LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

INSTITUT FÜR STATISTIK

Leo Knüsel

Factor Analysis: Chisquare as Rotation Criterion

Technical Report Number 040, 2008
Department of Statistics
University of Munich

http://www.stat.uni-muenchen.de

Factor Analysis: Chisquare as Rotation Criterion

Leo Knüsel

Department of Statistics
University of Munich
80539 Munich, Germany
knuesel@stat.uni-muenchen.de

Abstract: The rotation problem in factor analysis consists in finding an orthogonal transformation of the initial factor loadings so that the rotated loadings have a simple structure that can be easily interpreted. The most popular orthogonal transformations are the quartimax and varimax procedure with Kaiser normalization. In this paper we propose the classical chisquare contingency measure as a rotation criterion. We think that this is a very natural and attractive criterion, not only for rotations but also for oblique transformations, that is not to be found in our popular statistical packages up to now.

Keywords: rotation criterion, chisquare, varimax, quartimax, oblique transformations.

1. Introduction and summary

The classical model of factor analysis (cf. Lawley and Maxwell, 1971) is given by

(1) $\mathbf{x} = \mathbf{\Lambda f} + \mathbf{v}$,

where $\mathbf{x} = (x_1, \dots, x_p)^\mathsf{T}$ is the vector of the observed variables, $\mathbf{f} = (f_1, \dots, f_k)^\mathsf{T}$ the vector of the latent common factors and $\mathbf{v} = (v_1, \dots, v_p)^\mathsf{T}$ the vector of the latent specific factors. The matrix $\mathbf{\Lambda} = (\lambda_{ir}) = (p \times k)$ is the socalled loading matrix. It is assumed that the variables $x_1, \dots, x_p$ are standardized so that $E(x_i) = 0$ and $Var(x_i) = 1$ for $i = 1, \dots, p$. Furthermore we assume that all common and specific factors are uncorrelated and that the common factors are standardized; then the covariance matrix $\mathbf{\Sigma} = (p \times p)$ of $\mathbf{x}$ is given by

(2) $\mathbf{\Sigma} = \mathbf{\Lambda\Lambda}^\mathsf{T} + \mathbf{V}$

where $\mathbf{V} = (p \times p)$ is a diagonal matrix with $v_{ii} = \text{var}(v_i)$. If $\mathbf{\Lambda}$ is replaced by $\tilde{\mathbf{\Lambda}} = \mathbf{\Lambda R}$ where $\mathbf{R} = (k \times k)$ is an arbitrary orthogonal matrix (rotation matrix) then $\tilde{\mathbf{\Lambda}}\tilde{\mathbf{\Lambda}}^\mathsf{T} = \mathbf{\Lambda R R}^\mathsf{T}\mathbf{\Lambda}^\mathsf{T} = \mathbf{\Lambda\Lambda}^\mathsf{T}$ and so equation (2) remains unchanged, and also the factor model (1) remains essentially unchanged as $\tilde{\mathbf{f}} = \mathbf{R}^\mathsf{T}\mathbf{f}$ is again a vector of standardized uncorrelated common factors and thus $\tilde{\mathbf{\Lambda}}\tilde{\mathbf{f}} = \mathbf{\Lambda R R}^\mathsf{T}\mathbf{f} = \mathbf{\Lambda f}$. So the loading matrix $\mathbf{\Lambda}$ is not uniquely fixed (not identifiable) by the factor model. But note that

$$\text{var}(x_i) = c_i + \text{var}(v_i) \quad \text{with} \quad c_i = \sum\nolimits_{r=1}^{k} \lambda_{ir}^2 ,$$

and as a diagonal element of $\mathbf{\Lambda\Lambda}^\mathsf{T}$ the term c_i remains unchanged under any orthogonal transformation of the loading matrix; c_i is called the communality of the variable x_i. The indeterminacy of the loading matrix can be used to find a rotation matrix **R** such that the rotated loading matrix $\tilde{\mathbf{\Lambda}} = \mathbf{\Lambda R}$ can be easily interpreted. The ideal simple structure of $\tilde{\mathbf{\Lambda}}$ (perfect cluster configuration, cf. Browne, 2001, p.116) were given if in every row there were just one loading different from zero. This would mean that every variable x_i were influenced by just one common factor, and the subset of variables that is influenced by a factor f_r ($r = 1, \dots, k$) would allow a natural characterization of this factor.

In section 2 we describe the quartimax and varimax criterion together with the chisquare criterion and show up some theoretical advantages and disadvantages of these criteria. Section 3, 4 and 5 describe the numerical solution with these three criteria according to the method of Lawley-Maxwell (1971). Section 6 shows how these algorithms can be modified if the iteration procedure fails to converge; an implementation of the algorithms in Maple and R is given in the Appendix. Section 7 gives some examples and the conclusion is found in section 8. The implementation of the algorithms and the data sets used in section 7 can be obtained from URL *www.stat.uni-muenchen.de/~knuesel*.

2. Rotation criteria

a) Quartimax criterion

We denote the matrix of squared factor loadings by $\mathbf{F} = (f_{ir}) = (\lambda_{ir}^2)$. According to Harman (1976, pp 283) the criterion to be maximized by rotating the loading matrix is the simple variance of the squared loadings $f_{ir} = \lambda_{ir}^2$:

$$Q = \frac{1}{pk}\sum_{i,r}(f_{ir} - \overline{f})^2 \quad \text{with} \quad \overline{f} = \frac{1}{pk}\sum_{i,r} f_{ir} .$$

We obviously have

$$Q = \frac{1}{pk}\sum_{i,r} f_{ir}^2 - \overline{f}^2 .$$

Now the communalities

$$c_i = \sum_{r=1}^{k}\lambda_{ir}^2 = \sum_{r=1}^{k} f_{ir} = f_{i\bullet},\ i = 1,\ldots,p ,$$

remain fixed under any rotation and thus also $\overline{f}$ remains fixed. So maximizing Q is equivalent to maximizing

