METHODS
AND
ALGORITHMS

\[\begin{align*}
G_{a+1} &= P G (C' P^2 G')^{-1} \\
H_{a+1} &= Q H (H' Q^2 H')^{-1} \\
E_{a+1} &= R E (E' R^2 E')^{-1}
\end{align*}\]
4.1 INTRODUCTION

In the previous chapter we have reviewed a large number of scalar-product models for three-mode data. In this chapter we will concentrate on two of them, viz. the Tucker3 and Tucker2 models. Both are the most general in the class of component models with three and two reduced modes, respectively. As much of the development of the solutions for the two Tucker models runs almost parallel, we will treat only the Tucker3 model in detail, and the Tucker2 model very briefly. After a discussion of the widely used methods developed by Tucker (1966a) to solve the three-mode principal component model and of their advantages and disadvantages, we will present in some (theoretical) detail an alternating least squares procedure for the Tucker3 model. This presentation will include theorems about the conditions for a unique solution, the development of an algorithm, and convergence properties of the algorithm. Furthermore two small examples using three-mode Hilbert matrices will be used to assess the accuracy of the algorithm, and a small Monte Carlo study will be reported which was designed to assess how increasing errors influence the recovery of a particular component space.

4.2 TUCKER'S METHODS FOR THREE-MODE PRINCIPAL COMPONENT ANALYSIS

Tucker (1966a, p.294-301) describes three methods to deal with the estimation of the parameters in the three-mode principal component model. The last of these models belongs more to the covariance structure approach, and will not be discussed here.
4.2

The essence of Method I is that cross-product matrices \( P, Q, R \) are formed for the three modes:

\[
P_{ii'} = \sum_{j=1}^{m} \sum_{k=1}^{n} z_{ijk} z_{i'jk}
\]

\[
q_{jj'} = \sum_{k=1}^{n} \sum_{i=1}^{m} z_{ijk} z_{ij'k}
\]

\[
r_{kk'} = \sum_{i=1}^{m} \sum_{j=1}^{n} z_{ijk} z_{ijk'}
\]

and that standard principal component matrices, say \( G, H, \) and \( E \), are computed via an eigenvalue-eigenvector decomposition for each of them. In general one only wants to retain the first few eigenvectors, assuming that the other vectors correspond to random variation in the data rather than systematic variation. \( G, H, \) and \( E \) are thus columnwise orthonormal, rather than orthonormal. From the formulation of the model it can be derived, as Tucker does for his method (p.292) and we will do for ours below in section 4.3, that the matrices \( G, H, \) and \( E \) are all that is necessary to arrive at an estimate for the parameters of the core matrix:

\[
c_{pqr} = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{l} g_{ip} h_{jq} e_{kr} z_{ijk}
\]

As long as all eigenvectors corresponding to non-zero roots are retained the estimators for \( c_{pqr} \) turn out to be least squares ones. However, when \( G, H, \) and \( E \) are truncated to the eigenvectors corresponding to the \( s, t, \) and \( u \) largest eigenvalues, then the estimators of the core matrix are no longer least squares ones, as the deletions of the later eigenvectors in the component matrices affect the estimators of the core matrix in a complicated way. This implies, for instance, that the relation

\[
\sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{l} z_{ijk}^2 = \sum_{s=1}^{m} \sum_{t=1}^{n} \sum_{u=1}^{r} c_{pqr}^2
\]

which is true for the complete decomposition with \( s=l \), \( t=m \), and \( r=n \), no longer has a simple equivalence for the reduced decomposition, e.g. in the form of a separate sum of squares for the core matrix and one for the error. In section 4.3 and Chapter 7 it will be shown, that the partitioning of the total sum of squares is a very powerful tool for interpretation. With Tucker's Method I it is not
clear how much of the total sum of squares of the data is accounted for by the reduced decomposition. Each of the modes will nearly always have different amounts of variation accounted for. It should, however, be noted that the better a model fits, the smaller the differences in amount of variation accounted for between the modes will be.

In the complete decomposition each squared element, \( c^2_{pqr} \) of the core matrix indicates the amount of variation accounted for by that element, or the combination of components that this element stands for - here the \( p \)-th component of the first mode, the \( q \)-th component of the second mode, and the \( r \)-th component of the third mode. This interpretation cannot be maintained when the amount of variation accounted for is different for each mode.

Tucker's Method II differs from Method I in the sequence of calculations. The components of the first mode are computed first, and they are immediately used to reduce the total data matrix to dimensions \( s \times m \times n \). This reduced matrix is then employed for the computation of the component matrix for the second mode, and the resulting components are used to reduce the data matrix once again now to dimension \( s \times l \times n \). A final singular value decomposition (see section 2.2) is used to find the core matrix, and the remaining components for the third mode. The purpose of this procedure is to circumvent the solving of the possibly large eigenvalue-eigenvector problem for one of the modes (usually individuals). The problems with discarding small roots are rather serious in this procedure, as the errors of approximation at one stage are passed on to the next stage. Incidentally, this way of solving the Tucker3 model demonstrates Bloxom's (forthcoming) point that the model is a third-order factor component model, and it fits nicely in the presentation of the model as three nested sets of linear combinations (see section 2.2).

Summing up it seems fair to say that, although Tucker's methods are acceptable in case of a complete decomposition, and in case of a good fit, they introduce problems in the reduced case with respect to the properties of the estimators, the interpretation of the core matrix, and the amount of variation accounted for by the components.
4.2

In the following sections we develop procedures which do not suffer from these defects, but which are conceptually and computationally more complex. It will be possible with the results of these procedures to assess the quality of the fit of the model and to partition the total sum of squares (see Chapter 7).

4.3 LEAST SQUARES SOLUTIONS FOR THE TUCKER3 MODEL

**Loss functions.** In section 2.3 and 3.2 we introduced the Tucker3 model

\[ z_{ijk} = \sum_{p=1}^{s} \sum_{q=1}^{t} \sum_{r=1}^{u} g_{ip} h_{jq} e_{kr} c_{pq} \]

with the restrictions that the columns of \( G = \{ g_{ip} \} \), \( H = \{ h_{jq} \} \), and \( E = \{ e_{kr} \} \) are orthonormal. A matrix formulation of the model can be given if we use Kronecker products, \( \otimes \), of the components (Bellman, 1960), and 'combination modes' (Tucker, 1966a, p.289) for the data matrix and the core matrix:

\[ Z = GC(H' \otimes E') \]

with the \((s \times m)\) matrix \( Z \), and the \((s \times tu)\) matrix \( C \) written with combination modes. By symmetry other matrix formulations are possible by using different combination modes and other component matrices in the Kronecker product. By using summation notation we can avoid the use of Kronecker products. In doing so, one formulation will suffice, and the symmetry of the model in the three modes is reflected by the formulation.

If we were interested in exactly decomposing \( Z \) into all its components, Tucker's methods would suffice to provide a solution for the decomposition, as we remarked above. However, in practical applications one is interested only in the first few principal components for each of the modes. In general this precludes finding an exact decomposition of \( Z \) into \( G, H, E, \) and \( C \). One therefore has to be satisfied with an approximation \( Z = GC(H' \otimes E') \), i.e. finding \( G, H, E, \) and \( C \) such that the difference between the model and the data is minimal according to some loss function. In slightly different terms, we have to look for the best approximate decomposition \( Z \) of
the three-mode matrix into G, H, E, and C according to the Tucker3 model.

In our case, as in many similar situations, we define a mean-squared loss function. Thus we search for an approximate solution Z such that

\[
f(G,H,E,C) = \sum_{i=1}^{\xi} \sum_{j=1}^{\eta} \sum_{k=1}^{\zeta} (z_{ijk} - \tilde{z}_{ijk})^2 = \sum_{i=1}^{\xi} \sum_{j=1}^{\eta} \sum_{k=1}^{\zeta} \sum_{p=1}^{s} \sum_{q=1}^{t} \sum_{u=1}^{u} g_{ip} h_{jq} e_{kr} c_{pqr}
\]

is minimal. The minimization has to be carried out under the restrictions of the model, i.e. G, H, and E have to be columnwise orthonormal. The Z for which f attains its minimum will be designated as \( Z = \{ \tilde{z}_{ijk} \} \), with

\[
\tilde{z}_{ijk} = \sum_{p=1}^{s} \sum_{q=1}^{t} \sum_{u=1}^{u} g_{ip} h_{jq} e_{kr} c_{pqr}
\]

**Existence of a best approximate solution.** Our first task is to show that there exist indeed such G, H, E, and C that the loss function attains a (global) minimum. This can be done by showing first that for given G, H, and E a unique C can be found which minimizes the loss function, and which can be expressed in terms of G, H, and E. In this way we are left with a minimization over G, H, and E.

