C                                      Eigenvalues and eigenvectors of matrix C
      CALL EIGEN
C
      IF (ICONT.EQ.1) THEN
C                            Components on orthogonal axes (Loading matrix)
         CALL COMPTS
C
         CALL EST
C
      END IF
      STOP
C****                          End of MAIN program                   *****
      END
C
C*********************************************************************
C*                            Procedure LOAD  (load raw data)         *
C*********************************************************************
```

```
      SUBROUTINE LOAD
      REAL*16 A(800,40),ZERO,ONE,TEST
      COMMON /LABEL1/ZERO,ONE,TEST,NOBJ,NVAR,NEND,MAL,ISCALE,IUNIT,IPE
     @              ,ICONT,ICON,IFAULT
     @       /LABEL2/A
C****                  Program reads numbers of elements, end-members etc
      READ(12,*) NVAR,NEND,ISCALE,IUNIT,IPE,ICONT,ICON
C****                      Program reads and counts rows of input matrix
      NOBJ = 1
    1 CONTINUE
C****                                  Input matrix must be in freefield
      READ (10,*, END = 2) (A(NOBJ,J),J = 1,NVAR)
      NOBJ = NOBJ + 1
      GO TO 1
    2 CONTINUE
C****                  Program counts one more than true number of records
      NOBJ = NOBJ - 1
      RETURN
C****                     End of procedure LOAD                        *****
      END
C*************************************************************************
C                            Procedure SCALE  (scale columns)             *
C*************************************************************************
      SUBROUTINE SCALE
      REAL*16 A(800,40),RANGE(40),RMAX,RMIN,ZERO,ONE,TEST
      COMMON /LABEL1/ZERO,ONE,TEST,M,N,KE,MAL,ISCALE,IUNIT,IPE,ICONT,
     @              ICON,IFAULT
     @       /LABEL2/A
     @       /LABEL3/RANGE
C*** Column ranges are initialized to one in case there is no scaling
      DO 1 J=1,N
         RANGE(J) = ONE
    1 CONTINUE
      IF (ISCALE.GT.0) THEN
         CALL TITLE
         DO 6 J=1,N
            M1=1
            DO 2 K=2,M
               IF (A(K,J).GT.A(M1,J)) M1 = K
    2       CONTINUE
            RMAX = A(M1,J)
            M1=1
            DO 3 K=2,M
               IF (A(K,J).LT.A(M1,J)) M1 = K
    3       CONTINUE
            RMIN = A(M1,J)
            IF (ISCALE.EQ.1) RANGE(J) = RMAX
            IF (ISCALE.EQ.2) RANGE(J) = RMAX - RMIN
            DO 4 I=1,M
               IF (ISCALE.EQ.1) A(I,J)= A(I,J)/RMAX
               IF (ISCALE.EQ.2) A(I,J)=(A(I,J)-RMIN)/RANGE(J)
    4       CONTINUE
```

```
            WRITE(13,5) J,RMAX,RMIN
     5      FORMAT(1X/5X,'VARIABLE',I3,5X,'MAXIMUM =',F13.6,
      @             5X,'MINIMUM =',F13.6)
     6    CONTINUE
       END IF
       RETURN
C****                        End of procedure SCALE                    ****
       END
C*************************************************************************
C                          Procedure UNIT  (unit vectors)                 *
C*************************************************************************
       SUBROUTINE UNIT
       REAL*16 A(800,40),RLNGTH(800),R,SSQ,ZERO,ONE,TEST
       COMMON /LABEL1/ZERO,ONE,TEST,M,N,KE,MAL,ISCALE,IUNIT,IPE,ICONT,
      @              ICON,IFAULT
      @       /LABEL2/A
      @       /LABEL6/RLNGTH
C**** Row lengths stored as ONE in case of no row unitizing
       DO 1 I=1,M
          RLNGTH(I) = ONE
     1 CONTINUE
C****                        If IUNIT = 1, the rows of A become unit vectors
       IF (IUNIT.EQ.1) THEN
           DO 4 I=1,M
              SSQ=ZERO
              DO 2 J=1,N
                 SSQ = SSQ + A(I,J)**2
     2        CONTINUE
                           R = QSQRT(SSQ)
                 RLNGTH(I) = R
              DO 3 J=1,N
                 A(I,J) = A(I,J)/R
     3        CONTINUE
     4     CONTINUE
       END IF
       RETURN
C****                       End of procedure UNIT                      ****
       END
C*************************************************************************
C                         Procedure SYMM (A transpose * A)                *
C*************************************************************************
       SUBROUTINE SYMM
       REAL*16 A(800,40),C(40,40),V(40,40),ZERO,ONE,TEST
       COMMON /LABEL1/ZERO,ONE,TEST,M,N,KE,MAL,ISCALE,IUNIT,IPE,ICONT,
      @              ICON,IFAULT
      @       /LABEL2/A
      @       /LABEL4/C,V
       DO 3 I=1,N
          DO 2 J=1,N
             C(I,J)=ZERO
             DO 1 K=1,M
                C(I,J)=C(I,J)+A(K,I)*A(K,J)
```

```
     1       CONTINUE
     2    CONTINUE
     3 CONTINUE
       RETURN
C****                        End of procedure SYMM                          ****
       END
C*************************************************************************
C                              Procedure EIGEN                              *
C*************************************************************************
       SUBROUTINE EIGEN
       REAL*16 C(40,40),V(40,40),EVAL(40),X1(40),X2(40),ZERO,ONE,
      1        SUMM,TEST,SUM1,SUM2,VLNGTH,DIAG,EIGVAL,CUMI
       COMMON /LABEL1/ZERO,ONE,TEST,NOBJ,N,KE,MAL,ISCALE,IUNIT,IPE,
      @               ICONT,ICON,IFAULT
      @       /LABEL4/C,V
C
       CALL TITLE
C****                          Sum the diagonal elements of symmetric matrix C
       DIAG = ZERO
       DO 2 I=1,N
          DIAG = DIAG + C(I,I)
          DO 1 J=1,N
             V(I,J) = C(I,J)
     1    CONTINUE
     2 CONTINUE
C****                           Form matrix C squared, to separate eigenvalues
       DO 5 I=1,N
          DO 4 J=1,N
             C(I,J) = ZERO
             DO 3 K=1,N
                C(I,J) = C(I,J) + V(I,K)*V(K,J)
     3       CONTINUE
     4    CONTINUE
     5 CONTINUE
C***                          M = eigenvalue number, starting with the largest
C      SUMM = ZERO
          M  = 1
     6 CONTINUE
          IT = 0
       DO 7 I=1,N
          X1(I) = ONE
     7 CONTINUE
     8 IT = IT + 1
       DO 9 I=1,N
          X2(I) = ZERO
          DO 9 J=1,N
              X2(I) = X2(I) + C(I,J)*X1(J)
     9 CONTINUE
       EIGVAL  = X2(1)
       SUM1    = ZERO
       DO 10 I=1,N
          X2(I) = X2(I)/EIGVAL
```

```
            SUM1  = SUM1 + QABS(X2(I) - X1(I))
            X1(I) = X2(I)
     10  CONTINUE
C
         IF (IT.LT.1000) THEN
            GO TO 14
         ELSE
            IF (SUM1.GT.SUM2) THEN
               WRITE (6,13)
     13        FORMAT (2(/),5X,'Iteration diverging, processing stopped')
               MAL = 1
               GO TO 21
            END IF
         END IF
     14  CONTINUE
         SUM2 = SUM1
         IF (SUM1.GT.TEST) GO TO 8
C*****                                Otherwise, end of iteration for M-th EV
C        EVAL(M) = EIGVAL
         EVAL(M) = QSQRT(EIGVAL)
          SUMM   = SUMM + EVAL(M)
          CUMI   = SUMM/DIAG
     15  SUM1 = ZERO
         DO 16 I=1,N
             SUM1 = SUM1 + X2(I)*X2(I)
     16  CONTINUE
         VLNGTH = QSQRT(SUM1)
         DO 17 I=1,N
            V(I,M) = X2(I)/VLNGTH
     17  CONTINUE
C
         DO 18 I=1,N
            DO 18 J=1,N
               C(I,J) = C(I,J) - V(I,M)*V(J,M)*EIGVAL
     18  CONTINUE
C
         IF ((M.LT.KE).AND.(CUMI.LT.0.9999Q+00)) THEN
C        IF  (M.LT.KE)                           THEN
            M = M + 1
            GO TO 6
         ELSE
            IF (M.LT.KE) THEN
                 KE = M
                 WRITE(13,*) '>>>>>>> Number of endmembers = ',KE
                 WRITE( 6,*) '>>>>>>> Number of endmembers = ',KE
                 WRITE(13,20)
     20          FORMAT(2(/))
            END IF
         END IF
C****                                 End of computation of eigenvectors
     21  CONTINUE
         WRITE(5,24)
```