(3) $$x_Q = \sum_{i,r} f_{ir}^2 = \sum_{i,r}\lambda_{ir}^4 .$$

This form explains the name quartimax. It can be easily proved (see Appendix A1) that x_Q becomes maximal if and only if every row of the loading matrix $\mathbf{\Lambda}$ contains only one element different from zero, and this means that the loading matrix has the ideal simple structure. The maximum value of x_Q is given by $\max x_Q = \sum c_i^2$, and this maximum can also be achieved if all loadings are concentrated on just one factor. According to Harman (1976, p 290) the tendency toward a general factor (one column of the loading matrix with a dominating sum of the squared loadings) is one of the main shortcomings of the quartimax solution.

b) Varimax criterion

According to Harman (1976, pp 290) the varimax criterion is given by

$$V = \frac{1}{k}\sum_{r=1}^{k}\sigma_r^2 \quad \text{with} \quad \sigma_r^2 = \frac{1}{p}\sum_{i=1}^{p}(f_{ir} - \overline{f}_r)^2, \quad f_{ir} = \lambda_{ir}^2, \quad \overline{f}_r = \frac{1}{p}\sum_{i=1}^{p} f_{ir} .$$

Here σ_r^2 is the variance of the squared loadings in column r. We obviously have

$$V = \frac{1}{pk}\sum_{r=1}^{k}\left(\sum_{i=1}^{p} f_{ir}^2 - p\,\overline{f}_r^2\right) = \frac{1}{pk}\sum_{r=1}^{k}\left(\sum_{i=1}^{p} f_{ir}^2 - d_r^2/p\right) \quad \text{with} \quad d_r = p\,\overline{f}_r = f_{\bullet r} = \sum_{i=1}^{p} f_{ir} = \sum_{i=1}^{p}\lambda_{ir}^2 .$$

So maximizing V is equivalent to maximizing

(4) $$x_V = \sum_{r=1}^{k}\left(\sum_{i=1}^{p} f_{ir}^2 - d_r^2/p\right) = x_Q - \frac{1}{p}\sum_{r=1}^{k} d_r^2 .$$

The varimax criterion becomes maximal if x_Q takes on its maximum and if $\sum d_r^2$ takes on its minimum which is the case if all column sums $d_1,\ldots,d_k$ are equal (see Appendix A2). The example below shows that for the varimax criterion to become maximal it is not enough that the loading matrix has an ideal simple structure. The varimax solution shows a tendency toward factors with equal sums $d_1,\dots,d_k$ whereas the quartimax solution shows a tendency to one dominating value of $d_1,\dots,d_k$ (general factor). If the communalities $c_1,\dots,c_p$ are different the variables $x_1,\dots,x_p$ do not have the same influence on the rotation (see Harman, 1976, p.291). Therefore one usually recommends to normalize the matrix $\mathbf{F}=(f_{ir})=(\lambda_{ir}^2)$ before rotation so that all row sums are 1 (Kaiser normalization).

Example: Let

$$\mathbf{F}_1=\begin{pmatrix}1&0\\1&0\\1&0\\1&0\\0&1\end{pmatrix}\qquad \mathbf{F}_2=\begin{pmatrix}1&0\\1&0\\0&1\\0&1\\0.5&0.5\end{pmatrix}$$

The matrix $\mathbf{F}_1$ has an ideal simple structure and the value of the varimax criterion is $x_V=1.6$, whereas $\mathbf{F}_2$ does not possess an ideal simple structure but its varimax criterion $x_V=2$ is larger than that of $\mathbf{F}_1$.

c) Chisquaremax criterion

Let $f_{ir}, i=1,\dots,p, r=1,\dots,k$ be the frequencies (nonnegative integers) in a contingency table with *p* rows and *k* columns. The well known classical contingency measure (measure of dependence) is given by

$$\chi^2=\sum_{i,r}\frac{(f_{ir}-e_{ir})^2}{e_{ir}} \quad \text{where} \quad e_{ir}=\frac{f_{i\bullet}f_{\bullet r}}{n},\ n=\sum_{i,r}f_{ir}\,.$$

It is well known that for fixed *n* and for $p\ge k$ the chisquare criterion χ^2 becomes maximal if and only if each row contains only one frequency f_{ir} different from zero and all column sums are positive, and the maximum is given by $\chi^2_{\max}=n(k-1)$ (see Cramer, 1945, p.443). We have

$$\chi^2=\sum_{i,r}\frac{(f_{ir}-e_{ir})^2}{e_{ir}}=\sum_{i,r}\frac{f_{ir}^2}{e_{ir}}-n=n\left(\sum_{i,r}\frac{f_{ir}^2}{f_{i\bullet}f_{\bullet r}}-1\right).$$

So maximizing χ^2 is equivalent to maximizing

$$x_C=\sum_{i,r}\frac{f_{ir}^2}{f_{i\bullet}f_{\bullet r}}$$

and the maximum of x_C is given by *k*.

Now we consider again a factorial model and set $f_{ir}=\lambda_{ir}^2$, $f_{i\bullet}=c_i$ (= communality of variable x_i) and $f_{\bullet r}=d_r$. The same property as with the classical chisquare criterion holds true; the criterion

$$(5)\qquad x_C=\sum_{i,r}\frac{f_{ir}^2}{f_{i\bullet}f_{\bullet r}}=\sum_{i,r}\frac{\lambda_{ir}^4}{c_i d_r}$$

becomes maximal if and only if all column sums $d_1,\dots,d_k$ are positive and each row of the matrix $\mathbf{F}=(f_{ir})$ contains only one element different from zero, and the maximum is given by $\max x_C=k$ (see Appendix A3). So the criterion x_C becomes maximal if and only if the loading matrix has an ideal simple structure and all column sums $d_1,\dots,d_k$ are positive. The shortcomings of quartimax (tendency to a general factor) and varimax (not always maximal for an ideal simple structure, tendency to factors with $d_1=\cdots=d_k$) are not present with the chisquare criterion. Thus we think that this criterion is a promising alternative to quartimax and varimax. Note that our findings concerning x_C are also true with oblique transformations where the row sums $f_{i\bullet}=c_i$ and the total sum $f_{\bullet\bullet}=\sum c_i$ are not fixed in general.