A lemma due to Penrose (1955 – see also Kroonenberg & De Leeuw, 1977, p.3-3, 3-4) states that there exists a unique \( \hat{C} \), such that the function \( h \),

\[
h(C) = \sum_{i=1}^{\xi} \sum_{j=1}^{\eta} \sum_{k=1}^{\zeta} (z_{ijk} - \hat{z}_{ijk})^2 = \sum_{i=1}^{\xi} \sum_{j=1}^{\eta} \sum_{k=1}^{\zeta} \sum_{i'=1}^{s} \sum_{j'=1}^{t} \sum_{k'=1}^{u} g_{i'p} h_{j'q} e_{kr} c_{i'j'k'}
\]

minimizes f for fixed G, H, and E, and that
\[ f(Z) = 0 \text{ iff } z_{ijk} \]

\[
\begin{align*}
&= \sum_{p=1}^{s} \sum_{q=1}^{t} \sum_{r=1}^{u} \sum_{i'=1}^{l} \sum_{j'=1}^{m} \sum_{k'=1}^{n} g_{ip} e_{i'} p_{jq} h_{j'} q_{kr} e_{k'} r_{i'j'k'}.
\end{align*}
\]

The implication of this result is that we may minimize \( f \) over \( G, H, \) and \( E \), after substituting the expression for \( \tilde{C} \) into the loss function. The optimal core matrix \( \tilde{C} \) can then be computed afterwards from the optimal \( \tilde{C}, \tilde{H}, \) and \( \tilde{E} \) by substituting their values in the above equation for \( \tilde{C} \).

The function \( f \) thus has to be optimized over the domain \( S \)

\[ S = \{ s | s = (G,H,E), G,H, \text{ and } E \text{ columnwise orthonormal} \} \]

which is a compact subset in a finite dimensional real space. Furthermore, \( f \) is a bounded continuous function on \( S \), therefore there exists a point \( s = (G,H,E) \) in \( S \), such that \( f \) attains its minimum. In other words the minimization problem always has a solution.

**Partitioning of total sum of squares.** Using the optimal \( \tilde{C} \) from the previous section we can thus write \( \tilde{Z} \) as

\[
\tilde{Z}_{ijk} = \sum_{p=1}^{s} \sum_{q=1}^{t} \sum_{r=1}^{u} \sum_{i'=1}^{l} \sum_{j'=1}^{m} \sum_{k'=1}^{n} g_{ip} e_{i'} p_{jq} h_{j'} q_{kr} e_{k'} r_{i'j'k'}.
\]

Using this formulation we can rewrite the loss function as

\[
f'(G,H,E) = \sum_{i=1}^{l} \sum_{j=1}^{m} \sum_{k=1}^{n} (z_{ijk} - \tilde{z}_{ijk})^2 =
\]

\[
= \sum_{i,j,k} \tilde{z}_{ijk}^2 + \sum_{i,j,k} \tilde{z}_{ijk}^2 - 2 \sum_{i,j,k} z_{ijk} \tilde{z}_{ijk}.
\]

The last term can be shown to be equal after algebraic manipulation, using the orthonormality of \( G, H, \) and \( E \), to

\[
2 \sum_{i,j,k} \sum_{p,q,r} g_{ip} e_{i'} p_{jq} h_{j'} q_{kr} e_{k'} r_{i'j'k'} z_{ijk} \tilde{z}_{i'j'k'} = 2 \sum_{i,j,k} \tilde{z}_{ijk}^2.
\]

In Appendix 4.1 we show that the orthonormality is not necessary for the above equation to hold but as we already assumed that \( G, H, \) and \( E \) are orthonormal, it is convenient to use this property. In a sense demanding orthonormality in a fixed model is thus analogous
to assuming statistical independence between the model and the error vectors in models with one random vector (see the discussion in section 3.5).

Thus

\[ f'(G,H,E)) = \sum_{i=1}^{\ell} \sum_{j=1}^{m} \sum_{k=1}^{n} (z_{ijk} - \bar{z}_{ijk})^2 = \sum_{i,j,k} z_{ijk}^2 - \sum_{i,j,k} \bar{z}_{ijk}^2 \]

A convenient way of expressing this is as follows

\[
\text{SS(Residuals)} = \text{SS(Data)} - \text{SS(Model)},
\]

or

\[
\text{SS(Res)} = \text{SS(Total)} - \text{SS(Fit)}. \]

Thus the total sum of squares may be partitioned into a residual sum of squares and a fitted sum of squares, and the above formulation shows that the minimization of the SS(Res) is equal to the maximization of the SS(Fit).

It is furthermore worthwhile to note that both the SS(Fit) and the SS(Res) can be further partitioned in many different ways, none of which involve cross-product terms. The sum of squares are after all built up from the squares of the contributions of each point \((i,j,k)\) to the SS(Fit) and the SS(Res). We may therefore partition them in an analysis-of-variance way in order to find influential or ill-fitting elements in each of the modes, or even three-mode combinations which contribute too much or too little to the fit or residual. These matters will be taken up in Chapter 7.

**Nature of the approximate solution.** As mentioned in the previous section we can either minimize the residual sum of squares, or maximize the fitted sum of squares to find \(G, H, \) and \(E.\) It turns out to be more efficient to maximize the SS(Fit):

\[ p(G,H,E) = \sum_{i=1}^{\ell} \sum_{j=1}^{m} \sum_{k=1}^{n} z_{ijk}^2 \]

We have already shown that

\[ \sum_{i,j,k} z_{ijk}^2 = \sum_{i,j,k} \bar{z}_{ijk}^2 \]

and

\[ \sum_{i,j,k} \bar{z}_{ijk}^2 = \sum_{i,j,k} \bar{z}_{ijk} \]

thus

\[ p(G,H,E) = \sum_{i,j,k} \sum_{p,q,r} \sum_{i',j',k'} g_{ip}^* e_{i'} h_{j'} q_{k'} k_{r} \bar{z}_{ijk} z_{i'j'k'} \]

with the constraints that \(G, H, \) and \(E\) are columnwise orthonormal.
4.3

Incorporating these constraints into the maximization equation we get

\[ p'(G,H,E,\Lambda,M,N) = \sum_{i,j,k} \sum_{p,q,r} \sum_{i',j',k'} \xi_{i'p} \xi_{j'q} \xi_{k'r} e_{i'j'k}' \]

\[ \sum_{p} p' \tilde{p}' \sum_{i} \sum_{j} \sum_{q} \sum_{r} \sum_{k} \xi_{i,p} \xi_{j,q} \xi_{k,r} \delta_{ii'} \delta_{jj'} \delta_{qq'} \delta_{rr'} \]

which reads in matrix notation

\[ p'(G,H,E,\Lambda,M,N) = \text{tr} G' \{ Z(HOE)(H'OE')Z' \} G - \text{tr} \Lambda (G'G-I_S) \]

\[ - \text{tr} M(H'H-I_T) - \text{tr} N(E'E-I_U) \]

The maximum of \( p \) follows from the requirement that the first order partial derivatives of \( p' \) are simultaneously zero at the maximum of \( p \), and the Hessian is negative. In Theorem 4.1 the nature of the solution of the maximization is given, while the proof can be found in the Appendix to Kroonenberg & De Leeuw (1980).