```
   24 FORMAT(10(/),2X,'    EIGENVALUES',(/))
      WRITE (5,25) (EVAL(I),I=1,KE)
   25 FORMAT(2X,F15.7)
      CUMI = ZERO
      SUMM = ZERO
      DO 30 I=1,KE
         SUMM = SUMM + EVAL(I)
        EVALI = 100*EVAL(I)/DIAG
         CUMI = CUMI + EVALI
        WRITE (13,29) I,EVAL(I),EVALI,CUMI
   29    FORMAT (1X,'EIGENVALUE ',I3,F15.7,F15.2,' %',F15.2,' %')
   30 CONTINUE
      WRITE (13,31) SUMM,DIAG,M
   31 FORMAT (2(/),1X,'SUM OF EIGENVALUES = ',F9.4,5X,
     1        'SUM OF DIAGONAL ELEMENTS =',F10.4,5X,'M =',I4)
C
C  Write to disk the NEND eigenvectors that are associated with the
C  NEND largest eigenvalues in order of magnitude. These vectors form
C  an approximate basis for matrix A as at beginning of this procedure
C
      CALL TITLE
      WRITE (13,36) KE
      DO 34 I=1,N
      WRITE (19,33) (V(I,J),J=1,KE)
      WRITE (13,37) (V(I,J),J=1,KE)
   33 FORMAT (10F8.4)
   34 CONTINUE
      WRITE(6,35) KE
   35 FORMAT(/5X,I2,' Eigenvectors stored and written to disk')
   36 FORMAT('0','THE',I3,' EIGENVECTORS')
   37 FORMAT(1X,10F10.4)
      RETURN
C****                  End of procedure EIGEN                          ****
      END
C*************************************************************************
C                      Procedure COMPTS (Components or loading matrix)   *
C*************************************************************************
      SUBROUTINE COMPTS
      REAL*16 A(800,40),D(800,10),C(40,40),V(40,40),
     @        DJ(800),SSQ(10),SSQI,SSQJ,DMAXJ,DMINJ,ZERO,ONE,TEST
      INTEGER IROW(800,10)
      COMMON /LABEL1/ZERO,ONE,TEST,NOBJ,NVAR,NEND,MAL,ISCALE,IUNIT,IPE,
     @               ICONT,ICON,IFAULT
     @       /LABEL2/A
     @       /LABEL4/C,V
     @       /LABEL5/D
C
C     Compute the loading matrix D(I,J).
C
      CALL TITLE
      DO 3 I=1,NOBJ
         DO 2 J=1,NEND
```

```
            D(I,J) = ZERO
            DO 1 K=1,NVAR
               D(I,J) = D(I,J) + A(I,K)*V(K,J)
    1       CONTINUE
    2    CONTINUE
    3 CONTINUE
C****   Construct ranks of elements in columns of initial loading matrix
      DO 600 J=1,NEND
         DMAXJ   = D(1,J)
      DO 601 I=1,NOBJ
         DJ(I) = D(I,J)
         IF (DJ(I).GT.DMAXJ) DMAXJ = DJ(I)
  601 CONTINUE
      DO 6 K=1,NOBJ
         DMINJ   = DMAXJ

         DO 4 I=1,NOBJ
            IF (DJ(I).LE.DMINJ) THEN
               DMINJ = DJ(I)
               IMINJ = I
            END IF
    4    CONTINUE
         IROW(IMINJ,J) = K
C***                                                    Push out of reach
         DJ(IMINJ)     = 2*DMAXJ
    6 CONTINUE
  600 CONTINUE

C****                        Write loadings to diskfiles, ddnames 13 and 15
      WRITE (13,7) NEND
    7 FORMAT ('0',1X,'OBJECT NUMBER'/
     @           ,5X,'       THE',I3,' COLUMNS OF INITIAL LOADING MATRIX
     @ ')
      DO 10 I=1,NOBJ
         WRITE(13,8) I,(D(I,J),J=1,NEND)
         WRITE(15,9)   (D(I,J),J=1,NEND)
    8    FORMAT(1X,I3,10(1X,F10.4))
    9    FORMAT(10F8.4)
   10 CONTINUE
C***********                  Row-unitized data
      IF (IUNIT.EQ.1) THEN
      DO 12 J=1,NEND
         SSQJ = ZERO
         DO 11 I=1,NOBJ
            SSQJ = SSQJ + D(I,J)*D(I,J)
   11    CONTINUE
         SSQ(J) = SSQJ
   12 CONTINUE
      WRITE(13,13)
   13 FORMAT(/1X,'COLUMN SUMS OF SQUARES OF INITIAL LOADINGS (ROW UNITIZ
     @ED DATA)'/)
         WRITE(13,14) (SSQ(J),J=1,NEND)
```

```
   14    FORMAT(4X,10(1X,F10.4))
      END IF
      CALL TITLE
      WRITE(13,703)
  703 FORMAT(1X,'RANKS OF ELEMENTS IN COLUMNS OF INITIAL LOADING MATRIX'
     @/)
      DO 700 I=1,NOBJ
         WRITE(13,701) I,(IROW(I,J),J=1,NEND)
  701    FORMAT(1X,I4,2X,10I5)
  700 CONTINUE
      RETURN
C****                End of procedure COMPTS                          ****
      END
C*************************************************************************
C       Procedure EST (Estimate matrix A using end-members as a basis)*
C*************************************************************************
      SUBROUTINE EST
      REAL*16 A(800,40),D(800,10),AE(800,40),V(40,40),C(40,40),DJ(800),
     @        EM(40,40),X(40),XE(40),
     @       RLNGTH(800),RANGE(40),SUM,ZERO,ONE,TEST,PI,ANGLE,SUMA,SUMAE
      COMMON /LABEL1/ZERO,ONE,TEST,NOBJ,NVAR,NEND,MAL,ISCALE,IUNIT,IPE
     @              ,ICONT,ICON,IFAULT
     @       /LABEL2/A
     @       /LABEL3/RANGE
     @       /LABEL4/C,V
     @       /LABEL5/D
     @       /LABEL6/RLNGTH
C
C      Compute the estimate of A (AE) using NEND columns of D and
C      NEND columns of V (NEND rows of V transpose)
C      Since matrix multiplication is associative, columns are rescaled
C      before rows.
      CALL TITLE
      CALL TRNSPS(V,EM,NVAR,NEND)
C
C     Call the A estimate AE(I,J).
C
      DO 4 I=1,NOBJ
         SUMA  = ZERO
         SUMAE = ZERO
         DJ(I) = ZERO
         DO 2 J=1,NVAR
              AE(I,J) = ZERO
              DO 1 K=1,NEND
                   AE(I,J) = AE(I,J) + D(I,K)*V(J,K)
    1         CONTINUE
              X(J) =  A(I,J)
              XE(J) = AE(I,J)
    2    CONTINUE
         IF (ICON.EQ.1) THEN
              CALL CONLSQ(EM,X,XE,NVAR,NEND)
         END IF
```

```
          DO 101 J = 1,NVAR
C****                          Constrained LSQ estimate only if IFAULT = 0
               IF (IFAULT.EQ.0) AE(I,J) = XE(J)
               SUMA  = SUMA  +  A(I,J)*A(I,J)
               SUMAE = SUMAE + AE(I,J)*AE(I,J)
  101     CONTINUE
          SUMAE = QSQRT(SUMAE)
          SUMA  = QSQRT(SUMA)
          DO 3 J=1,NVAR
             DJ(I) = DJ(I) + (AE(I,J)/SUMAE)*(A(I,J)/SUMA)
    3     CONTINUE
    4 CONTINUE
C****                           Inverse operations to procedures SCALE and UNIT
      DO 7 I=1,NOBJ
          SUMA  = ZERO
          SUMAE = ZERO
          DO 5 J=1,NVAR
C****                                                  Rescale all matrix elements
            A(I,J) =  A(I,J)*RANGE(J)*RLNGTH(I)
           AE(I,J) = AE(I,J)*RANGE(J)*RLNGTH(I)
             SUMA   =    SUMA  +  A(I,J)
             SUMAE  =    SUMAE + AE(I,J)
    5     CONTINUE
C****                               If IPE = 1, input data summed to 100%
          IF (IPE.EQ.1) THEN
             DO 6 J=1,NVAR
                 A(I,J) =  100*A(I,J)/SUMA
                AE(I,J) = 100*AE(I,J)/SUMAE
    6        CONTINUE
          END IF
    7 CONTINUE
C
C  Write out the estimate of A to disk, ddname = 11
C
      DO 11 I=1,NOBJ
C**       WRITE (11,8) NEND
C** 8     FORMAT (I2)
          J1 = 1
          J2 = 8
C****                Integer arithmetic. To obtain 8 data-values per record
          KQ = NVAR/8
          KS = 8*(NVAR/8)
          IF (KS.LT.NVAR) KQ = KQ + 1
          DO 10 JJ =1,KQ
             IF (NVAR.GT.J2) THEN
                WRITE (11,9) (AE(I,J),J=J1,J2)
                WRITE (17,9) ( A(I,J),J=J1,J2)
             ELSE
                WRITE (11,9) (AE(I,J),J=J1,NVAR)
                WRITE (17,9) ( A(I,J),J=J1,NVAR)
             END IF
    9     FORMAT (8F10.4)
```