3. Determination of the quartimax solution

The derivations in this section are analogous to those for the varimax criterion to be found in Lawley and Maxwell (1971, pp 72). Let

$\mathbf{\Lambda_0} = (\ell_{ir}) = (p \times k) =$ matrix of unrotated loadings

$\mathbf{\Lambda}_0^{\mathsf{T}} = (\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_p) = (k \times p)$

$\mathbf{M} = (k \times k) = (\mathbf{m}_1, \ldots, \mathbf{m}_k)$ orthogonal rotation matrix

$\mathbf{\Lambda} = \mathbf{\Lambda}_0 \mathbf{M} = (\lambda_{ir}) = (k \times p) =$ matrix of rotated loadings

We have $\lambda_{ir} = \boldsymbol{\ell}_i^{\mathsf{T}} \mathbf{m}_r$. The quartimax criterion is given by

$$x_Q = \sum_{i,r} \lambda_{ir}^4 = \sum_{i=1}^{p} \sum_{r=1}^{k} (\boldsymbol{\ell}_i^{\mathsf{T}} \mathbf{m}_r)^4 .$$

We are looking for the maximum of x_Q under the side condition that $\mathbf{M} = (\mathbf{m}_1, \ldots, \mathbf{m}_k)$ is an orthogonal matrix. According to Lagrange's method we set

$$y = x_Q - 2 \sum_r \sum_s a_{rs} (\mathbf{m}_r^{\mathsf{T}} \mathbf{m}_s - \delta_{rs}) \quad \text{with} \quad \delta_{rs} = \begin{cases} 1 & \text{for } r = s \\ 0 & \text{for } r \neq s \end{cases}$$

where $A = (a_{rs}) = (k \times k)$ is the matrix of indeterminate multipliers with $a_{rs} = a_{sr}$. We have

$$\frac{\partial y}{\partial \mathbf{m}_s} = 4 \sum_i c_{is} \boldsymbol{\ell}_i - 4 \sum_r a_{rs} \mathbf{m}_r$$

where $c_{ir} = \lambda_{ir}^3$. Taking all values of s into account we have

$$\frac{\partial y}{\partial \mathbf{M}} = 4(\mathbf{B} - \mathbf{MA})$$

where

$$\mathbf{B} = \mathbf{\Lambda}_0^{\mathsf{T}} \mathbf{C} = (k \times k) \quad \text{with} \quad \mathbf{C} = (c_{ir}) = (\lambda_{ir}^3) = (p \times k) .$$

The condition $\partial y / \partial \mathbf{A} = 0$ is equivalent to the side condition that **M** has to be orthogonal.The orthogonal matrix **M** that maximizes x_Q thus satisfies the equation $\mathbf{MA} = \mathbf{B}$, and $\mathbf{A}$ has to be symmetric and positive definite. Premultiplying by $\mathbf{M}^{\mathsf{T}}$ gives $\mathbf{A} = \mathbf{M}^{\mathsf{T}} \mathbf{B} = \mathbf{\Lambda}^{\mathsf{T}} \mathbf{C}$ and we have $a_{rr} = \sum_i \lambda_{ir} c_{ir} = \sum_i \lambda_{ir}^4$ and so

$$trace(\mathbf{A}) = \sum_{r=1}^{k} a_{rr} = \sum_{i,r} \lambda_{ir}^4 = x_Q .$$

*Iterative procedure to determine **M**, **A**, and **B**:*

1. Start with $\mathbf{\Lambda}_0 = (\ell_{ir}) = (p \times k)$ and $\mathbf{M}_1 = (k \times k) = \mathbf{I}_k$.
2. Compute $\mathbf{\Lambda}_1 = \mathbf{\Lambda}_0 \mathbf{M}_1 = (\lambda_{ir}) = (p \times k)$, $\mathbf{C}_1 = (\lambda_{ir}^3) = (p \times k)$ and $\mathbf{B}_1 = \mathbf{\Lambda}_0^{\mathsf{T}} \mathbf{C}_1 = (k \times k)$.
3. Compute the singular value decomposition of $\mathbf{B}_1$: $\mathbf{B}_1 = \mathbf{U \Delta V}^{\mathsf{T}}$ where $\mathbf{U} = (k \times k)$ and $\mathbf{V} = (k \times k)$ are orthogonal, and where $\mathbf{\Delta} = diag(\delta_1, \ldots, \delta_k)$ with $\delta_r \geq 0$ for all r;
   set $\mathbf{A}_1 = \mathbf{V \Delta V}^{\mathsf{T}}$ and $\mathbf{M}_2 = \mathbf{UV}^{\mathsf{T}}$. $\mathbf{M}_2$ is orthogonal and $\mathbf{A}_1$ is symmetric and positive definite in the regular case that all $\delta_r > 0$, and we have $\mathbf{M}_2 \mathbf{A}_1 = (\mathbf{U V}^{\mathsf{T}})(\mathbf{V \Delta V}^{\mathsf{T}}) = \mathbf{U \Delta V}^{\mathsf{T}} = \mathbf{B}_1$.
4. Repeat the procedure (step 2 to 4) with $\mathbf{M}_2$ in place of $\mathbf{M}_1$.

The iterative procedure converges (generally) to a solution $\mathbf{M}, \mathbf{A}$ such that $\mathbf{MA} = \mathbf{B}$ with $\mathbf{M}$ orthogonal and **A** symmetric and positive definite; the sum of the singular values ($= trace(\mathbf{A}_1)$) then converges to the maximum of the quartimax criterion. See section 6 for modifications if this procedure fails to converge.