**Theorem 4.1 (Approximate solution)**

Let the following quantities be defined:

- \( Z \) is a three-mode data matrix
- \( P = \text{tr} G' \{ Z(HH'E'E')Z' \} G \)
- \( S = \{ s \mid s = (G,H,E) G, H, E \text{ columnwise orthonormal} \} \)
- \( U \) is eigenvector matrix of \( P = \{ p_{ii} \} \),

\[
P_{ii}' = \sum_{j=1}^{m} \sum_{k=1}^{n} \sum_{l=1}^{n} \sum_{t=1}^{t} \sum_{u=1}^{u} \sum_{v=1}^{v} \sum_{w=1}^{w} \sum_{z=1}^{z} \sum_{i'=1}^{i} \sum_{j'=1}^{j} \sum_{k'=1}^{k} \sum_{l'=1}^{l} \sum_{t'=1}^{t} \sum_{u'=1}^{u} \sum_{v'=1}^{v} \sum_{w'=1}^{w} \sum_{z'=1}^{z} \sum_{i''=1}^{i''} \sum_{j''=1}^{j''} \sum_{k''=1}^{k''} \sum_{l''=1}^{l''} \sum_{t''=1}^{t''} \sum_{u''=1}^{u''} \sum_{v''=1}^{v''} \sum_{w''=1}^{w''} \sum_{z''=1}^{z''} \sum_{i'''=1}^{i'''} \sum_{j'''=1}^{j'''} \sum_{k'''=1}^{k'''} \sum_{l'''=1}^{l'''} \sum_{t'''=1}^{t'''} \sum_{u'''=1}^{u'''} \sum_{v'''=1}^{v'''} \sum_{w'''=1}^{w'''} \sum_{z'''=1}^{z'''} \sum_{i''''=1}^{i''''} \sum_{j''''=1}^{j''''} \sum_{k''''=1}^{k''''} \sum_{l''''=1}^{l''''} \sum_{t''''=1}^{t''''} \sum_{u''''=1}^{u''''} \sum_{v''''=1}^{v''''} \sum_{w''''=1}^{w''''} \sum_{z''''=1}^{z''''} \sum_{i'''':''=1}^{i'''''} \sum_{j'''':''=1}^{j'''''} \sum_{k'''':''=1}^{k'''''} e_{i'j'k'} e_{i''j''k''} e_{i'''j'''k'''} \]

- \( V \) is eigenvector matrix of \( Q = \{ q_{jj} \} \),

\[
q_{jj}' = \sum_{k=1}^{n} \sum_{l=1}^{n} \sum_{t=1}^{t} \sum_{u=1}^{u} \sum_{v=1}^{v} \sum_{w=1}^{w} \sum_{z=1}^{z} \sum_{i'=1}^{i} \sum_{j'=1}^{j} \sum_{k'=1}^{k} \sum_{l'=1}^{l} \sum_{t'=1}^{t} \sum_{u'=1}^{u} \sum_{v'=1}^{v} \sum_{w'=1}^{w} \sum_{z'=1}^{z} \sum_{i''=1}^{i''} \sum_{j''=1}^{j''} \sum_{k''=1}^{k''} \sum_{l''=1}^{l''} \sum_{t''=1}^{t''} \sum_{u''=1}^{u''} \sum_{v''=1}^{v''} \sum_{w''=1}^{w''} \sum_{z''=1}^{z''} \sum_{i'''=1}^{i'''} \sum_{j'''=1}^{j'''} \sum_{k'''=1}^{k'''} \sum_{l'''=1}^{l'''} \sum_{t'''=1}^{t'''} \sum_{u'''=1}^{u'''} \sum_{v'''=1}^{v'''} \sum_{w'''=1}^{w'''} \sum_{z'''=1}^{z'''} \sum_{i''''=1}^{i''''} \sum_{j''''=1}^{j''''} \sum_{k''''=1}^{k''''} \sum_{l''''=1}^{l''''} \sum_{t''''=1}^{t''''} \sum_{u''''=1}^{u''''} \sum_{v''''=1}^{v''''} \sum_{w''''=1}^{w''''} \sum_{z''''=1}^{z''''} e_{i'j'k'} e_{i''j''k''} e_{i'''j'''k'''} \]

- \( W \) is eigenvector matrix of \( R = \{ r_{kk} \} \),

\[
r_{kk}' = \sum_{i=1}^{l} \sum_{j=1}^{l} \sum_{k=1}^{l} \sum_{t=1}^{t} \sum_{u=1}^{u} \sum_{v=1}^{v} \sum_{w=1}^{w} \sum_{z=1}^{z} \sum_{i'=1}^{i} \sum_{j'=1}^{j} \sum_{k'=1}^{k} \sum_{l'=1}^{l} \sum_{t'=1}^{t} \sum_{u'=1}^{u} \sum_{v'=1}^{v} \sum_{w'=1}^{w} \sum_{z'=1}^{z} \sum_{i''=1}^{i''} \sum_{j''=1}^{j''} \sum_{k''=1}^{k''} \sum_{l''=1}^{l''} \sum_{t''=1}^{t''} \sum_{u''=1}^{u''} \sum_{v''=1}^{v''} \sum_{w''=1}^{w''} \sum_{z''=1}^{z''} \sum_{i'''=1}^{i'''} \sum_{j'''=1}^{j'''} \sum_{k'''=1}^{k'''} \sum_{l'''=1}^{l'''} \sum_{t'''=1}^{t'''} \sum_{u'''=1}^{u'''} \sum_{v'''=1}^{v'''} \sum_{w'''=1}^{w'''} \sum_{z'''=1}^{z'''} \sum_{i''''=1}^{i''''} \sum_{j''''=1}^{j''''} \sum_{k''''=1}^{k''''} \sum_{l''''=1}^{l''''} \sum_{t''''=1}^{t''''} \sum_{u''''=1}^{u''''} \sum_{v''''=1}^{v''''} \sum_{w''''=1}^{w''''} \sum_{z''''=1}^{z''''} e_{i'j'k'} e_{i''j''k''} e_{i'''j'''k'''} \]

- \((U,V,W) \in S\).
Then

a. \((\mathcal{O}, \mathcal{H}, \mathcal{E}) \in \mathcal{S}\) is a stationary point of \(p\) if and only if \(\mathcal{O} = \mathcal{U}\), \(\mathcal{H} = \mathcal{V}\), and \(\mathcal{E} = \mathcal{W}\), or orthonormal rotations thereof;

b. \((\mathcal{O}, \mathcal{H}, \mathcal{E}) \in \mathcal{S}\) maximizes \(p\) if and only if their columns are eigenvectors corresponding to the largest \(s, t,\) and \(u\) eigenvalues of \(P(\mathcal{H}, \mathcal{E})\), \(Q(\mathcal{E}, \mathcal{O})\), and \(R(\mathcal{O}, \mathcal{H})\) respectively, or orthonormal rotations thereof.

In essence this theorem tells us that \(p\) can be maximized by simultaneously solving the eigenvalue-eigenvector problems of \(P\), \(Q\), and \(R\). Not surprisingly this cannot be done analytically, but only iteratively.

**Nature of exact solution.** Apart from knowing what the approximate solution looks like, it is also of theoretical and practical importance to know what the exact solution of the maximization problem looks like, if only the assess how well we have succeeded with our approximation. We will see that for the initialization of the algorithm for the approximate solution, the outcome of the theorem is relevant as well (section 4.5). As before we will only state the result here, the proof can be found in the Appendix to Kroonenberg & De Leeuw (1980).

**Theorem 4.2 (Exact solution)**

**A:**

Let the following quantities be defined:

- \(Z\) is a three-mode matrix

\[
\begin{align*}
  g & m n \\
  f(G, H, E, \mathcal{C}) = \sum_{i=1}^{g} \sum_{j=1}^{m} \sum_{k=1}^{n} (z_{ijk} - \bar{z}) \prod_{p=1}^{s} \prod_{q=1}^{t} \prod_{r=1}^{u} \mathcal{C}_{ip} \mathcal{H}_{jq} \mathcal{E}_{kr} \\
  p(G, H, E) = \sum_{i=1}^{g} \sum_{j=1}^{m} \sum_{k=1}^{n} \mathcal{C}_{ip} \mathcal{H}_{jq} \mathcal{E}_{kr} \\
  C = \{c_{pqr}\}, \text{ with } c_{pqr} = \sum_{i'=1}^{g} \sum_{j'=1}^{m} \sum_{k'=1}^{n} \mathcal{C}_{i'p} \mathcal{H}_{j'q} \mathcal{E}_{k'r}
\end{align*}
\]

Then the following statements are equivalent:
4.3

1. \( f(\hat{\mathbf{G}}, \hat{\mathbf{R}}, \hat{\mathbf{E}}, \hat{\mathbf{C}}) = 0, \)

2. \( p(\hat{\mathbf{G}}, \hat{\mathbf{R}}, \hat{\mathbf{E}}) = \sum_{i=1}^{s} \sum_{j=1}^{t} \sum_{k=1}^{u} z_{ijk} \)

3. \( z_{ijk} = \sum_{p=1}^{m} \sum_{q=1}^{n} \sum_{r=1}^{i'} \sum_{j'=1}^{j} \sum_{k'=1}^{k} \sum_{i''=1}^{i} \sum_{j''=1}^{j} \sum_{k''=1}^{k'} \hat{G}_{i'}^{p} \hat{R}_{j'}^{q} \hat{E}_{k'}^{r} Z_{i'}^{i''} Z_{j'}^{j''} Z_{k'}^{k''} \)

4. \((\hat{\mathbf{G}}, \hat{\mathbf{R}}, \hat{\mathbf{E}}, \hat{\mathbf{C}})\) is an exact solution of the minimization problem.