```
          J1 = J1 + 8
          J2 = J2 + 8
   10     CONTINUE
   11  CONTINUE
       WRITE(13,14)
       WRITE(13,15)
       PI = 4*QATAN(ONE)
       SUMA  = ZERO
       DO 13 I=1,NOBJ
          ANGLE = 180*QARCOS(DJ(I))/PI
          SUMA  = SUMA + ANGLE
          WRITE(13,12) I,DJ(I),ANGLE
   12     FORMAT(1X,I4,2F10.4)
   13  CONTINUE
       SUMA = SUMA/NOBJ
       WRITE(13,16) SUMA
   14  FORMAT(1X,'GOODNESS OF FIT BY ANGLES (BEFORE RESCALING OF DATA)')
   15  FORMAT('0',1X,'OBJECT NUMBER'/
     @        9X,'COSINES OF ANGLES BETWEEN PREDICTED AND OBSERVED'/
     @       19X,'ANGLES (DEGREES) BETWEEN PREDICTED AND OBSERVED')
   16  FORMAT('0',4X,'MEAN ANGULAR ERROR =',F10.4,' DEGREES')
       RETURN
C****                          End of procedure EST                          ****
       END
C*************************************************************************
C                              Procedure TITLE (Page throw and title)
C*************************************************************************
       SUBROUTINE TITLE
       REAL*16 ZERO,ONE,TEST
       CHARACTER*3 R,S
       COMMON /LABEL1/ZERO,ONE,TEST,NOBJ,NVAR,NEND,MAL,ISCALE,IUNIT,IPE
      @              ,ICONT,ICON,IFAULT
       WRITE (13,1)
     1 FORMAT ('1')
       R = 'No '
       S = 'No'
       IF (IUNIT.EQ.1) R = 'Yes'
       IF (IPE.EQ.1)   S = 'Yes'
       WRITE (13,2) NOBJ,NVAR,NEND,ISCALE,R,S
     2 FORMAT(1X,'NUMBERS OF OBJECTS =',I4,
      @          ',  VARIABLES =',I3,
      @          ',  END-MEMBERS =',I2,
      @          ',  SCALE NUMBER = ',I2,
      @          ',  ROW-UNITIZE = ',A3,
      @          ',  100% ROW-SUM = ',A3,/,'0')
       RETURN
       END
C*************************************************************************
C*                            Procedure CONLSQ                           *
C*************************************************************************
       SUBROUTINE CONLSQ(EM,X,XE,NVAR,NEND)
       REAL*16 EM(40,40),BZ(10,40),C(10,10),DZ(10),X(40),XE(40),
```

```
      @         Y(10),RMIN,RMAX,EMIJZ
       ZERO = 0.0Q+00
       RMIN = ZERO
         IZ = 0
C****                    Identify the largest negative
       DO   1 J=1,NVAR
          IF (XE(J).LT.RMIN) THEN
               IZ = IZ + 1
               JZ = J
             RMIN = XE(J)
          END IF
    1  CONTINUE
C****                       If a 'largest' negative estimate exists then ...
       IF (IZ.GT.0) THEN
           RMAX = QABS(EM(1,JZ))
             IM = 1
C****                          Identify the largest element in column JZ
           DO   2 I=1,NEND
              EMIJZ = QABS(EM(I,JZ))
              IF (EMIJZ.GT.RMAX) THEN
                  RMAX = EMIJZ
                    IM = I
              END IF
    2      CONTINUE
C****                          Exclude X(JZ) from vector X
          DO  3 J =1,NVAR
               J1 = J
               IF (J.GT.JZ) J1 = J - 1
               X(J1) = X(J)
    3     CONTINUE
C****                          Compute the new basis matrix
          DO 5 I = 1,NEND
               I1 = I
               IF (I.GT.IM) I1 = I - 1
               DO 4 J = 1,NVAR
                    J1 = J
                    IF (J.GT.JZ) J1 = J - 1
                    BZ(I,J) = EM(I,J) - EM(I,JZ)*EM(IM,J)/EM(IM,JZ)
                    BZ(I1,J1) = BZ(I,J)
    4          CONTINUE
    5     CONTINUE
          DO 8 I = 1,NEND - 1
               DO 7 J = 1,NEND - 1
                    C(I,J) = ZERO
                    DO 6 K = I,NVAR - 1
                         C(I,J) = C(I,J) + BZ(I,K)*BZ(J,K)
    6               CONTINUE
    7          CONTINUE
    8     CONTINUE
          NEND1 = NEND - 1
          CALL INVERS(C,NEND1,IFAULT)
C****                                   Continue provided matrix C non-sing
```

```
      IF (IFAULT.EQ.0) THEN
      DO  10 J = 1,NEND - 1
           Y(J) = ZERO
           DO   9 I = 1,NVAR - 1
                Y(J) = Y(J) + X(I)*BZ(J,I)
    9      CONTINUE
   10 CONTINUE
      DO 12 J = 1,NEND - 1
           DZ(J) = ZERO
           DO 11 K = 1,NEND - 1
                DZ(J) = DZ(J) + Y(K)*C(K,J)
   11      CONTINUE
   12 CONTINUE
      DO 14 J = 1,NVAR - 1
           XE(J) = ZERO
           DO 13 K = 1,NEND - 1
                XE(J) = XE(J) + DZ(K)*BZ(K,J)
   13      CONTINUE
   14 CONTINUE
C****                              Shuffle components of XE along
      DO 15 J = 1,NVAR - 1
           J1 = J - 1
           IF ((NVAR+1-J).LT.JZ) J1 = J
           XE(NVAR-J1) = XE(NVAR-J)
   15 CONTINUE
      XE(JZ) = ZERO
C****                            End if 'largest' negative ...
      END IF
C****                            End if IFAULT = 0
      END IF
      RETURN
C****                            End of procedure CONLSQ
      END
C*************************************************************************
C*                      Procedure INVERS                                 *
C*************************************************************************
      SUBROUTINE INVERS(A,N,IFAULT)
      REAL*16  A(10,10),B(10,10),ZERO,ONE,TEST,DET,PVT,RMAX,DUM
C****                 Form the inverse of NXN matrix A, and return as A
        ZERO = 0.0Q+00
        ONE  = 1.0Q+00
        TEST = 1.0Q-15
      IFAULT = 0
      DO 2 I=1,N
         DO 1 J=1,N
            B(I,J) = ZERO
    1    CONTINUE
         B(I,I) = ONE
    2 CONTINUE
      DET = ONE
C****                                                Outside loop starts below
      DO 9 J=1,N
```

```
C****                 Find largest element in column j (j < N) of matrix A
      KMAX = J
      IF (J.LT.N) THEN
         RMAX = QABS(A(J,J))
         DO 3 K=J+1,N
            IF (QABS(A(K,J)).GT.RMAX) THEN
               RMAX = QABS(A(K,J))
               KMAX = K
            END IF
    3    CONTINUE
      END IF
C****           Interchange the j-th and KMAX-th rows, maximising pivot
      IF (KMAX.GT.J) THEN
         DO 4 J1=1,N
            DUM = A(J,J1)
            A(J,J1) = A(KMAX,J1)
            A(KMAX,J1) = DUM
            DUM = B(J,J1)
            B(J,J1) = B(KMAX,J1)
            B(KMAX,J1) = DUM
    4    CONTINUE
      END IF
      PVT = A(J,J)
      DET = DET*PVT
      IF (QABS(PVT).GT.TEST) THEN
         DO 5 J1 = 1,N
            A(J,J1) = A(J,J1)/PVT
            B(J,J1) = B(J,J1)/PVT
    5    CONTINUE
         DO 7 I=1,N
            DUM = A(I,J)
            DO 6 J1=1,N
               IF (I.NE.J) THEN
                  A(I,J1) = A(I,J1) - A(J,J1)*DUM
                  B(I,J1) = B(I,J1) - B(J,J1)*DUM
               END IF
    6       CONTINUE
    7    CONTINUE
C****                                 Warning, near-singularity of matrix A
      ELSE
         WRITE(6,8)
    8    FORMAT(2(/),5X,'Processing stopped. Determinant approaching',
     1                 ' zero. ')
C****                                  Set flag to stop further processing
      IFAULT = 1
      GO TO 12
      END IF
    9 CONTINUE
      DO 11 I=1,N
         DO 10 J=1,N
            A(I,J) = B(I,J)
   10    CONTINUE
```