4. Determination of the varimax solution

The derivations in this section are to be found in Lawley and Maxwell (1971, pp 72). Let

$\mathbf{\Lambda_0} = (\ell_{ir}) = (p \times k) =$ matrix of unrotated loadings

$\mathbf{\Lambda}_0^\top = (\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_p) = (k \times p)$

$\mathbf{M} = (k \times k) = (\mathbf{m}_1, \ldots, \mathbf{m}_k)$ orthogonal rotation matrix

$\mathbf{\Lambda} = \mathbf{\Lambda}_0 \mathbf{M} = (\lambda_{ir}) = (k \times p) =$ matrix of rotated loadings

We have $\lambda_{ir} = \boldsymbol{\ell}_i^\top \mathbf{m}_r$. The varimax criterion is given by

$$x_V = \sum_r \left(\sum_i \lambda_{ir}^4 - d_r^2 / p \right) = \sum_r \left(\sum_i (\boldsymbol{\ell}_i^\top \mathbf{m}_r)^4 - d_r^2 / p \right).$$

We are looking for the maximum of x_V under the side condition that $\mathbf{M} = (\mathbf{m}_1, \ldots, \mathbf{m}_k)$ is an orthogonal matrix. According to Lagrange's method we set

$$y = x_V - 2 \sum_r \sum_s a_{rs} (\mathbf{m}_r^\top \mathbf{m}_s - \delta_{rs}) \quad \text{with} \quad \delta_{rs} = \begin{cases} 1 & \text{for } r = s \\ 0 & \text{for } r \neq s \end{cases}$$

where $A = (a_{rs}) = (k \times k)$ is the matrix of indeterminate multipliers with $a_{rs} = a_{sr}$. We have

$$\frac{\partial y}{\partial \mathbf{m}_s} = 4 \sum_i c_{is} \boldsymbol{\ell}_i - 4 \sum_r a_{rs} \mathbf{m}_r$$

where

$$c_{ir} = \lambda_{ir}^3 - \frac{d_r \lambda_{ir}}{p}.$$

Taking all values of *s* into account we have

$$\frac{\partial y}{\partial \mathbf{M}} = 4(\mathbf{B} - \mathbf{MA})$$

where

$$\mathbf{B} = \mathbf{\Lambda}_0^\top \mathbf{C} = (k \times k) \quad \text{with} \quad \mathbf{C} = (c_{ir}) = (p \times k).$$

The condition $\partial y / \partial \mathbf{A} = 0$ is equivalent to the side condition that **M** has to be orthogonal. The orthogonal matrix **M** that maximizes x_Q thus satisfies the equation $\mathbf{MA} = \mathbf{B}$, and $\mathbf{A}$ has to be symmetric and positive definite. Premultiplying by $\mathbf{M}^\top$ gives $\mathbf{A} = \mathbf{M}^\top \mathbf{B} = \mathbf{\Lambda}^\top \mathbf{C}$ and we have $a_{rr} = \sum_i \lambda_{ir} c_{ir} = \sum_i \lambda_{ir}^4 - d_r^2 / p$ and so

$$trace(\mathbf{A}) = \sum_r a_{rr} = \sum_r \left(\sum_i \lambda_{ir}^4 - d_r^2 / p \right) = x_V .$$

*Iterative procedure to determine **M**, **A**, and **B**:*

1. Start with $\mathbf{\Lambda}_0 = (\ell_{ir}) = (p \times k)$ and $\mathbf{M}_1 = (k \times k) = \mathbf{I}_k$.
2. Compute

   $\mathbf{\Lambda}_1 = \mathbf{\Lambda}_0 \mathbf{M}_1 = (\lambda_{ir}) = (p \times k),$

   $\mathbf{C}_1 = (c_{ir}) = (p \times k)$ with $c_{ir} = \lambda_{ir}^3 - d_r \lambda_{ir} / p,$

   $\mathbf{B}_1 = \mathbf{\Lambda}_0^\top \mathbf{C}_1 = (k \times k).$
3. Compute the singular value decomposition of $\mathbf{B}_1$: $\mathbf{B}_1 = \mathbf{U \Delta V}^\top$ where $\mathbf{U} = (k \times k)$ and $\mathbf{V} = (k \times k)$ are orthogonal, and where $\mathbf{\Delta} = diag(\delta_1, \ldots, \delta_k)$ with $\delta_r \geq 0$ for all *r*;

   set $\mathbf{A}_1 = \mathbf{V \Delta V}^\top$ and $\mathbf{M}_2 = \mathbf{UV}^\top$. $\mathbf{M}_2$ is orthogonal and $\mathbf{A}_1$ is symmetric and positive definite in the regular case that all $\delta_r > 0$, and we have $\mathbf{M}_2 \mathbf{A}_1 = (\mathbf{U V}^\top)(\mathbf{V \Delta V}^\top) = \mathbf{U \Delta V}^\top = \mathbf{B}_1$.

4. Repeat the procedure (step 2 to 4) with $\mathbf{M}_2$ in place of $\mathbf{M}_1$.

The iterative procedure converges (generally) to a solution $\mathbf{M}, \mathbf{A}$ such that $\mathbf{MA} = \mathbf{B}$ with $\mathbf{M}$ orthogonal and $\mathbf{A}$ symmetric and positive definite; the sum of the singular values ($= trace(\mathbf{A}_1)$) then converges to the maximum of the varimax criterion. See section 6 for modifications if this procedure fails to converge.

Remarks:

a) In Lawley and Maxwell (1971, pp 72) the eigenvalue decomposition (spectral decomposition) of the symmetric matrix $\mathbf{B}_1^\mathsf{T}\mathbf{B}_1$ is used instead of the singular value decomposition of $\mathbf{B}_1$ which is implemented in the varimax procedure in R. In my view the method of R with the singular value decomposition makes the solution simpler.

b) With Kaiser-normalization the following steps are performed:

(i) Normalize $\mathbf{\Lambda}_0$: $\tilde{\mathbf{\Lambda}}_0 = \mathbf{D}^{-1}\mathbf{\Lambda}_0$ with $D = diag\left(\sqrt{c_1}, \ldots, \sqrt{c_k}\right)$.

(ii) Determine the optimal rotation $\mathbf{M}$ according to the above procedure: $\tilde{\mathbf{\Lambda}} = \tilde{\mathbf{\Lambda}}_0\mathbf{M}$.

(iii) Restore the original communalities: $\mathbf{\Lambda} = \mathbf{D}\tilde{\mathbf{\Lambda}} = \mathbf{D}\tilde{\mathbf{\Lambda}}_0\mathbf{M} = \mathbf{D}\mathbf{D}^{-1}\mathbf{\Lambda}_0\mathbf{M} = \mathbf{\Lambda}_0\mathbf{M}$.