\textit{B.1:}

Let \((\hat{\mathbf{G}}, \hat{\mathbf{R}}, \hat{\mathbf{E}}, \hat{\mathbf{C}})\) be an exact solution of the minimization problem, then

\( \hat{\mathbf{G}} \) is the eigenvector matrix (or an orthonormal rotation thereof) corresponding to the \( s \) non-zero eigenvalues of

\[ p = \{p_{ii}\}, \text{ with } p_{ii} = \sum_{j=1}^{m} \sum_{k=1}^{n} z_{ijk}^2 i'j'k' \]

\( \hat{\mathbf{R}} \) is the eigenvector matrix (or an orthonormal rotation thereof) corresponding to the \( t \) non-zero eigenvalues of

\[ q = \{q_{jj}\}, \text{ with } q_{jj} = \sum_{i=1}^{s} \sum_{k=1}^{u} z_{ijk} z_{ij'k'} \]

\( \hat{\mathbf{E}} \) is the eigenvector matrix (or an orthonormal rotation thereof) corresponding to the \( u \) non-zero eigenvalues of

\[ r = \{r_{kk}\}, \text{ with } r_{kk} = \sum_{i=1}^{s} \sum_{j=1}^{t} z_{ijk} z_{ijk'} \]

\( \hat{\mathbf{C}} \) is the eigenvector matrix (or an orthonormal rotation thereof) corresponding to the \( \hat{\mathbf{C}} \) non-zero eigenvalues of

\[ c = \{c_{pq}\}, \text{ with } c_{pq} = \sum_{i'=1}^{s} \sum_{j'=1}^{s} \sum_{k'=1}^{s} \hat{G}_{i'}^{p} \hat{R}_{j'}^{q} \hat{E}_{k'}^{r} Z_{i'}^{i''} Z_{j'}^{j''} Z_{k'}^{k''} \]

\textit{B.2:}

On the other hand, if \( \hat{\mathbf{G}}, \hat{\mathbf{R}}, \hat{\mathbf{E}}, \hat{\mathbf{C}} \) are defined as in B.1, the eigenvalues associated with \( \hat{\mathbf{G}}, \hat{\mathbf{R}}, \hat{\mathbf{E}} \) are different for each matrix separately, and A.3 is satisfied, then

\((\hat{\mathbf{G}}, \hat{\mathbf{R}}, \hat{\mathbf{E}}, \hat{\mathbf{C}})\) is the exact unique solution.

It should be noted that statement B.2 is not as strong as one would like to have it, as any set of columnwise orthonormal matrices \( \mathbf{G}, \mathbf{H}, \mathbf{E} \), which satisfy A.3 determines an exact solution.
4.4 ALTERNATING LEAST SQUARES ALGORITHM FOR THE TUCKER3 MODEL

Introduction. Obviously we would like to construct an algorithm for the maximization of the fitted sum of squares, p, that converges to a global maximum. Unfortunately p is the cross-product term of a multivariate polynomial of the sixth degree, and in general it is not possible to prove that methods to solve such nonlinear problems attain a global optimum. In the present case this also seems to be true. We will have to be satisfied with proving that the algorithm outlined below will converge to some stationary point which is not a minimum rather than to a global maximum.

Alternating least squares approach. The method to be described utilizes the so-called alternating least squares (ALS) technique, already referred to above. The essential feature of the ALS approach is that in solving optimization problems with more than one set of parameters, each set is estimated in turn by applying conditional least squares procedures holding the other sets fixed. After all sets have been estimated once, the procedure is repeated again and again until convergence. Further details and references to applications of the ALS approach can, for instance, be found in Young, de Leeuw, & Takane (1980), Young (1981).

In order to see how the ALS approach can be applied in the present context, let us return to the definition of f:

\[
 f(G, H, E, C) = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{s} \sum_{l=1}^{t} (z_{ijk} - \bar{z}_{i} \bar{z}_{j} \bar{z}_{k} h_{lpq} e_{lqrst} c)^2
\]

The sets of parameters are here G, H, E, and C can be derived from the other three (see section 4.3). Minimizing f over G holding H and E fixed is identical to solving one conditional least squares problem, minimizing over H holding E and G fixed, and minimizing over E with G and H fixed are the two others. Although we are in practice maximizing over p the problem is still an ALS one.

From the above discussion a rough outline for an algorithm is readily deduced. First choose an arbitrary \( H_0 \) and \( E_0 \) and maximize over G to get a new \( G_1 \), maximize subsequently over H with the just
computed $G_1$ and $F_0$ fixed to get a new $H_1$, finally maximize over $E$ with $G_1$ and $H_1$ fixed to get a new $E_1$, and iterate the procedure until - one hopes - convergence. According to Theorem 4.1 the maximizations are essentially equal to searches for eigenvectors and eigenvalues of matrices of the order $\ell,m$, and $n$ respectively. As $\ell,m$, and $n$ can be quite large, while $s,t$, and $u$ are typically very small, say 2,3, or 4, we want to use a technique for solving the eigenvector-eigenvalue problem (or eigenproblem for short) which is particularly efficient in finding only the first few eigenvectors.

A very appropriate technique in this situation is the so-called simultaneous iteration method (or Treppen Iteration) of Bauer-Rutishauser (Rutishauser, 1969).

Thus, the maximization of $p$ consists of an, in principle, infinite iteration process, in which at each step three eigenproblems have to be solved. Clearly, solving these eigenproblems by another infinite iteration process has its drawbacks. The whole procedure is likely to become computationally cumbersome. In order to avoid this we perform only one single step towards the solution of the eigenproblems, instead of complete iterations. A similar approach has been applied by De Leeuw and others in a number of cases when using an ALS technique. The experience has been that carrying out the complete iteration to solve the eigenproblem only serves to decrease the overall efficiency of the procedure, while using only one step has no effect on the eventual convergence point (Takane, Young, & De Leeuw, 1977, p.59). They suggest that the reason for this behaviour might be found in the same reasons that often cause relaxation procedures to be more efficient than non-relaxation procedures.

**Simultaneous iteration method.** Let $A$ be a real $n \times n$ symmetric positive definite matrix, and $s$ the desired number of eigenvectors. Furthermore let $X$ be defined as a real $n \times s$ matrix which has for its columns the iteration vectors. If we write $X$ after a iterations as $X_\lambda$, the method of Bauer-Rutishauser is defined as follows.
i. Choose an arbitrary orthonormal $X_0$.

ii. $Y_a = AX_a$, and

iii. $B_a = Y_a^T Y_a$.

iv. Solve the eigenproblem for $B_a$, i.e. determine an orthonormal $T_a$ and a diagonal matrix $L_a$ with $\sum_{a=2}^{k} g_a$, such that $T_a^T B_a T_a = L_a$, and $T_a$ is the eigenvector matrix of $B_a$; then define

\[ v. \quad X_{a+1} = Y_a^T L_a^{-\frac{1}{2}}. \]

Schwartz et al. (1968, p.182-187) show that for $a \rightarrow \infty$, $L_a^{-\frac{1}{2}}$ converges to the matrix with the largest $s$ eigenvalues of $A$ on the diagonal, and the columns of $X_a$ converge to the associated eigenvectors, provided $A$ is positive definite, the columns of $X$ are not orthogonal to one or more of the eigenvectors, and the $s$-th and $(s+1)$-th eigenvalues are different. We will write ii. through v. somewhat more concisely by postmultiplying $v.$ with $T_a$:

\[ X_{a+1} = Y_a^T L_a^{-\frac{1}{2}} = AX_a^{-1} = AX_a (X_a^T A X_a)^{-1} \]

In practical applications we will not perform this postmultiplication as it would yield a rotated version of the principal components rather than the components themselves. For theoretical purposes the postmultiplication is immaterial, but convenient to work with. In Kroonenberg & De Leeuw (1980) the postmultiplication with $T_a$ for step v. was incorrectly included in the description and implementation of the algorithm presented in that paper. As stated above this does not affect the theoretical results, but in their example the components of what they called the 'unrotated' space were in fact rotated (by some $T_a$), and their 'rotated' components were approximately the principal components.