```
   11 CONTINUE
   12 CONTINUE
C****                                                    End of outside loop
      RETURN
C****                      End of procedure INVERS
      END
C*************************************************************************
C                          Procedure TRNSPS (transpose)
C*************************************************************************
      SUBROUTINE TRNSPS(V,EM,NVAR,NEND)
      REAL*16 EM(40,40),V(40,40)
      DO 2 I=1,NEND
         DO 1 J=1,NVAR
              EM(I,J) = V(J,I)
    1        CONTINUE
    2    CONTINUE
      RETURN
C****                       End of procedure TRNSPS                    ****
      END
```

```
C************************ LSQSEEK0 FORTRAN ****************************
C
C   NOTES:  This program reads the arguments for the iterative least
C           squares partitioning program LSQSEEK1 FORTRAN
C
      INTEGER IROW(10)
      CHARACTER*1 REPLY
      COMMON /LABEL1/IROW,NVAR,NEND,MAL,ISCALE,IUNIT,IG
C**********************************************************************
C                              MAIN program                            *
C**********************************************************************
C                                   Read required constants from terminal
      CALL LOAD
C                                        Write required constants to disk
      CALL WRITE
C
      STOP
C****                          End of MAIN program                  *****
      END
C
C**********************************************************************
C*                            Procedure LOAD  (load raw data)          *
C**********************************************************************
      SUBROUTINE LOAD
      INTEGER IROW(10)
      CHARACTER*1 REPLY
      COMMON /LABEL1/IROW,NVAR,NEND,MAL,ISCALE,IUNIT,IG
    1 CONTINUE
      WRITE(6,2)
    2 FORMAT(2(/),5X,'Enter the number of variables (at most 40)')
      READ(5,*) NVAR
      WRITE(6,3)
    3 FORMAT(/,5X,'Enter the number of end-members (at most 10)')
      READ(5,*) NEND
      WRITE(6,21)  NEND
   21 FORMAT (/,5X,'Key in the ',I2,' row numbers which identify end-mem
     @ers',/)
      DO 22 I=1,NEND
         READ(5,*) IROW(I)
   22 CONTINUE
    4 CONTINUE
      WRITE(6,5)
    5 FORMAT(/,5X,'Key in 0 for no scaling of variables,'/
     @          5X,'       1 for division by observed maximum,'/
     @          5X,'       2 for fractional ranges')
      READ(5,*) ISCALE
      IF (ISCALE.NE.0.AND.ISCALE.NE.1.AND.ISCALE.NE.2) GO TO 4
    6 CONTINUE
      WRITE(6,7)
    7 FORMAT(/,5X,'Row normalize (objects into unit vectors) ?  y/n')
      READ(6,8) REPLY
    8 FORMAT(A1)
```

```
      IF (REPLY.NE.'Y'.AND.REPLY.NE.'y'.AND.
     @    REPLY.NE.'N'.AND.REPLY.NE.'n') GO TO 6
      IUNIT = 1
      IF (REPLY.EQ.'N'.OR.REPLY.EQ.'n') IUNIT = 0
    9 CONTINUE
      WRITE(6,10)
   10 FORMAT(/,5X,'Key in number of iterations (or 0 for none)')
      READ(5,*) MAL
   11 CONTINUE
      WRITE(6,12)
   12 FORMAT(/,5X,'Key in 0 for LSQ  error vector coefficients,'/
     @         5X,'       1 for mean error vector coefficients')
      READ(5,*) IG
      IF (IG.NE.0.AND.IG.NE.1) GO TO 11
      WRITE(6,15)
   15 FORMAT(/,5X,'All correct ?   y/n ')
      READ(5,16) REPLY
   16 FORMAT(A1)
      IF (REPLY.NE.'Y'.AND.REPLY.NE.'y') GO TO 1
      RETURN
C****                    End of procedure LOAD                        *****
      END
C
C*************************************************************************
C*                          Procedure WRITE (Write constants to disk)   *
C*************************************************************************
      SUBROUTINE WRITE
      INTEGER IROW(10)
      COMMON /LABEL1/IROW,NVAR,NEND,MAL,ISCALE,IUNIT,IG
      WRITE(11,1) NVAR,NEND,MAL,ISCALE,IUNIT,IG
    1 FORMAT(1X,6I8)
      WRITE(11,2) (IROW(I),I=1,NEND)
    2 FORMAT(1X,10I6)
      RETURN
C****                    End of procedure WRITE                       *****
      END
```

```
C************************* LSQSEEK1 FORTRAN ****************************
C
C  NOTES: (1) A least squares iterative partitioning program
C         (2) Input raw data must be in freefield, row totals = 100%
C         (3) Summary information is written to disk ddname 13,
C             the merged matrices A plus estimated A ddname 11,
C             the loading matrix, ddname 15, the eigenvectors ddname 19,
C             and when k = 2, the two most extreme endmembers possible
C             ddname 21.
C
C
C
      REAL*16 A(800,40),AE(800,40),E(800,40),D(800,40),R(800),
     @        RANGE(40),C(40,40),B(40,40),EX(40,40),OP(40,40),
     @        SSQS,ZERO,ONE,TEST
      INTEGER IROW(40)
C
      COMMON /LABEL1/ZERO,ONE,TEST,NOBJ,NVAR,NEND,MAL,ISCAL,IUNT,IFAULT
     @       /LABEL2/A,EX,OP,IROW
     @       /LABEL3/RANGE
     @       /LABEL4/B
     @       /LABEL5/D,AE
     @       /LABEL6/R
     @       /LABEL7/E
     @       /LABEL8/IG
      ZERO = 0.0Q+00
      ONE  = 1.0Q+00
      TEST = 1.0Q-20
C*********************************************************************
C                              MAIN program                           *
C*********************************************************************
C                                                  Load raw data from disk
      CALL LOAD
C                                       Rescale columns into fractional ranges
      CALL SCALE
C                                              Rescale rows into unit vectors
      CALL UNIT
C
      CALL EMS
C
      CALL SEEK
C
      CALL COMPTS
C
      CALL ENDMEM
C
      CALL EST
C
      STOP
C****                          End of MAIN program                    *****
```

```
      END
C
C*************************************************************************
C*                          Procedure LOAD  (load raw data)              *
C*************************************************************************
      SUBROUTINE LOAD
      REAL*16 A(800,40),EX(40,40),OP(40,40),ZERO,ONE,TEST
      INTEGER IROW(40)
      COMMON /LABEL1/ZERO,ONE,TEST,NOBJ,NVAR,NEND,MAL,ISCAL,IUNT,IFAULT
     @       /LABEL2/A,EX,OP,IROW
     @       /LABEL8/IG
C****                  Program reads numbers of elements, end-members etc
      READ(12,*) NVAR,NEND,MAL,ISCAL,IUNT,IG
      READ(12,*) (IROW(I),I=1,NEND)
C****                       Program reads and counts rows of input matrix
      NOBJ = 1
    1 CONTINUE
C****                                    Input matrix must be in freefield
      READ (10,*, END = 2) (A(NOBJ,J),J = 1,NVAR)
      NOBJ = NOBJ + 1
      GO TO 1
    2 CONTINUE
C****                 Program counts one more than true number of records
      NOBJ = NOBJ - 1

      RETURN
C****                      End of procedure LOAD                      *****
      END
C*************************************************************************
C                           Procedure SCALE  (scale columns)             *
C*************************************************************************
      SUBROUTINE SCALE
      REAL*16 A(800,40),EX(40,40),OP(40,40),
     @        RANGE(40),RMAX,RMIN,ZERO,ONE,TEST
      INTEGER IROW(40)
      COMMON /LABEL1/ZERO,ONE,TEST,M,N,KE,MAL,ISCAL,IUNT,IFAULT
     @       /LABEL2/A,EX,OP,IROW
     @       /LABEL3/RANGE
C*** Column ranges are initialized to one in case there is no scaling
      DO 1 J=1,N
         RANGE(J) = ONE
    1 CONTINUE
      IF (ISCAL.GT.0) THEN
         CALL TITLE
         DO 6 J=1,N
            M1=1
            DO 2 K=2,M
               IF (A(K,J).GT.A(M1,J)) M1 = K
    2       CONTINUE
            RMAX = A(M1,J)
            M1=1
            DO 3 K=2,M
```