5. Determination of the chisquaremax solution

The derivations in this section are again analogous to those for the varimax criterion to be found in Lawley and Maxwell (1971, pp 72). Let

$\mathbf{\Lambda_0} = (\ell_{ir}) = (p \times k) =$ matrix of unrotated loadings,

$\mathbf{\Lambda}_0^\mathsf{T} = (\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_p) = (k \times p)$,

$\mathbf{M} = (k \times k) = (\mathbf{m}_1, \ldots, \mathbf{m}_k)$ orthogonal rotation matrix,

$\mathbf{\Lambda} = \mathbf{\Lambda}_0\mathbf{M} = (\lambda_{ir}) = (k \times p) =$ matrix of rotated loadings.

We have $\lambda_{ir} = \boldsymbol{\ell}_i^\mathsf{T}\mathbf{m}_r$. The chisquaremax criterion is given by

$$x_C = \sum_{i,r}\frac{\lambda_{ir}^4}{c_i d_r} = \sum_{i,r}\frac{(\boldsymbol{\ell}_i^\mathsf{T}\mathbf{m}_r)^4}{c_i d_r} \quad \text{where} \quad c_i = \sum_{r=1}^{k}\lambda_{ir}^2 \quad \text{and} \quad d_r = \sum_{i=1}^{p}\lambda_{ir}^2 .$$

We are looking for the maximum of x_C under the side condition that $\mathbf{M} = (\mathbf{m}_1, \ldots, \mathbf{m}_k)$ is an orthogonal matrix. According to Lagrange's method we set

$$y = x_C - 2\sum_r\sum_s a_{rs}(\mathbf{m}_r^\mathsf{T}\mathbf{m}_s - \delta_{rs}) \quad \text{with} \quad \delta_{rs} = \begin{cases} 1 & \text{for } r = s \\ 0 & \text{for } r \neq s \end{cases}$$

where $A = (a_{rs}) = (k \times k)$ is the matrix of indeterminate multipliers with $a_{rs} = a_{sr}$. We have

$$\frac{\partial y}{\partial \mathbf{m}_s} = \frac{\partial x_C}{\partial \mathbf{m}_s} - 4\sum_r a_{rs}\mathbf{m}_r .$$

Now

$$\frac{\partial x_C}{\partial \mathbf{m}_s} = \sum_{i,r}\frac{1}{(c_i d_r)^2}\left(c_i d_r \frac{\partial}{\partial \mathbf{m}_s}(\boldsymbol{\ell}_i^\mathsf{T}\mathbf{m}_r)^4 - \lambda_{ir}^4 \frac{\partial}{\partial \mathbf{m}_s}(c_i d_r)\right)$$

$$= \sum_i \frac{1}{c_i d_s}\frac{\partial}{\partial \mathbf{m}_s}(\boldsymbol{\ell}_i^\mathsf{T}\mathbf{m}_s)^4 - \frac{e_s}{d_s^2}\frac{\partial d_s}{\partial \mathbf{m}_s} \quad \text{with} \quad e_s = \sum_i \frac{\lambda_{is}^4}{c_i}$$

and

$$\frac{\partial}{\partial \mathbf{m}_s}(\boldsymbol{\ell}_i^\mathsf{T}\mathbf{m}_s)^4 = 4(\boldsymbol{\ell}_i^\mathsf{T}\mathbf{m}_s)^3\boldsymbol{\ell}_i = 4\lambda_{is}^3\boldsymbol{\ell}_i$$

$$\frac{\partial d_s}{\partial \mathbf{m}_s} = \frac{\partial}{\partial \mathbf{m}_s}\sum\nolimits_i \lambda_{is}^2 = \frac{\partial}{\partial \mathbf{m}_s}\sum\nolimits_i (\boldsymbol{\ell}_i^\mathsf{T}\mathbf{m}_s)^2 = 2\sum\nolimits_i \lambda_{is}\boldsymbol{\ell}_i .$$

So we obtain

$$\frac{\partial x_C}{\partial \mathbf{m}_s} = \sum_i \frac{4\lambda_{is}^3}{c_i d_s}\boldsymbol{\ell}_i - 2\frac{e_s}{d_s^2}\sum_i \lambda_{is}\boldsymbol{\ell}_i \;=\; \left(\frac{4}{d_s}\sum_i \frac{\lambda_{is}^3}{c_i} - \frac{2e_s}{d_s^2}\sum_i \lambda_{is}\right)\boldsymbol{\ell}_i .$$

Taking all values of s into account we have

$$\frac{\partial y}{\partial \mathbf{M}} = 4(\mathbf{B} - \mathbf{MA})$$

where

$$\mathbf{B} = \mathbf{\Lambda}_0^\mathsf{T}\mathbf{C} = (k\times k) \quad \text{with} \quad \mathbf{C} = (c_{ir}) = (p\times k) \text{ and } c_{ir} = \frac{\lambda_{ir}^3}{c_i d_r} - \frac{e_r}{2d_r^2}\lambda_{ir} .$$

The condition $\partial y/\partial \mathbf{A} = 0$ is equivalent to the side condition that $\mathbf{M}$ has to be orthogonal. The orthogonal matrix $\mathbf{M}$ that maximizes x_C thus satisfies the equation $\mathbf{MA} = \mathbf{B}$, and $\mathbf{A}$ has to be symmetric and positive definite. Premultiplying by $\mathbf{M}^\mathsf{T}$ gives $\mathbf{A} = \mathbf{M}^\mathsf{T}\mathbf{B} = \mathbf{\Lambda}^\mathsf{T}\mathbf{C}$ and we have

$$a_{rr} = \sum_i \lambda_{ir}c_{ir} = \sum_i \frac{\lambda_{ir}^4}{c_i d_r} - \frac{1}{2d_r^2}\left(\sum_j \frac{\lambda_{jr}^4}{c_j}\right)\sum_i \lambda_{ir}^2 = \tfrac{1}{2}\sum_i \frac{\lambda_{ir}^4}{c_i d_r}$$

and so

$$trace(\mathbf{A}) = \sum_r a_{rr} \;=\; \tfrac{1}{2}\sum_{i,r}\frac{\lambda_{ir}^4}{c_i d_r} \;=\; \tfrac{1}{2}x_C .$$