We will define the following function to be used later. When we use in the sequel functions like $\phi$ we mean to say that

\[ \phi(X_a) = X_{a+1} = AX_a (X_a^T A X_a)^{-1} \]

can be computed by carrying out one step of the Bauer-Rutishauser method. It should be noted that the inverse square root of $X_a^T A X_a$ exists, and is uniquely defined, if the expression is positive definite. This implies that in such a case $\phi$ is well-defined, and it can be proven that $\phi$ is continuous as well (see the Appendix to Kroonenberg & De Leeuw, 1980). As will be shown below, rather strong
convergence theorems can be used for the algorithm to be described if \( \phi \) is continuous. It seems, therefore, worthwhile to take measures in constructing the algorithm to ensure the positive definiteness of \( X'A^2X \). An inspection of the method to arrive at \( X_{a+1} \) shows that in fact only the inverse square root is taken of the eigenvalues of \( E_a \). One therefore only has to check in each iteration step if all eigenvalues are larger than zero, or in practice larger than some very small number. If one of the eigenvalues is too small, one can restart the iteration procedure with a smaller number of components. There is, however, no guarantee that this will solve the singularity problem. On the other hand, if no singularities have occurred one knows that at each step \( \phi \) must have been uniquely defined and continuous. As we check for the positiveness of the eigenvalues in our programs we will from now on assume that expressions like \( X'A^2X \) are positive definite.

In Kroonenberg & De Leeuw (1977) the proofs of the algorithm for the Tucker2 model were formulated without the extra condition that \( X'A^2X \) is always positive definite by using so-called point-to-set maps. Although convergence could then be proven, the uniqueness of the solution at each step is no longer assured. The additional complexity of working with point-to-set maps is, however, not really necessary as long as steps are taken to assure the positive definiteness.

**TUCKALS3 algorithm.** In this subsection we will describe the algorithm to solve the maximization of \( p \). Here \( Z \) is again defined as the \( E \times m \times n \) three-mode data matrix, and \( s, t, \) and \( u \) will be the desired number of components for the three component matrices. Furthermore the orthonormal matrices \( G, H, \) and \( E \) will be the matrices whose columns are the iteration vectors. We will write \( G, H, \) and \( E \) as they are after a iteration steps as \( G_a, H_a, \) and \( E_a \). One main iteration step of the TUCKALS3 algorithm is then defined as follows:
As mentioned before, each G, H, and E substep is one step of an inner iteration to find the eigenvectors of P, Q, and R respectively, and together they define one step of the main iteration.

**Convergence of the TUCKALS3 algorithm.** Before discussing the convergence of the algorithm itself, it is necessary to introduce some new notation.

F: \( S \rightarrow S \) is a function on \( S \); and \( F \) defines a complete step of the main iteration; and \( S \) is defined as in section 4.3 and **Theorem 4.1**. \( F = F_3 F_2 F_1 \) with \( F_i : S \rightarrow S \) for \( i = 1, 2, 3 \) such that

\[
F_1(G_{a+1}, H_{a+1} E_a) = (\phi_1(G_{a+1}), H_{a+1} E_a) = (G_{a+1}, H_{a+1} E_a)
\]
\[
F_2(G_{a+1}, H_{a+1} E_a) = (G_{a+1}, \phi_2(H_{a+1} E_a)) = (G_{a+1}, H_{a+1} E_a)
\]
\[
F_3(G_{a+1}, H_{a+1} E_{a+1}) = (G_{a+1}, \phi_3(E_{a+1})) = (G_{a+1}, H_{a+1} E_{a+1})
\]

Thus \( F(s_a) = F(G_{a+1}, H_{a+1} E_a) = (G_{a+1}, H_{a+1} E_{a+1}) = s_{a+1} \)
Above we already remarked that $\phi$ as defined in the previous subsection was a continuous function. Because $F$ is a composite of continuous functions $F$ is continuous as well.

It can be shown that at each step of the main iteration, and at each substep the value of $p$ does not decrease (see the Appendix to Kroonenberg & De Leeuw, 1980). Thus

$$p(F(s_a)) = p(s_{a+1}) \geq p(s_a)$$

If $p$ is not increased strictly, i.e. $p(F(s_a)) = p(s_a)$, the algorithm stops. In that case $(G, H, E)$ satisfies the necessary conditions of Theorem 4.3. Consequently we can assume without loss of generality that the algorithm generates infinite sequences with $p(F(s_a)) > p(s_a)$.

The TUCKALS3 algorithm is a type of algorithm that has been described in the non-linear programming literature, and in that field various theorems about the convergence of algorithms such as ours exist. Appropriate to our case is the following "fixed point" theorem described and proven by d’Esopo (1959):

**Theorem 4.3 (d’Esopo’s convergence theorem)**

Let $F, p, S$ satisfy the following conditions.

1. a. $S$ is a subset of a finite dimensional space,
   b. $F$ is a continuous function,
   c. $p$ is a real function defined and continuous for all $s \in S$,
2. $p(F(s)) \geq p(s)$,
3. if $p(F(s)) = p(s)$, then $F(s) = s$
4. if the sequence $s_0, s_1, \ldots$ satisfies $p(s_{a+1}) \geq p(s_a)$ with $s_a \in S$, then for every accumulation point $\hat{s}$ of $s_0, s_1, \ldots$ $F(\hat{s}) = \hat{s}$.

In the previous subsections we have discussed all the conditions of Theorem 4.3, and we may therefore conclude that it applies to the TUCKALS3 algorithm. As $S$ is a bounded real subspace, any infinite sequence $s_0, s_1, \ldots$ is bounded, and thus the sequences generated by the algorithm are bounded as well. A theorem due to Weierstrass shows that such sequences have at least one accumulation point. It is shown in the Appendix to Kroonenberg & De Leeuw
(1980) that every point \( \tilde{s} \) such that \( F(\tilde{s}) = \tilde{s} \) is a stationary point of \( p \), and because we know that at every step \( p \) increases, we know that stationary points will not be minima.

Still assuming the positive definiteness of expressions such as \( X'A^X \) we can use a theorem due to Meyer (1976, p.110) to show that \( [(s_{a+1} - s_a)] \neq 0 \), i.e. the normed difference between the component matrices in successive iterations becomes arbitrarily small. It would go too far to present the theorem in detail as it is formulated in terms of point-to-set maps and related concepts (see, however, Kroonenberg & De Leeuw, 1977, p.48 for details). As has been shown by Ostrowski (1966) the set of accumulation points of \( \{s_a\} \) consists either of a single point or a continuum. The latter case, however, is very unlikely in practical applications, as is the occurrence of equal eigenvalues in real data matrices. In order to be able to use Meyer's results, it seems sensible to control the convergence of the algorithm both by the objective function, and by the normed difference of all component matrices. In practice it generally turns out that the convergence for the components (i.e. the quantities we especially are interested in) is far slower than the convergence for the objective function.

4.5 NESTING OF COMPONENTS AND INITIALIZATION

Two questions with respect to increasing the number of components need to be answered. First, if we increase the number of components in the modes, does the SS(Fit) always increase as well? Secondly, are the solutions nested, i.e. is the configuration resulting from, say, just two components in the first mode the same as the two-dimensional configuration from the three component configuration of the first mode?

Nested solutions? The configurations of three-mode principal component analysis in its alternating least squares formulation are in general not nested. This can readily be seen from the algorithm itself. When for instance, the number of components of the first mode is increased, then the functions \( Q \) and \( R \) in the other substeps are directly influenced by this increased number of components, and
therefore different eigenproblems have to be solved. Table 4.1 gives an example of this lack of nesting. On the other hand, the better the fit, the better the nesting, and the more the alternating least squares solution resembles the initialization solution (Tucker’s method solution – see below), the better the nesting.