```
               IF (A(K,J).LT.A(M1,J)) M1 = K
     3         CONTINUE
               RMIN = A(M1,J)
               IF (ISCAL.EQ.1) RANGE(J) = RMAX
               IF (ISCAL.EQ.2) RANGE(J) = RMAX - RMIN
               DO 4 I=1,M
                  IF (ISCAL.EQ.1) A(I,J)= A(I,J)/RMAX
                  IF (ISCAL.EQ.2) A(I,J)=(A(I,J)-RMIN)/RANGE(J)
     4         CONTINUE
               WRITE(13,5) J,RMAX,RMIN
     5         FORMAT(1X/5X,'VARIABLE',I3,5X,'MAXIMUM =',F13.6,
      @                5X,'MINIMUM =',F13.6)
     6    CONTINUE
       END IF
       RETURN
C****                      End of procedure SCALE                        ****
       END
C*************************************************************************
C                          Procedure UNIT  (unit vectors)                  *
C*************************************************************************
       SUBROUTINE UNIT
       REAL*16 A(800,40),EX(40,40),OP(40,40),
      @        R(800),SSQ,ZERO,ONE,TEST,RLNGTH
       INTEGER IROW(40)
       COMMON /LABEL1/ZERO,ONE,TEST,M,N,KE,MAL,ISCAL,IUNT,IFAULT
      @       /LABEL2/A,EX,OP,IROW
      @       /LABEL6/R
C****                       If IUNT = 1, the rows of A become unit vectors
C*****                    R(I) is the length of the i-th row vector
          DO 3 I=1,M
             R(I) = ONE
             IF (IUNT.EQ.1) THEN
                  SSQ=ZERO
                  DO 1 J=1,N
                     SSQ = SSQ + A(I,J)**2
     1            CONTINUE
                  RLNGTH = QSQRT(SSQ)
                  DO 2 J=1,N
                     A(I,J) = A(I,J)/RLNGTH
     2            CONTINUE
                  R(I) = RLNGTH
             END IF
     3     CONTINUE
       RETURN
C****                      End of procedure UNIT                         ****
       END
C*************************************************************************
C                       Procedure EMS (Copy rows of A into end-members)
C*************************************************************************
       SUBROUTINE EMS
       REAL*16 A(800,40),B(40,40),C(40,40),EX(40,40),OP(40,40),
      @        ZERO,ONE,TEST
```

```
      INTEGER IROW(40)
      COMMON /LABEL1/ZERO,ONE,TEST,NOBJ,NVAR,NEND,MAL,ISCAL,IUNT,IFAULT
     @       /LABEL2/A,EX,OP,IROW
     @       /LABEL4/B
C****                 Form array of end-members from specified rows of A
      DO 2 I=1,NEND
         DO 1 J=1,NVAR
            B(I,J) = A(IROW(I),J)
    1    CONTINUE
    2 CONTINUE
      DO  5 I=1,NEND
         DO  4 J=1,NEND
            C(I,J) = ZERO
            DO  3 K=1,NVAR
                C(I,J) = C(I,J) + B(I,K)*B(J,K)
    3       CONTINUE
    4    CONTINUE
    5 CONTINUE
      CALL INVERS(C,NEND,IFAULT)
      DO 8 I=1,NEND
         DO 7 J=1,NVAR
            EX(I,J) = ZERO
            DO 6 K=1,NEND
               EX(I,J) = EX(I,J) + C(I,K)*B(K,J)
    6       CONTINUE
    7    CONTINUE
    8  CONTINUE
C**  Construct the orthogonal projection operator OP (pxp), into S-space
      DO 11 I=1,NVAR
         DO 10 J=1,NVAR
            OP(I,J) = ZERO
            DO  9 K=1,NEND
               OP(I,J) = OP(I,J) + B(K,I)*EX(K,J)
    9       CONTINUE
   10    CONTINUE
   11 CONTINUE
C**         Store initial EMs in matrix EX  (Extremes), spanning k-space
      DO 13 I=1,NEND
         DO 12 J=1,NVAR
            EX(I,J) = B(I,J)
   12    CONTINUE
   13 CONTINUE
      RETURN
C****                 End of procedure EMS
      END
C*********************************************************************
C                     Procedure SEEK (Search for EMS)
C*********************************************************************
      SUBROUTINE SEEK
      REAL*16 A(800,40),AE(800,40),E(800,40),RANGE(40),X(40),XE(40),
     @        D(800,40),C(40,40),EX(40,40),OP(40,40),B(40,40),F(40,40),
     @        Y(40),ZERO,ONE,TEST,SUM,EIJ
```

```
      INTEGER NE(800),NF(40),IROW(40)
      COMMON /LABEL1/ZERO,ONE,TEST,NOBJ,NVAR,NEND,MAL,ISCAL,IUNT,IFAULT
     @       /LABEL2/A,EX,OP,IROW
     @       /LABEL3/RANGE
     @       /LABEL4/B
     @       /LABEL5/D,AE
     @       /LABEL7/E
     @       /LABEL8/IG
C
      CALL TITLE
      WRITE (6,*) '>>>>>>>> IG = ',IG
C***                          Permit zero number of iterations
      MAL = MAL + 1
      DO 31 IT = 1,MAL
          IT1 = IT - 1
       DO 3 I=1,NEND
          DO 2 J=1,NEND
             C(I,J) = ZERO
             DO 1 K=1,NVAR
                C(I,J) = C(I,J) + B(I,K)*B(J,K)
    1        CONTINUE
    2     CONTINUE
    3  CONTINUE
          CALL INVERS(C,NEND,IFAULT)

C**                          Compute the loading matrix D(I,J).
       DO  8 M=1,NOBJ
          DO 5 J=1,NEND
             Y(J) = ZERO
             DO 4 I=1,NVAR
                Y(J) = Y(J) + A(M,I)*B(J,I)
    4        CONTINUE
    5     CONTINUE
          DO 7 J=1,NEND
             D(M,J)  = ZERO
             DO 6 K=1,NEND
                D(M,J) = D(M,J) + Y(K)*C(K,J)
    6        CONTINUE
C****  All loadings must be non-negative
           IF (D(M,J).LT.ZERO) D(M,J) = ZERO
    7     CONTINUE
    8  CONTINUE
      IF (IUNT.EQ.0) THEN
C***                      Rescale loadings to unit row-sums
       DO 11 I=1,NOBJ
          SUM = ZERO
          DO  9 J=1,NEND
             SUM = SUM + D(I,J)
    9     CONTINUE
          DO 10 J=1,NEND
             D(I,J) = D(I,J)/SUM
   10     CONTINUE
```

```
   11  CONTINUE
       END IF
C****                  Form AE, estimate of (scaled) matrix A          *****
        DO  14 I=1,NOBJ
           DO  13 J=1,NVAR
              AE(I,J) = ZERO
              DO  12 K=1,NEND
                 AE(I,J) = AE(I,J) + D(I,K)*B(K,J)
   12         CONTINUE
C***       Construct the error matrix E,  E(I,J) = EIJ
                    EIJ  = A(I,J) - AE(I,J)
C***       Prevent possible underflow later
                   IF ((-TEST.LT.EIJ).AND.(EIJ.LT.TEST)) EIJ = ZERO
                   E(I,J) = EIJ
   13       CONTINUE
   14  CONTINUE
       CALL MEANSQ(IT1)
C*****                          **** Test to continue iterations       *****
       IF (IT.LT.MAL) THEN
C*****                          **** Identify zero error vectors       *****
       DO 300 I = 1,NOBJ
         SSQ = ZERO
         DO 301 J = 1,NVAR
              SSQ = SSQ + E(I,J)*E(I,J)
  301    CONTINUE
         NE(I) = 0
         IF (SSQ.GT.TEST) NE(I) = 1
  300  CONTINUE
C*****                          **** Count the non-zero vectors        *****
       DO 303 J = 1,NEND
       NF(J) = 0
         DO 304 I = 1,NOBJ
            NC = 0
            IF (D(I,J).GT.ZERO) NC = 1
            NF(J) = NF(J) + NC*NE(I)
  304    CONTINUE
         IF (NF(J).EQ.0) NF(J) = 1
  303  CONTINUE
C***  Improve matrix B, construct error vector coefficient matrix G
C
C***  If IG = 0 then construct kxk matrix D-transpose*D (k = NEND)
C
       IF (IG.EQ.0) THEN
       DO 17 I=1,NEND
          DO 16 J=1,NEND
             C(I,J) = ZERO
             DO 15 K=1,NOBJ
                C(I,J) = C(I,J) + D(K,I)*D(K,J)
   15        CONTINUE
   16     CONTINUE
   17  CONTINUE
       CALL INVERS(C,NEND,IFAULT)
```