*Iterative procedure to determine **M**, **A**, and **B**:*

1. Start with $\mathbf{\Lambda}_0 = (\ell_{ir}) = (p\times k)$ and $\mathbf{M}_1 = (k\times k) = \mathbf{I}_k$.
2. Compute

   $$\mathbf{\Lambda}_1 = \mathbf{\Lambda}_0\mathbf{M}_1 = (\lambda_{ir}) = (p\times k),$$

   $$\mathbf{C}_1 = (c_{ir}) = (p\times k) \text{ with } c_{ir} = \frac{\lambda_{ir}^3}{c_i d_r} - \frac{1}{2d_r^2}\left(\sum_j \frac{\lambda_{jr}^4}{c_j}\right)\lambda_{ir},$$

   $$\mathbf{B}_1 = \mathbf{\Lambda}_0^\mathsf{T}\mathbf{C}_1 = (k\times k).$$

3. Compute the singular value decomposition of $\mathbf{B}_1$: $\mathbf{B}_1 = \mathbf{U\Delta V}^\mathsf{T}$ where $\mathbf{U} = (k\times k)$ and $\mathbf{V} = (k\times k)$ are orthogonal, and where $\mathbf{\Delta} = diag(\delta_1,\ldots,\delta_k)$ with $\delta_r \ge 0$ for all r;
   set $\mathbf{A}_1 = \mathbf{V\Delta V}^\mathsf{T}$ and $\mathbf{M}_2 = \mathbf{UV}^\mathsf{T}$. $\mathbf{M}_2$ is orthogonal and $\mathbf{A}_1$ is symmetric and positive definite in the regular case that all $\delta_r > 0$, and we have $\mathbf{M}_2\mathbf{A}_1 = (\mathbf{U}\,\mathbf{V}^\mathsf{T})(\mathbf{V\Delta V}^\mathsf{T}) = \mathbf{U\Delta V}^\mathsf{T} = \mathbf{B}_1$.
4. Repeat the procedure (step 2 to 4) with $\mathbf{M}_2$ in place of $\mathbf{M}_1$ until convergence takes place.

The iterative procedure converges (generally) to a solution $\mathbf{M}, \mathbf{A}$ such that $\mathbf{MA} = \mathbf{B}$ with $\mathbf{M}$ orthogonal and $\mathbf{A}$ symmetric and positive definite; the sum of the singular values ($= trace(\mathbf{A}_1)$) then converges to $x_C/2$ where x_C is the maximum of the chisquare criterion. See section 6 for modifications if this procedure fails to converge.

6. Modification if the algorithm fails to converge

It can happen that the iterative procedures described in the foregoing sections fail to converge. In order to show the problem and to describe a helpful modification we consider the algorithm for the chisquaremax solution in greater detail.

Input: $\Lambda_0 = (p \times k)$, $\varepsilon =$ bound for relative accuracy (e.g. $\varepsilon = 10^{-9}$).

Output:

$\mathbf{M} = (k \times k)$ orthogonal (= optimal rotation matrix),

$\mathbf{\Lambda} = \mathbf{\Lambda}_0 \mathbf{M} = (p \times k)$ such that the chisquare criterion for $\mathbf{\Lambda}$ becomes maximal,

iter = number of necessery iterations to find the solution.

Algorithm:

1. Start with $\mathbf{\Lambda}_0 = (\ell_{ir}) = (p \times k)$, $\mathbf{M}_1 = (k \times k) = \mathbf{I}_k$ and $\mathbf{\Lambda}_1 = \mathbf{\Lambda}_0 \mathbf{M}_1 = (\lambda_{ir}) = (p \times k)$;

   set $x_1^{\text{old}} = 0,\ x_2^{\text{old}} = 0$.

2. Compute

   $$\mathbf{C}_1 = (c_{ir}) = (p \times k) \text{ with } c_{ir} = \frac{\lambda_{ir}^3}{c_i d_r} - \frac{1}{2d_r^2}\left(\sum_{j=1}^{p} \frac{\lambda_{jr}^4}{c_j}\right)\lambda_{ir},\ c_i = \sum_{r=1}^{k} \lambda_{ir}^2,\ \ d_r = \sum_{i=1}^{p} \lambda_{ir}^2,$$

   $$\mathbf{B}_1 = \mathbf{\Lambda}_0^{\mathsf{T}} \mathbf{C}_1 = (k \times k).$$

3. Compute the singular value decomposition of $\mathbf{B}_1$: $\mathbf{B}_1 = \mathbf{U}\mathbf{\Delta}\mathbf{V}^{\mathsf{T}}$ where $\mathbf{U} = (k \times k)$ and $\mathbf{V} = (k \times k)$ are orthogonal, and where $\mathbf{\Delta} = diag(\delta_1, \ldots, \delta_k)$ with $\delta_r \geq 0$ for all *r*.

   Set $\mathbf{A}_1 = \mathbf{V}\mathbf{\Delta}\mathbf{V}^{\mathsf{T}}$ and $\mathbf{M}_2 = \mathbf{U}\mathbf{V}^{\mathsf{T}}$; $\mathbf{M}_2$ is orthogonal and $\mathbf{A}_1$ is symmetric and positive definite in the regular case that all $\delta_r > 0$, and we have $\mathbf{M}_2 \mathbf{A}_1 = (\mathbf{U}\mathbf{V}^{\mathsf{T}})(\mathbf{V}\mathbf{\Delta}\mathbf{V}^{\mathsf{T}}) = \mathbf{U}\mathbf{\Delta}\mathbf{V}^{\mathsf{T}} = \mathbf{B}_1$.

4. Compute

   $\mathbf{\Lambda}_2 = \mathbf{M}_2 \mathbf{\Lambda}_0$,

   $x_1 = 2 \times trace(\mathbf{A}_1)$,

   $x_2 = x_C(\mathbf{\Lambda}_2) =$ chisquare criterion x_C for the matrix $\mathbf{\Lambda}_2$.