Table 4.1  Attachment study: Effect of increasing number of components on percentages explained variation of components

<table>
<thead>
<tr>
<th>solution</th>
<th>components</th>
<th>episodes</th>
<th>interactive scales</th>
<th>children</th>
<th>sum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1 2 3</td>
<td>1 2 3</td>
<td>1 2 3 4</td>
<td></td>
</tr>
<tr>
<td>2x2x2</td>
<td></td>
<td>35 23</td>
<td>39 19</td>
<td>40 8</td>
<td>57</td>
</tr>
<tr>
<td>2x2x3</td>
<td></td>
<td>37 27</td>
<td>41 22</td>
<td>49 8 6</td>
<td>63</td>
</tr>
<tr>
<td>2x2x4</td>
<td></td>
<td>39 27</td>
<td>42 24</td>
<td>48 9 6 2</td>
<td>66</td>
</tr>
<tr>
<td>3x3x3</td>
<td></td>
<td>36 27 5</td>
<td>41 25 2</td>
<td>49 10 9</td>
<td>68</td>
</tr>
<tr>
<td>3x3x4</td>
<td></td>
<td>37 28 8</td>
<td>45 25 3</td>
<td>49 10 8 6</td>
<td>73</td>
</tr>
</tbody>
</table>

Note: All analyses are based on 4 episodes, 4 interactive scales, and 53 children.

When increasing the number of components for any one mode, the SS(Fit) will in general increase. If the new solution is the global maximum of the SS(Fit) given the number of components, then the new SS(Fit) will never be smaller than the old one. If, however, the iteration stops at a local maximum, this is not necessarily so.

**Initialization.** In the algorithm we need some $G_o$, $H_o$, and $F_o$ to initialize the procedure. It seems sensible to choose them in such a way that they are optimal in some sense. Given that an exact solution exists, an initialization procedure which finds this solution without even entering the main iteration seems a reasonable choice. Theorem 4.28.1 tells us that the eigenvectors of $F$, $Q$,
and \( R \) with

\[
P_{ii}' = \sum_{j=1}^{m} \sum_{k=1}^{n} z_{ijk} \bar{z}_{i'j'k} \quad i'j'\quad \sum_{k=1}^{n} \sum_{j=1}^{l} z_{ijk} \bar{z}_{i'j'k}
\]

\[
r_{kk}' = \sum_{i=1}^{l} \sum_{j=1}^{m} z_{ijk} \bar{z}_{i'j'k}
\]

should be used for \( G, H, \) and \( E \) respectively to achieve this purpose. Referring back to section 4.2, the initialization procedure is seen to be identical to Method 1 proposed by Tucker (1966a, p.297). It is, in general, not necessary to determine these eigenvectors very precisely, because they are only the starting points for the main iteration. Here also the Bauer-Rutishauser procedure is used to compute the eigenvectors.

**Upper bounds for SS(Fit).** This particular initialization has an additional advantage, because it provides us with upper bounds for the SS(Fit) which is maximized in the main iteration procedure.

**Theorem 4.4 (Upper bounds for the SS(Fit))**

Let \((\bar{G}, \bar{H}, \bar{E})\) maximize \( p \) as defined above, then

\[
p (\bar{G}, \bar{H}, \bar{E}) \leq \min \left( \sum_{p}^{X} \sum_{q}^{X} \sum_{r}^{X} \sum_{i}^{Z} \sum_{j}^{Z} \sum_{k}^{Z} z_{ijk} \bar{z}_{i'j'k} \right)
\]

with the \( X \)'s, \( \mu_{q} \)'s, and \( \nu_{r} \)'s the largest eigenvalues of \( P, Q, \) and \( R \) respectively, or

\[
SS(Fit) \leq \min \{ SS(Fit_1), SS(Fit_2), SS(Fit_3) \} \leq SS(Total)
\]

The proof of this theorem can be found in the Appendix 4.2. In words the theorem says that the fitted sum of squares of the main iteration procedure, \( SS(Fit) \), can never be larger than the smallest of the fitted sum of squares resulting from standard principal component analyses on the cross-product matrices from each of the modes \( SS(Fit_m) \), \( m=1,2,3 \). These \( SS(Fit_m) \)s are in turn always smaller than or equal to the total sum of squares, \( SS(Total) \).

The practical importance of this result is that it is possible to gauge how well the iteration procedure has succeeded in finding
an optimal solution, given the number of components for each model. It can also point the way to improving the overall fit for the data, especially in the case where the choice of number of components for one mode was such that the fit possible with that number of components is very much smaller than the fit for the other modes. Suppose, for example, that the relative fit for the first mode is only 0.50, while for the other two modes a relative fit is possible of 0.80. The overall relative fit will then be necessarily no greater than 0.50 according to Theorem 4.4. The only really effective way to increase the overall relative fit is to include more components for the first mode, as it provides the smallest upper bound.

4.6 ALTERNATING LEAST SQUARES ALGORITHM FOR TUCKER2 MODEL

After the extensive discussion of the solution for the Tucker3 model, we can be very brief about the solution for the Tucker2 model. Its estimation poses no new problems; all one has to do is delete the $E$ substep from the TUCKALS3 algorithm, and insert the $k \times k$ identity matrix for $E$ in the other substeps. With these provisions the TUCKALS3 algorithm can be used for the Tucker2 model. Computationally it is, however, more efficient to solve the model by an analogue of the TUCKALS3 algorithm. The proofs for the TUCKALS2 algorithm are entirely parallel to those for the TUCKALS3 ones, and therefore need not be discussed again. Explicit proofs have been given in Kroonenberg & De Leeuw (1977) using a more general formulation of convergence, as was already mentioned above.

TUCKALS2 algorithm. We will only present the TUCKALS2 algorithm, and we will not discuss it in any detail, nor will we do so with the model. We will return to the model in Chapter 5 when we discuss the problem of transformations of the extended core matrix, and the uses of such transformations. In later chapters we will present a number of examples of the Tucker2 model.

\[
\begin{align*}
(a+1)^{-th} \text{ step of TUCKALS2} \\
G\text{-substep: } p_{ii}^a = \sum_{k=1}^n \sum_{j=1}^m \sum_{j'=1}^m \sum_{q=1}^t \sum_{i'k} h_{ij}^a h_{ij'}^{a'} a_{ij} a_{i'k} \\
G_{a+1} = P G (G' P^2 G)^{-\frac{1}{2}} 
\end{align*}
\]
4.6

\[ H_{a+1} = Q_n H_n \left( \theta^T \Omega \theta \right)^{-\frac{1}{2}} \]

4.7 COMPUTATIONAL ACCURACY AND PROPAGATION OF ERRORS

Developing formal expressions for the numerical accuracy of the programs is somewhat difficult considering the complexity of the algorithm. Especially modelling the propagation of rounding errors in successive iterations is not straightforward. The orthogonalization of the components at each iteration substep assures, however, that the errors of the components never run out of hand. In addition, the manipulation of very large or very small numbers is avoided by rescaling the overall variation of the data set to be analysed to \( \theta = n \times n \) (= volume of the data cube). Such scaling does not affect the components themselves, but only the absolute sizes of the eigenvalues and the elements of the core matrix (see also section 6.6). However, those quantities are in generally best interpreted as percentages of the total variation (see section 6.9), and in that respect the rescaling does not affect interpretation. The rescaling has the additional advantage that the convergence criteria are more or less equally strict for all problems.

In the examples below we will give some indication of the numerical accuracy of the TUCKALS3 program, and of the effect of introducing error in data with a known structure. No attempt has been made to treat these problems analytically or exhaustively.

Hilbert cubes and replicated Hilbert matrices. To our knowledge no three-way matrices with known eigenvalues have been published; therefore we have taken some substitutes, i.e. the Hilbert cube and the replicated Hilbert matrix.

The Hilbert matrix of the order \( n \), \( H_n \), is defined as \( H_n = \{ h_{ij}^n \} \) with \( h_{ij}^n = 1/(i+j-1) \), \( i,j=1,...,n \). Analogously the three-mode Hilbert cube can be defined as \( H_n = \{ h_{ijk}^n \} \) with \( h_{ijk}^n = 1/(i+j+k-2) \).
i, j, k = 1, ..., n. This cube is three-way symmetric (i.e. symmetric around the body diagonal). The component matrices derived from it should, therefore, be identical, and the core matrix should be three-way symmetric as well, provided the same number of components is taken for every mode. The identities and symmetry were realized until at least the tenth decimal. Several tests have been carried out with the initialization procedure described in section 4.5, and with random start matrices. In both cases the same results were obtained, be it that a solution was found almost without using the main iteration procedure when the initialization routine was employed. In other words, as was pointed out in section 4.5, if there exists an exact solution, the initialization procedure will find it first.

These results are, of course, no guarantee that the solutions are correct, they only have the desired form. But lacking analytical results it is the best we can do for the moment. For future reference the results for the Hilbert cube of order 4 are given in Table 4.2.