```
      END IF
C
C**** Construct the transpose of error vector coefficient matrix G
        DO 20 I=1,NOBJ
             DO 19 J=1,NEND
C***                                    EIJ is a dummy variable
                  EIJ   = ZERO
                  IF (IG.EQ.0) THEN
                       DO 18 K=1,NEND
                            EIJ = EIJ + D(I,K)*C(K,J)
   18                  CONTINUE
                  ELSE
                       EIJ = D(I,J)/NF(J)
                  END IF
                  AE(I,J) = EIJ
   19        CONTINUE
   20    CONTINUE
         DO 100 I=1,NOBJ
              NZ  = 0
              DO 101 J=1,NEND
                   IF (D(I,J).GT.ZERO)       NZ = NZ + 1
                   IF ((D(I,J).LT.TEST).AND.(IG.EQ.1)) AE(I,J) = ZERO
  101         CONTINUE
              IF (NZ.EQ.NEND) THEN
                   DO 110 J = 1,NEND
                        AE(I,J) = ZERO
  110              CONTINUE
              END IF
  100    CONTINUE
         DO 103 I=1,NEND
              DO 104 J=1,NVAR
                   F(I,J) = ZERO
                   DO 105 K=1,NOBJ
                        F(I,J) = F(I,J) + AE(K,I)*E(K,J)
  105              CONTINUE
              B(I,J) = B(I,J) + F(I,J)
  104         CONTINUE
  103    CONTINUE
C****           Project new EMs into space S (constrained if necessary)
       DO  27 I = 1,NEND
              DO  23 J = 1,NVAR
                   C(I,J) = B(I,J)
   23         CONTINUE
              DO  25 J = 1,NVAR
                   B(I,J) = ZERO
                        DO  24 K=1,NVAR
                          B(I,J) = B(I,J) +  C(I,K)*OP(K,J)
   24                   CONTINUE
   25         CONTINUE
   27  CONTINUE
C*****            Enforce constant row-sums on matrix B and rescale  ****
       DO 30 I=1,NEND
```

```
            SUM = ZERO
            DO 28 J=1,NVAR
               IF (B(I,J).LT.ZERO) B(I,J) = ZERO
               B(I,J) = B(I,J)*RANGE(J)
                  SUM = SUM + B(I,J)
   28       CONTINUE
            DO 29 J=1,NVAR
               B(I,J) = 100*B(I,J)/(SUM*RANGE(J))
   29       CONTINUE
   30     CONTINUE
          END IF
   31   CONTINUE
        WRITE(13,32) (IROW(J),J=1,NEND)
   32   FORMAT(5(/),1X,'INITIAL END-MEMBERS AT ROWS',10I6)
        RETURN
C****                  End of procedure SEEK                                ****
        END
C*************************************************************************
C                       Procedure COMPTS (Components or loading matrix)  *
C*************************************************************************
        SUBROUTINE COMPTS
        REAL*16 A(800,40),AE(800,40),D(800,40),R(800),C(40,40),B(40,40),
       @         EX(40,40),OP(40,40),
       @         Y(40),SSQ(40),SUM,SSQI,SSQJ,ZERO,ONE,TEST
        INTEGER IROW(40)
        COMMON /LABEL1/ZERO,ONE,TEST,NOBJ,NVAR,NEND,MAL,ISCAL,IUNT,IFAULT
       @        /LABEL2/A,EX,OP,IROW
       @        /LABEL4/B
       @        /LABEL5/D,AE
       @        /LABEL6/R
C
        CALL TITLE
C****                          Unitized rows require rescaled loadings
        IF (IUNT.EQ.1) THEN
           DO 2 I=1,NOBJ
              DO 1 J=1,NEND
                 D(I,J) = D(I,J)/R(IROW(J))
    1         CONTINUE
    2      CONTINUE
           DO 5 I=1,NOBJ
              SUM = ZERO
              DO 3 J=1,NEND
                 SUM = SUM + D(I,J)
    3         CONTINUE
              DO 4 J=1,NEND
                 D(I,J) = D(I,J)/SUM
    4         CONTINUE
    5      CONTINUE
        END IF
C****                         Write loadings to diskfiles, ddnames 13 and 15
        WRITE (13,11) NEND
   11   FORMAT ('0',1X,'OBJECT NUMBER'/
```

```
     @           ,5X,'       THE',I3,' COLUMNS OF LSQ LOADING MATRIX'/)
      WRITE(13,19)  (IROW(J),J=1,NEND)
      WRITE(13,20)
      DO 14 I=1,NOBJ
         WRITE(13,12)  I,(100*D(I,J),J=1,NEND)
         WRITE(15,13)    (D(I,J),J=1,NEND)
   12    FORMAT(1X,I3,10(1X,F10.2))
   13    FORMAT(10F8.4)
   14 CONTINUE
C**********                  Row-unitized data
      IF (IUNT.EQ.1) THEN
      DO 16 J=1,NEND
         SSQJ = ZERO
         DO 15 I=1,NOBJ
            SSQJ = SSQJ + D(I,J)*D(I,J)
   15    CONTINUE
         SSQ(J) = SSQJ
   16 CONTINUE
      WRITE(13,17)
   17 FORMAT(/1X,'COLUMN SUMS OF SQUARES OF INITIAL LOADINGS (ROW UNITIZ
     @ED DATA)'/)
         WRITE(13,18)  (SSQ(J),J=1,NEND)
   18    FORMAT(4X,10(1X,F10.4))
   19 FORMAT(5X,10(1X,I10))
   20 FORMAT('0')
      END IF
      RETURN
C****                End of procedure COMPTS                          ****
      END
C*************************************************************************
C        Procedure ENDMEM (Rescale, store and print estimated EMs)      *
C*************************************************************************
      SUBROUTINE ENDMEM
      REAL*16 A(800,40),D(800,40),AE(800,40),B(40,40),C(40,40),DJ(800),
     @        EX(40,40),OP(40,40),
     @        RANGE(40),SUM,ZERO,ONE,TEST,PI,ANGLE,SUMA,SUMAE,DEL
      INTEGER IROW(40)
      COMMON /LABEL1/ZERO,ONE,TEST,NOBJ,NVAR,NEND,MAL,ISCAL,IUNT,IFAULT
     @       /LABEL2/A,EX,OP,IROW
     @       /LABEL3/RANGE
     @       /LABEL4/B
     @       /LABEL5/D,AE
C
      IF (NEND.EQ.2) THEN
         CALL TWOEXT
      END IF
C
C      Rescale the estimate of B
C
C      Since matrix multiplication is associative, columns are rescaled
C      before rows.
      CALL TITLE
```

```
C
      DO   3 I=1,NEND
         SUMA = ZERO
         DO   1 J=1,NVAR
            B(I,J) = B(I,J)*RANGE(J)
            SUMA   = SUMA + B(I,J)
    1    CONTINUE
         DO   2 J=1,NVAR
            B(I,J) = 100*B(I,J)/SUMA
    2    CONTINUE
    3 CONTINUE
C
C  Write out the estimated endmembers
C
      DO 11 I=1,NEND
         WRITE(13,8) IROW(I),I
    8    FORMAT(1X,'(',I4,')  END-MEMBER ',I2)
         J1 = 1
         J2 = 10
C****             Integer arithmetic. To obtain 10 data-values per record
         KQ = NVAR/10 + 1
         DO 10 JJ =1,KQ
            IF (NVAR.GT.J2) THEN
               WRITE (13,9) ( B(I,J),J=J1,J2)
            ELSE
               WRITE (13,9) ( B(I,J),J=J1,NVAR)
            END IF
    9    FORMAT (20X,10F10.4)
         J1 = J1 + 10
         J2 = J2 + 10
   10    CONTINUE
   11 CONTINUE
      DO 14 I=1,NEND
         J1 = 1
         J2 = 8
C****             Integer arithmetic. To obtain 8 data-values per record
         KQ = NVAR/8 + 1
         DO 13 JJ =1,KQ
            IF (NVAR.GT.J2) THEN
               WRITE (19,12) ( B(I,J),J=J1,J2)
            ELSE
               WRITE (19,12) ( B(I,J),J=J1,NVAR)
            END IF
   12    FORMAT (8F10.4)
         J1 = J1 + 8
         J2 = J2 + 8
   13    CONTINUE
   14 CONTINUE
      RETURN
C****                        End of procedure ENDMEM                    ****
      END
C*************************************************************************
```