   If $\dfrac{\left|x_1 - x_1^{\text{old}}\right|}{x_1} < \varepsilon$ and $\dfrac{\left|x_2 - x_2^{\text{old}}\right|}{x_2} < \varepsilon$ and $\dfrac{|x_1 - x_2|}{x_1} < \varepsilon$ then stop the iteration procedure.

5. Set $x_1^{\text{old}} = x_1,\ x_2^{\text{old}} = x_2$, replace $\mathbf{\Lambda}_1$ by $\mathbf{\Lambda}_2$ and continue with step 2.

When the iteration procedure stops, set

$\mathbf{M} = \mathbf{M}_2$, $\mathbf{\Lambda} = \mathbf{\Lambda}_2$, *iter* = number of iterations.

Now we give an example of the foregoing algorithm. Let

$$\text{(6)} \qquad \mathbf{\Lambda}_0 = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 0.9 & 0 & 0.3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and set the bound for the relative accuracy to $\varepsilon = 10^{-9}$. Then we find the results given in Table 1. This table shows that the algorithm finally flutters between two positions, the first one with $(x_1, x_2) =$ (2.747..., 2.649...) and the second one with $(x_1, x_2) =$ (2.750..., 2.645...). Thus the algorithm will never converge for the given relative accuracy of $\varepsilon = 10^{-9}$ without some modification.

Table 1: Values of $x_1 = 2 \times trace(\mathbf{A})$ and $x_2 = x_C(\mathbf{\Lambda})$ in the first 500 iterations with $\varepsilon = 10^{-9}$

iter	x_1	x_2
1	2.749822213	2.730058156
2	2.751688844	2.730047592
3	2.750902428	2.731099610
4	2.752166652	2.729093874
5	2.751454734	2.728168484
6	2.752553821	2.724443517
7	2.751783373	2.722048335
8	2.752881199	2.716717826
9	2.751924311	2.713262696
10	2.753096960	2.706562010
⋮	⋮	⋮
100	2.750255585	2.645655537
101	2.747389294	2.649464918
102	2.750255585	2.645655537
103	2.747389294	2.649464918
104	2.750255585	2.645655537
105	2.747389294	2.649464918
106	2.750255585	2.645655537
107	2.747389294	2.649464918
108	2.750255585	2.645655537
109	2.747389294	2.649464918
110	2.750255585	2.645655537
⋮	⋮	⋮
490	2.750255585	2.645655537
491	2.747389294	2.649464918
492	2.750255585	2.645655537
493	2.747389294	2.649464918
494	2.750255585	2.645655537
495	2.747389294	2.649464918
496	2.750255585	2.645655537
497	2.747389294	2.649464918
498	2.750255585	2.645655537
499	2.747389294	2.649464918
500	2.750255585	2.645655537
⋮	⋮	⋮

We can overcome this problem by reducing the step width in the iterative procedure. Instead of

5. Set $x_1^{\text{old}} = x_1,\ x_2^{\text{old}} = x_2$, replace $\mathbf{\Lambda}_1$ by $\mathbf{\Lambda}_2$, and continue with step 2.

we use

5. Set $x_1^{\text{old}} = x_1,\ x_2^{\text{old}} = x_2$, replace $\mathbf{\Lambda}_1$ by $\gamma\mathbf{\Lambda}_2 + (1-\gamma)\mathbf{\Lambda}_1$ with $0 < \gamma \le 1$, and continue with step 2.

For $\gamma = 1$ we have the old procedure, but by choosing $\gamma < 1$ we may be able to eliminate fluttering of the algorithm. We now try to find the solution for the above example with $\gamma = 0.5$. Table 2 shows that now the algorithm achieves the desired relative accuracy of $\varepsilon = 10^{-9}$ in 24 iterations.

Table 2: Values of $x_1 = 2 \times trace(\mathbf{A})$ and $x_2 = x_C(\mathbf{\Lambda})$ with $\varepsilon = 10^{-9}$ and $\gamma = 0.5$

iter	x_1	x_2
1	2.749822213	2.730058156
2	2.745603243	2.751546198
3	2.748602501	2.751663762
4	2.750133293	2.751664495
5	2.750898898	2.751664514
6	2.751281706	2.751664515
7	2.751473111	2.751664515
8	2.751568813	2.751664515
9	2.751616664	2.751664515
10	2.751640590	2.751664515
11	2.751652552	2.751664515
12	2.751658534	2.751664515
13	2.751661524	2.751664515
14	2.751663020	2.751664515
15	2.751663767	2.751664515
16	2.751664141	2.751664515
17	2.751664328	2.751664515
18	2.751664422	2.751664515
19	2.751664468	2.751664515
20	2.751664492	2.751664515
21	2.751664503	2.751664515
22	2.751664509	2.751664515
23	2.751664512	2.751664515
24	2.751664514	2.751664515

In Table 3 the the results of the chisquaremax solution are presented together with the corresponding results for the quartimax and varimax solution. The chisquaremax procedure achieves the maximum 2.752 of the chisquare criterion x_C in 24 iterations with $\gamma = 0.5$, the quartimax procedure achieves the maximum 4.364 of the quartimax criterion x_Q (with Kaiser normalization) in 9 iterations (with the standard value $\gamma = 1$), and the varimax procedurer achieves the maximum 2.508 of the varimax criterion x_V (with Kaiser normalization) in 11 iterations (with the standard value $\gamma = 1$). One can see that the resulting loading matrix is approximately the same for all three procedures. Programs in Maple (very detailed) and R that compute the given solutions are to be found in the Appendix A4 and A5.