<table>
<thead>
<tr>
<th>Table 4.2 Hilbert cube of order 4 (with initialization procedure)</th>
</tr>
</thead>
<tbody>
<tr>
<td>component matrices and standardized weights</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>----</td>
</tr>
<tr>
<td>1. 0.7383  103</td>
</tr>
<tr>
<td>2. 0.4805  923</td>
</tr>
<tr>
<td>3. 0.3666  971</td>
</tr>
<tr>
<td>4. 0.2991  024</td>
</tr>
<tr>
<td>stand. weight 0.9819  994</td>
</tr>
<tr>
<td>standardized core matrix</td>
</tr>
<tr>
<td>c(1,1,1) = 0.9866 264</td>
</tr>
<tr>
<td>c(2,1,1) = -0.0022 338</td>
</tr>
<tr>
<td>c(3,1,1) = 0.0005 113</td>
</tr>
<tr>
<td>c(2,2,1) = 0.0923 996</td>
</tr>
<tr>
<td>c(3,2,1) = 0.0008 774</td>
</tr>
</tbody>
</table>

To have at least a partial check on the accuracy and the correctness of the algorithm and the program, a replicated Hilbert matrix of order 4 was used, i.e. \( Z = \{z_{ij, k}\} \) with \( z_{ij, k} = \frac{h^0_{ij, k}}{i+j-1} \). Each frontal plane is thus equal to the
same Hilbert matrix, and there are \( n \) of them. It is simple to show that \( G = H = K_n \), with \( K_n \) the eigenvector matrix of the Hilbert matrix of order \( n \), in which the eigenvectors are standardized at 1. \( E \) will be one dimensional with elements equal to \((1/n)^{1/3}\). In Table 4.3 the results for the replicated Hilbert matrix of order 4 are compared with the results given in Gregory & Karney (1969).

Table 4.3  Replicated Hilbert matrix of order 4

eigenvectors for mode 1 and 2

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>stem</td>
<td>G6K T3</td>
<td>stem</td>
</tr>
<tr>
<td>1</td>
<td>0.79260</td>
<td>829 832</td>
<td>0.58207</td>
</tr>
<tr>
<td>2</td>
<td>0.45192</td>
<td>312 306</td>
<td>-0.37050</td>
</tr>
<tr>
<td>3</td>
<td>0.32241</td>
<td>639 641</td>
<td>-0.50957</td>
</tr>
<tr>
<td>4</td>
<td>0.25216</td>
<td>117 119</td>
<td>-0.51404</td>
</tr>
</tbody>
</table>

| stand. weight | 0.98742| 851 902| 0.01255| 157 159| 0.00001| 992 992|

G6K = Gregory & Karney (1969); T3 = TUCKALS3; stem = same for both

With the replicated Hilbert matrix, as with the Hilbert cube, it was not possible to find the smallest root \((1.87 \times 10^{-8})\). In fact this eigenvalue was so small that the \(4 \times 4\) matrices \( P_a \) and \( Q_a \) (see section 4.4), of which the eigenvectors had to be computed, were considered singular by the program. The restart procedure, which reduces the number of components by the number of eigenvalues considered to be too small (see section 4.4), took care of this situation, and caused the program to start again with the reduced number of components.

Propagation of errors in similarity judgements. A small Monte Carlo experiment was conducted to gain some insight in the error propagation in the TUCKALS3 method. The data from the example presented in detail in Chapter 2 were subjected to various degrees of perturbation. More in particular, the data (unlike the real situation) were considered to be judgements on a nine-point scale of the similarity between the row and column points, with the row points serving as standards. To introduce error the observed simila-
rities were taken as the means of symmetric discrete distributions with varying heaviness of the tails. Thus if a stimulus combination \((i,j)\) was scored, for instance \(z_{ij} = 7\), then in error condition \(e1\) a score is generated from the discrete distribution:
\[
\begin{align*}
\Pr(z_{ij} = 6) &= \Pr(z_{ij} = 8) = 0.05; \\
\Pr(z_{ij} = 7) &= 0.90; \\
\Pr(z_{ij} = 1 \text{ or } 2 \text{ or } 3 \text{ or } 4 \text{ or } 5 \text{ or } 9) &= 0.
\end{align*}
\]
Table 4.4 shows how the end points were dealt with.

Table 4.4  Error structures for the De Gruijter data

<table>
<thead>
<tr>
<th>generated score, (Z_g)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>\ldots</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>original 2</td>
<td>95</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>\ldots</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>score</td>
<td>5</td>
<td>90</td>
<td>5</td>
<td>0</td>
<td>\ldots</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(Z_o)</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>\ldots</td>
<td>5</td>
<td>90</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>\ldots</td>
<td>0</td>
<td>5</td>
<td>95</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>generated score, (Z_g)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>75</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>\ldots</td>
</tr>
<tr>
<td>25</td>
<td>50</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>\ldots</td>
</tr>
<tr>
<td>15</td>
<td>50</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>\ldots</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>10</td>
<td>50</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>\ldots</td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>50</td>
<td>10</td>
<td>10</td>
<td>\ldots</td>
</tr>
</tbody>
</table>
| \(p_{ij} = \Pr\left(Z_g = j \mid Z_o = i\right)\)

The underlying idea in using this error structure was that a subject generally produces the number he intends (his 'true' score), but there is a non-zero probability that the similarity probably was meant to be higher or lower, and large differences from the intended score are less likely than smaller ones. Increasing errors, i.e. distributions with longer tails, imply increasing vagueness of the judgments. The results of the effects of increasing error in the data is summarized in Table 4.5, and Figure 4.1.
Table 4.5 Effects of perturbations on the De Gruijter data

Absolute differences with the unperturbed solution

**components for mode 1**

<table>
<thead>
<tr>
<th></th>
<th>e0</th>
<th>e1</th>
<th>e3</th>
<th>e5</th>
<th>e8</th>
<th>e0</th>
<th>e1</th>
<th>e3</th>
<th>e5</th>
<th>e8</th>
<th>e0</th>
<th>e1</th>
<th>e3</th>
<th>e5</th>
<th>e8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>48</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>-25</td>
<td>6</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>-4</td>
<td>0</td>
<td>5</td>
<td>4</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>48</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>-17</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>14</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>43</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>18</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>15</td>
<td>4</td>
<td>1</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>11</td>
<td>50</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>11</td>
<td>-9</td>
<td>1</td>
<td>7</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>-14</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>9</td>
<td>30</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>-20</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>22</td>
<td>7</td>
<td>3</td>
<td>4</td>
<td>8</td>
<td>-62</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>-22</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>29</td>
<td>1</td>
<td>1</td>
<td>11</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>8</td>
<td>-33</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>-33</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>14</td>
<td>29</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>39</td>
</tr>
<tr>
<td>9</td>
<td>-33</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>8</td>
<td>-28</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>3</td>
<td>34</td>
<td>0</td>
<td>7</td>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>-17</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>-46</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>-47</td>
<td>8</td>
<td>6</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

MAD  | 0.5  | 2.2  | 1.8  | 5.5  | 2.0  | 2.5  | 4.5  | 6.9  | 2.5  | 4.3  | 4.6  | 9.0  |

% | 61  | 56   | 52   | 45   | 28   | 21   | 20   | 20   | 17   | 14   | 11   | 11   | 9   | 7   |

**components for mode 3**

<table>
<thead>
<tr>
<th></th>
<th>e0</th>
<th>e1</th>
<th>e3</th>
<th>e5</th>
<th>e8</th>
<th>e0</th>
<th>e1</th>
<th>e3</th>
<th>e5</th>
<th>e8</th>
<th>e0</th>
<th>e1</th>
<th>e3</th>
<th>e5</th>
<th>e8</th>
<th>% variation accounted for SS(Fit)/SS(Total)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>42</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>68</td>
<td>6</td>
<td>9</td>
<td>44</td>
<td>67</td>
<td>42</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>34</td>
</tr>
<tr>
<td>2</td>
<td>40</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>-59</td>
<td>6</td>
<td>17</td>
<td>27</td>
<td>106</td>
<td>38</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>-19</td>
</tr>
<tr>
<td>3</td>
<td>38</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>-19</td>
<td>8</td>
<td>13</td>
<td>5</td>
<td>24</td>
<td>42</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>-10</td>
</tr>
<tr>
<td>4</td>
<td>39</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>-18</td>
<td>12</td>
<td>16</td>
<td>5</td>
<td>56</td>
<td>39</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>-10</td>
</tr>
</tbody>
</table>