```
C        Procedure EST (Estimate matrix A using end-members as a basis)*
C*************************************************************************
      SUBROUTINE EST
      REAL*16 A(800,40),D(800,40),AE(800,40),B(40,40),C(40,40),
     @         EX(40,40),OP(40,40),DJ(800),
     @        RANGE(40),SUM,ZERO,ONE,TEST,PI,ANGLE,SUMA,SUMAE,DEL
      INTEGER IROW(40)
      COMMON /LABEL1/ZERO,ONE,TEST,NOBJ,NVAR,NEND,MAL,ISCAL,IUNT,IFAULT
     @       /LABEL2/A,EX,OP,IROW
     @       /LABEL3/RANGE
     @       /LABEL4/B
     @       /LABEL5/D,AE
C
C      Compute the estimate of A (AE)
C
C      Since matrix multiplication is associative, columns are rescaled
C      before rows.
      CALL TITLE
C****                                    Row I must sum to 100 for A
      DO 7 I=1,NOBJ
         SUMA  = ZERO
         DO 5 J=1,NVAR
C****                                                   Rescale the columns
           A(I,J) =  A(I,J)*RANGE(J)
            SUMA   =    SUMA  +  A(I,J)
    5    CONTINUE
         DO 6 J=1,NVAR
            A(I,J) =  100*A(I,J)/SUMA
    6    CONTINUE
    7 CONTINUE
C
C      Call the A estimate AE(I,J).
C
      DEL = 1.00Q-20
      DO 4 I=1,NOBJ
         SUMA  = ZERO
         SUMAE = ZERO
         DJ(I) = ZERO
         DO 2 J=1,NVAR
            AE(I,J) = ZERO
            DO 1 K=1,NEND
               AE(I,J) = AE(I,J) + D(I,K)*B(K,J)
    1       CONTINUE
            SUMA  = SUMA  +  A(I,J)*A(I,J)
            SUMAE = SUMAE + AE(I,J)*AE(I,J)
    2    CONTINUE
         SUMAE = QSQRT(SUMAE)
         SUMA  = QSQRT(SUMA)
         DO 3 J=1,NVAR
            DJ(I) = DJ(I) + (AE(I,J)/SUMAE)*(A(I,J)/SUMA)
    3    CONTINUE
         DJ(I) = DJ(I) - DEL
```

```
    4 CONTINUE
C
C  Write out the estimate of A to disk, ddname = 11
C
      DO 11 I=1,NOBJ
         J1 = 1
         J2 = 8
C****              Integer arithmetic. To obtain 8 data-values per record
         KQ = NVAR/8 + 1
         DO 10 JJ =1,KQ
            IF (NVAR.GT.J2) THEN
               WRITE (11,9) (AE(I,J),J=J1,J2)
               WRITE (17,9) ( A(I,J),J=J1,J2)
            ELSE
               WRITE (11,9) (AE(I,J),J=J1,NVAR)
               WRITE (17,9) ( A(I,J),J=J1,NVAR)
            END IF
    9    FORMAT (8F10.4)
         J1 = J1 + 8
         J2 = J2 + 8
   10    CONTINUE
   11 CONTINUE
      WRITE(13,14)
      WRITE(13,15)
      PI = 4*QATAN(ONE)
      SUMA  = ZERO
      DO 13 I=1,NOBJ
         ANGLE = 180*QARCOS(DJ(I))/PI
         SUMA  = SUMA + ANGLE
         WRITE(13,12) I,DJ(I),ANGLE
   12    FORMAT(1X,I4,2F10.4)
   13 CONTINUE
      SUMA = SUMA/NOBJ
      WRITE(13,16) SUMA
   14 FORMAT(1X,'GOODNESS OF FIT BY ANGLES ')
   15 FORMAT('0',1X,'OBJECT NUMBER'/
     @       9X,'COSINES OF ANGLES BETWEEN PREDICTED AND OBSERVED'/
     @      19X,'ANGLES (DEGREES) BETWEEN PREDICTED AND OBSERVED')
   16 FORMAT('0',4X,'MEAN ANGULAR ERROR =',F10.4,' DEGREES')
      RETURN
C****                        End of procedure EST                        ****
      END
C*************************************************************************
C                            Procedure TITLE (Page throw and title)
C*************************************************************************
      SUBROUTINE TITLE
      REAL*16 ZERO,ONE,TEST
      CHARACTER*3 R
      COMMON /LABEL1/ZERO,ONE,TEST,NOBJ,NVAR,NEND,MAL,ISCAL,IUNT,IFAULT
      WRITE (13,1)
    1 FORMAT ('1')
      R = 'No '
```

```
      IF (IUNT.EQ.1) R = 'Yes'
      WRITE (13,2) NOBJ,NVAR,NEND,ISCAL,R
    2 FORMAT(2X,' NUMBER OF OBJECTS =',I4,
     @          ',  NUMBER OF VARIABLES =',I3,
     @          ',  NUMBER OF END-MEMBERS =',I2,
     @          ',  SCALE NUMBER = ',I2,
     @          ',  ROW-UNITIZE = ',A3,/,'0')
      RETURN
      END
C*********************************************************************
C      Procedure MEANSQ (Form mean of sum of squares of all nxp errors)*
C*********************************************************************
      SUBROUTINE MEANSQ(IT1)
      REAL*16 A(800,40),E(800,40),EX(40,40),OP(40,40),SSQ,ZERO,ONE,TEST
      INTEGER IROW(40)
      COMMON /LABEL1/ZERO,ONE,TEST,NOBJ,NVAR,NEND,MAL,ISCAL,IUNT,IFAULT
     @       /LABEL2/A,EX,OP,IROW
     @       /LABEL7/E
C
      SSQ = ZERO
      DO 2 I=1,NOBJ
         DO  1 J=1,NVAR
            SSQ = SSQ + E(I,J)*E(I,J)
    1    CONTINUE
    2 CONTINUE
      SSQ = SSQ/(NVAR*NOBJ)
      WRITE(13,16) IT1,SSQ
   16 FORMAT('0',4X,'ITERATION NUMBER ',I4,
     @             '      MEAN SQUARED ERROR =',F15.8)
      RETURN
C****                          End of procedure MEANSQ                  ****
      END
C*********************************************************************
C*                          Procedure CONLSQ                          *
C*********************************************************************
      SUBROUTINE CONLSQ(EM,X,XE,NVAR,NEND)
      REAL*16 EM(40,40),BZ(40,40),C(40,40),DZ(40),X(40),XE(40),
     @        Y(40),RMIN,RMAX,EMIJZ
      ZERO = 0.0Q+00
      RMIN = ZERO
        IZ = 0
C****                    Identify the largest negative
      DO   1 J=1,NVAR
         IF (XE(J).LT.RMIN) THEN
              IZ = IZ + 1
              JZ = J
            RMIN = XE(J)
         END IF
    1 CONTINUE
C****                      If a 'largest' negative estimate exists then ...
      IF (IZ.GT.0) THEN
          RMAX = QABS(EM(1,JZ))
```

```
          IM = 1
C****                          Identify the largest element in column JZ
       DO   2 I=1,NEND
          EMIJZ = QABS(EM(I,JZ))
          IF (EMIJZ.GT.RMAX) THEN
              RMAX = EMIJZ
                IM = I
          END IF
    2   CONTINUE
C****                          Exclude X(JZ) from vector X
      DO  3 J =1,NVAR
          J1 = J
          IF (J.GT.JZ) J1 = J - 1
          X(J1) = X(J)
    3 CONTINUE
C****                          Compute the new basis matrix
      DO 5 I = 1,NEND
          I1 = I
          IF (I.GT.IM) I1 = I - 1
          DO 4 J = 1,NVAR
               J1 = J
               IF (J.GT.JZ) J1 = J - 1
               BZ(I,J) = EM(I,J) - EM(I,JZ)*EM(IM,J)/EM(IM,JZ)
               BZ(I1,J1) = BZ(I,J)
    4          CONTINUE
    5     CONTINUE
          DO 8 I = 1,NEND - 1
               DO 7 J = 1,NEND - 1
                    C(I,J) = ZERO
                    DO 6 K = I,NVAR - 1
                         C(I,J) = C(I,J) + BZ(I,K)*BZ(J,K)
    6               CONTINUE
    7          CONTINUE
    8     CONTINUE
          NEND1 = NEND - 1
          CALL INVERS(C,NEND1,IFAULT)
C****                                  Continue provided matrix C non-sing
          IF (IFAULT.EQ.0) THEN
          DO  10 J = 1,NEND - 1
               Y(J) = ZERO
               DO   9 I = 1,NVAR - 1
                    Y(J) = Y(J) + X(I)*BZ(J,I)
    9          CONTINUE
   10     CONTINUE
          DO 12 J = 1,NEND - 1
               DZ(J) = ZERO
               DO 11 K = 1,NEND - 1
                    DZ(J) = DZ(J) + Y(K)*C(K,J)
   11          CONTINUE
   12     CONTINUE
          DO 14 J = 1,NVAR - 1
               XE(J) = ZERO
```