MAD  | 0.7  | 2.0  | 1.0  | 2.5  | 7   | 18  | 18  | 68  | 1   | 1   | 1   | 1   | 3   |

% | 91  | 86   | 81   | 70   | 46  | 1   | 1   | 1   | 1   |

Notes:
- e0: 1.00
- e1: .05 .90 .05
- e3: .05 .10 .70 .10 .05
- e5: .10 .10 .10 .10 .10 .10 .10 .05
- e8: .10 .10 .10 .20 .10 .10 .10 .10 .10 .10

MAD: mean absolute difference
%: percentage variation accounted for
Fig. 4.1 Party similarity study: Perturbed first mode component loadings

The overall spatial organization of the points in the first (and second) mode is not fundamentally altered by the errors introduced, especially when the weights or percentages of variation accounted for by the components is sizeable. When they are not, the distortion can be very serious as the second component of the third mode shows. It is not unlikely that the data set and error structures chosen make a far better recovery possible than generally will be the case. The structure in the data is extremely well accounted for by the solution. The error structures are very regular, and favour good recovery, but unlike in some other studies they bear some relation to reality.
4.8 CONCLUSION

In this chapter we have shown that it is possible to solve the estimation problem of the Tucker3 and Tucker2 model both exactly and approximately by alternating least squares methods. The essential difference between the ALS methods presented in this chapter and Tucker's methods, is that the ALS procedures take into account the reduction over the other modes, while Tucker's methods do not. Furthermore, the use of least squares loss functions allows assessment of the model, analysis of residuals, and provides a number of attractive interpretational possibilities as we shall see in later chapters. The examples show that the method as programmed is accurate, and, for data with a well-defined structure, is robust against errors which could have arisen during the production of the data.
APPENDIX 4.1 PROOF THAT SS(TOT) = SS(FIT) + SS(RES)

Let $Z$ be a three-mode data matrix, and $G$, $H$, and $E$ arbitrary but fixed columnwise orthonormal matrices (see section 1.5). Let $\hat{C}$, a three-mode core matrix, minimize the loss function

$$[[Z - \hat{Z}]]^2 = \sum_{i,j,k} (z_{ijk} - \hat{z}_{ijk})^2$$

(A4.1)

$$\hat{z}_{ijk} = \sum_{p,q,r} g_{ip} h_{jq} e_{kr} c_{pqr}$$

then $\hat{C} = (\hat{c}_{pqr})$ is equal to (see section 4.3)

$$\hat{c}_{pqr} = \sum_{i,j,k} g_{ip} h_{jq} e_{kr} z_{ijk}$$

The $\hat{Z}$ which minimizes the loss function (A4.1) is

$$\hat{z}_{ijk} = \sum_{p,q,r} g_{ip} h_{jq} e_{kr} \hat{c}_{pqr} = \sum_{p,q,r} g_{ip} h_{jq} e_{kr} c_{pqr} = \sum_{p,q,r} g_{ip} h_{jq} e_{kr} = \sum_{p,q,r} g_{ip} h_{jq} e_{kr} c_{pqr}$$

In section 4.3 it was shown that $\hat{c}_{pqr}$ is uniquely defined, thus for any $c_{pqr}^* = \lambda c_{pqr}$, (A4.1) attains its minimum only if $\lambda = 1$ for all $p$. Let $z_{ijk}^*$ be defined as

$$z_{ijk}^* = \sum_{p,q,r} g_{ip} h_{jq} e_{kr} c_{pqr}^* = \sum_{p,q,r} g_{ip} h_{jq} e_{kr} c_{pqr}^* = \sum_{p,q,r} g_{ip} h_{jq} e_{kr} = \sum_{p,q,r} g_{ip} h_{jq} e_{kr} c_{pqr}$$

$$= \sum_{p,q,r} \lambda (g_{ip} h_{jq} e_{kr} c_{pqr}) = \sum_{p,q,r} \lambda \hat{c}_{pqr}^*$$

The loss function (A4.1) becomes with $z_{ijk}^*$

$$\sum_{i,j,k} (z_{ijk} - z_{ijk}^*)^2 =$$

$$\sum_{i,j,k} z_{ijk}^2 - 2 \sum_{i,j,k} (z_{ijk} \lambda t_{ijk}^*) + \sum_{i,j,k} (\lambda t_{ijk}^*)^2$$

(A4.2)

(A4.2) has a minimum if all $\lambda_p = 1$ simultaneously, thus
\[
\frac{\delta}{\delta \lambda_p} \left( \sum_{i,j,k} (z_{ijk} - \bar{z}_{ijk})^2 \right) = 0 \text{ for all } \lambda_p = 1 \\
- \sum_{i,j,k} \sum_{t_{ijkp}} + \sum_{i,j,k} \sum_{t_{ijkp}} \lambda_p t_{ijkp} = 0 \text{ for } \lambda_p = 1 \\
- \sum_{i,j,k} \sum_{t_{ijkp}} + \sum_{i,j,k} \sum_{t_{ijkp}} t_{ijkp} = 0 \\
\sum_{i,j,k} \sum_{t_{ijkp}} = \sum_{i,j,k} \sum_{t_{ijkp}} \left( \sum_{i,j,k} t_{ijkp} \right) \text{ for all } p \text{ simultaneously} \\
\sum_{i,j,k} \sum_{t_{ijkp}} = \sum_{i,j,k} \sum_{t_{ijkp}} \left( \sum_{i,j,k} t_{ijkp} \right) \\
\sum_{i,j,k} \sum_{t_{ijkp}} (\bar{z}_{ijk})^2 = \sum_{i,j,k} (\bar{z}_{ijk})^2 \\
\sum_{i,j,k} \sum_{t_{ijkp}} = \sum_{i,j,k} (\bar{z}_{ijk})^2 \\
\text{Thus using this last result, we may conclude that} \\
\sum_{i,j,k} (z_{ijk} - \bar{z}_{ijk})^2 = \sum_{i,j,k} (\bar{z}_{ijk})^2 - 2 \sum_{i,j,k} (\bar{z}_{ijk})^2 + \sum_{i,j,k} (\bar{z}_{ijk})^2 \\
= \sum_{i,j,k} (\bar{z}_{ijk})^2 \\
\text{SS(Res)} = \text{SS(Tot)} - \text{SS(Fit)}, \\
\text{or SS(Tot)} = \text{SS(Fit)} + \text{SS(Res)}.\]
APPENDIX 4.2  BOUNDS FOR THE SS(FIT)

Theorem 4.4

Let $G$, $H$, and $E$ maximize $p$ as defined in Theorem 1, then

$$p(G,H,E) \leq \min_{p,q,r} \left( \sum_{p} \lambda_p, \sum_{q} \nu_q, \sum_{r} \varphi_r \right) \leq \sum_{p,q,r} \sum_{i,j,k} z_{ijk}^2$$

with the $\lambda_p$, $\nu_q$, and $\varphi_r$ the largest eigenvalues of $P$, $Q$, and $R$ (see section 4.2 for definitions), respectively.

Proof:

Define $Z \in \mathbb{R}^{k \times mn}$, and $Z_i \in \mathbb{R}^{m \times n}$, $i=1, \ldots, k$.

$$p(G,H,E) = \text{tr} \ G'(Z(HH'\Theta E')Z')G$$

$$\leq \text{tr} \ Z(HH'\Theta E')Z' + \text{tr} \ \Sigma_{i} H'Z_i E_i Z_i' H_i$$

$$\leq \text{tr} \ \Sigma_{i} Z_i E_i Z_i' = \text{tr} \ E(\Sigma Z_i Z_i')E = \text{tr} E'RE$$

$$\leq \text{tr} \ \Sigma_{i} \nu_i, \text{ with } \nu_i \text{ the } u \text{ largest eigenvalues of } R$$

$$\leq \text{tr} \ \Sigma_{i} \nu_i = \Sigma_{i} \Sigma_{j,k} z_{ijk}^2$$

The inequalities follow from the so-called separation theorem, which states that the eigenvalues of an arbitrary section, say $E'RE$, of a symmetric matrix $R$ separate the eigenvalues of $R$ itself (cf. Householder, 1964).

The analogous results hold clearly for the other two formulations of $p(G,H,E)$, and combining these results establishes the theorem.

\[ \blacksquare \]