```
                DO 13 K = 1,NEND - 1
                    XE(J) = XE(J) + DZ(K)*BZ(K,J)
   13           CONTINUE
   14       CONTINUE
C****                                  Shuffle components of XE along
            DO 15 J = 1,NVAR - 1
                J1 = J - 1
                IF ((NVAR+1-J).LT.JZ) J1 = J
                XE(NVAR-J1) = XE(NVAR-J)
   15       CONTINUE
            XE(JZ) = ZERO
C****                               End if 'largest' negative ...
         END IF
C****                               End if IFAULT = 0
         END IF
         RETURN
C****                               End of procedure CONLSQ
         END
C*************************************************************************
C*                        Procedure INVERS                               *
C*************************************************************************
         SUBROUTINE INVERS(A,N,IFAULT)
         REAL*16  A(40,40),B(40,40),ZERO,ONE,TEST,DET,PVT,RMAX,DUM
C****                  Form the inverse of NXN matrix A, and return as A
           ZERO = 0.0Q+00
           ONE  = 1.0Q+00
           TEST = 1.0Q-15
         IFAULT = 0
         DO 2 I=1,N
            DO 1 J=1,N
               B(I,J) = ZERO
    1       CONTINUE
            B(I,I) = ONE
    2    CONTINUE
         DET = ONE
C****                                                 Outside loop starts below
         DO 9 J=1,N
C****                  Find largest element in column j (j < N) of matrix A
            KMAX = J
            IF (J.LT.N) THEN
               RMAX = QABS(A(J,J))
               DO 3 K=J+1,N
                  IF (QABS(A(K,J)).GT.RMAX) THEN
                     RMAX = QABS(A(K,J))
                     KMAX = K
                  END IF
    3          CONTINUE
            END IF
C****              Interchange the j-th and KMAX-th rows, maximising pivot
         IF (KMAX.GT.J) THEN
            DO 4 J1=1,N
               DUM = A(J,J1)
```

```
            A(J,J1) = A(KMAX,J1)
            A(KMAX,J1) = DUM
            DUM = B(J,J1)
            B(J,J1) = B(KMAX,J1)
            B(KMAX,J1) = DUM
    4     CONTINUE
        END IF
        PVT = A(J,J)
        DET = DET*PVT
        IF (QABS(PVT).GT.TEST) THEN
           DO 5 J1 = 1,N
              A(J,J1) = A(J,J1)/PVT
              B(J,J1) = B(J,J1)/PVT
    5     CONTINUE
          DO 7 I=1,N
             DUM = A(I,J)
             DO 6 J1=1,N
                IF (I.NE.J) THEN
                    A(I,J1) = A(I,J1) - A(J,J1)*DUM
                    B(I,J1) = B(I,J1) - B(J,J1)*DUM
                END IF
    6        CONTINUE
    7     CONTINUE
C****                                  Warning, near-singularity of matrix A
        ELSE
           WRITE(6,8)
    8     FORMAT(2(/),5X,'Processing stopped. Determinant approaching',
     1                 ' zero. ')
C****                                   Set flag to stop further processing
        IFAULT = 1
        GO TO 12
        END IF
    9   CONTINUE
        DO 11 I=1,N
           DO 10 J=1,N
              A(I,J) = B(I,J)
   10     CONTINUE
   11   CONTINUE
   12   CONTINUE
C****                                                   End of outside loop
        RETURN
C****                      End of procedure INVERS
        END
C*************************************************************************
C         Procedure TWOEXT (Most extreme possible pair of EMs)           *
C*************************************************************************
        SUBROUTINE TWOEXT
        REAL*16 A(800,40),D(800,40),AE(800,40),B(40,40),C(40,40),DJ(800),
       @        EX(40,40),OP(40,40),
       @        RANGE(40),SUM,ZERO,ONE,TEST,PI,ANGLE,SUMA,SUMAE,DEL
        INTEGER IROW(40)
        COMMON /LABEL1/ZERO,ONE,TEST,NOBJ,NVAR,NEND,MAL,ISCAL,IUNT,IFAULT
```

```
     @        /LABEL2/A,EX,OP,IROW
     @        /LABEL3/RANGE
     @        /LABEL4/B
     @        /LABEL5/D,AE
C
C  If NEND = 2 then the set of feasible estimates is a straight line.
C  This procedure locates the most extreme possible estimates on that
C  line.
C
      CALL TITLE
C
      NEG = 1
        I = 0
   99 CONTINUE
           I = I + 1
         DEL = (1.00Q-06)*I
         DO 100 J = 1,NVAR
              SUMA = (ONE + DEL)*EX(1,J) - DEL*EX(2,J)
              IF (SUMA.LT.ZERO) NEG = -1
  100    CONTINUE
      IF (NEG.GT.0) GO TO 99
      DEL = (1.00Q-06)*(I - 1)
      WRITE(13,201) DEL
  201 FORMAT(5X,'DEL = ',F10.4)
      DO 101 J = 1,NVAR
         C(1,J) = (ONE + DEL)*EX(1,J) - DEL*EX(2,J)
  101 CONTINUE
C
      NEG = 1
        I = 0
  199 CONTINUE
           I = I + 1
         DEL = (1.00Q-06)*I
         DO 102 J = 1,NVAR
              SUMA = (ONE + DEL)*EX(2,J) - DEL*EX(1,J)
              IF (SUMA.LT.ZERO) NEG = -1
  102    CONTINUE
      IF (NEG.GT.0) GO TO 199
      DEL = (1.00Q-06)*(I - 1)
      WRITE(13,301) DEL
  301 FORMAT(5X,'DEL = ',F10.4)
      DO 103 J = 1,NVAR
          C(2,J) = (ONE + DEL)*EX(2,J) - DEL*EX(1,J)
  103 CONTINUE
C
      WRITE(13,104)
  104 FORMAT('0',5X,'THE TWO MOST EXTREME POINTS POSSIBLE',/)
      DO   3 I=1,NEND
         SUMA = ZERO
         DO   1 J=1,NVAR
            C(I,J) =C(I,J)*RANGE(J)
             SUMA    = SUMA + C(I,J)
```

```
    1     CONTINUE
          DO   2 J=1,NVAR
              C(I,J) = 100*C(I,J)/SUMA
    2     CONTINUE
    3 CONTINUE
C
C  Write out the most extreme possible pair of EMs
C
      DO 11 I=1,NEND
         WRITE(13,8) I
    8     FORMAT(1X,'MOST EXTREME ',I2)
          J1 = 1
          J2 = 10
C****              Integer arithmetic. To obtain 10 data-values per record
          KQ = NVAR/10 + 1
          DO 10 JJ =1,KQ
             IF (NVAR.GT.J2) THEN
                WRITE (13,9) ( C(I,J),J=J1,J2)
             ELSE
                WRITE (13,9) ( C(I,J),J=J1,NVAR)
             END IF
    9     FORMAT (20X,10F10.4)
          J1 = J1 + 10
          J2 = J2 + 10
   10     CONTINUE
   11 CONTINUE
      DO 14 I=1,NEND
         J1 = 1
         J2 = 8
C****              Integer arithmetic. To obtain 8 data-values per record
         KQ = NVAR/8 + 1
         DO 13 JJ =1,KQ
            IF (NVAR.GT.J2) THEN
               WRITE (21,12) ( C(I,J),J=J1,J2)
            ELSE
               WRITE (21,12) ( C(I,J),J=J1,NVAR)
            END IF
   12    FORMAT (8F10.4)
         J1 = J1 + 8
         J2 = J2 + 8
   13    CONTINUE
   14 CONTINUE
      RETURN
C****                        End of procedure TWOEXT                    ****
      END
```