Table 3: Quartimax, varimax and chisquaremax solution for inital loading matrix (6)

	before rotation			after rotation		
	Id.	loadings	c_i	Quartimax	Varimax	Chisquaremax
Loading matrix	1 2 3 4 5	0.5 0.5 0.0 0.9 0.0 0.3 0.0 1.0 0.0 0.0 0.0 1.0 1.0 0.0 0.0	0.5 0.9 1.0 1.0 1.0	0.497 0.501 -0.041 0.924 -0.000 0.216 -0.002 1.000 0.010 0.092 -0.010 0.996 0.996 0.003 -0.091	0.473 0.525 -0.031 0.918 0.044 0.235 -0.050 0.999 0.010 0.072 -0.006 0.997 0.996 0.051 -0.071	0.496 0.502 -0.040 0.922 0.002 0.222 -0.004 1.000 0.006 0.085 -0.005 0.996 0.996 0.004 -0.085
d_1, d_2, d_3		2.06 1.25 1.09	4.4	2.100 1.251 1.048	2.067 1.277 1.056	2.097 1.252 1.051
Quartimax criterion	4.320			4.364 (9)	4.354	4.364
Varimax criterion	2.476			2.498	2.508 (11)	2.499
Chisquare criterion	2.726			2.751	2.745	2.752 (24/0.5)

Comments:

Quartimax: With $\gamma = 1$ monotonous convergence from below for $trace(\mathbf{A})$ and $x_Q(\mathbf{\Lambda})$ in 9 iterations.

Varimax: With $\gamma = 1$ monotonous convergence from below for $trace(\mathbf{A})$ and $x_V(\mathbf{\Lambda})$ in 11 iterations.

Chisquare: With $\gamma = 1$ no convergence (see Table 1);
with $\gamma = 0.5$ monotonous convergence from below for $trace(\mathbf{A})$ and $x_C(\mathbf{\Lambda})$ (beginning with iteration 2) in 24 iterations (see Table 2).

The quartimax and chisquare solutions are nearly identical, but the varimax solution is also not very different.

7. Examples

a) Eight physical variables (cf. Harman, 1976, p.22 and p.254)
In this example the following eight variables of 305 girls from seven to seventeen years of age are measurred:

1. Height
2. Arm span
3. Length of forearm
4. Length of lower leg
5. Weight
6. Bitrochanteric diameter
7. Chest girth
8. Chest width

We consider the factorial model with two factors. Table 4 gives the results with the three rotation procedures. We can see that the quartimax and varimax solutions are nearly identical, but also the chisquare solution is not very different.

b) Box problem of Thurstone (cf. Thurstone, 1947, p. 369-371)
Measurements of a random collection of thirty boxes were made; the three dimensions x, y, z were recorded for each box, and a list of 26 score functions (variables) was then prepared:

i	Variable i	i	Variable i
1	x	14	y/x
2	y	15	x/z
3	z	16	z/x
4	xy	17	y/z
5	xz	18	z/y
6	yz	19	$2x+2y$
7	x^2y	20	$2x+2z$
8	xy^2	21	$2y+2z$
9	x^2z	22	$\sqrt{x^2+y^2}$
10	xz^2	23	$\sqrt{x^2+z^2}$
11	y^2z	24	$\sqrt{y^2+z^2}$
12	yz^2	25	xyz
13	x/y	26	$\sqrt{x^2+y^2+z^2}$

For example, variable 4 consisted of the area xy of one side of the box. As in the classical model of factor analysis (see section 1) the variables are connected with the unknown factors in a linear way, and as only a few of the variables in our example are linearly connected with x, y, z we cannot expect to find a rotation procedure that explains our 26 variables by the obvious physical factors x, y, z. Table 5 shows that the quartimax and the chisquare solution are nearly identical, but also the varimax solution is not very different.

c) *Twenty-four psychological tests* (cf. Harman, 1976, p.123 and p.215)
Twenty-four psychological tests were given to 145 seventh and eighth grade school children in a suburb of Chicago. Here is the list of these tests (variables):

i	Variable *i*	*i*	Variable *i*
1	Visual Perception	13	Straight-Curved Capitals
2	Cubes	14	Word Recognition
3	Paper Form Board	15	Number Recognition
4	Flags	16	Figure Recognition
5	General Information	17	Object-Number
6	Paragraph Comprehension	18	Number-Figure
7	Sentence Completion	19	Figure-Word
8	Word Classification	20	Deduction
9	Word Meaning	21	Numerical Puzzles
10	Addition	22	Problem Reasoning
11	Code	23	Series Completion
12	Counting Dots	24	Arithmetic Problems

We consider the factorial model with four factors. Table 6 shows that the chisquare solution is close to the varimax solution; the quartimax solution shows a clear tendency to a general factor ($d_1 = 6.511$ clearly larger than d_2, d_3, d_4).

d) *Thirteen psychological tests* (cf. Harman, 1976, p.172)
The first thirteen of the twenty-four psychological tests (see the foregoing example) are considered here, and the factorial model takes into account only three factors. Table 7 shows that the chisquare solution is close to the quartimax solution, but also the varimax solution is not very different.

e) *Change of scale* (cf. Hechenbichler, 1999, Veränderungsskalen-Datensatz, pp 140-150 and 166-196)
A questionnaire with 32 items is given to 165 people addicted to drugs, and a factorial model with four factors is considered. Table 8 shows that the three solutions are very similar.

f) *Jealousy data* (cf. Hechenbichler, 1999, Eifersuchtsdaten, pp 150-162 and 197-211)
A questionnaire with 39 items is given to 141 people from fifteen to fourty years of age, and a factorial model with eight factors is considered. Table 9 shows that the chisquare solution is similar to the varimax solution; the quartimax solution shows a clear tendency to a general factor ($d_1 = 6.609$ clearly larger than $d_2, \ldots, d_8$).

8. Conclusion

On the basis of our theoretical considerations in section 2 we expect that the quartimax solution (with and without Kaiser normalization) can show a tendency to a general factor whereas the varimax solution (with and without Kaiser normalization) can show a tendency to homogenous factors (measured with the colums sums $d_1, \ldots, d_k$). The chisquare solution does not show this asymmetry, and our examples confirm this conjecture although the difference between the three solutions is often very small. The nice properties of the criterion x_C remain unchanged with oblique transformations. So we can expect that this criterion will be a good alternative to the well known criteria not only with orthogonal but also with oblique transformations.

Table 4: Eight Physical Variables (cf. Harman, 1976, p.254)

	before rotation				after rotation					
	Id.	loadings		c_i	Quartimax		Varimax		Chisquaremax	
Loading matrix	1	0.853	0.332	0.838	0.873	0.275	0.872	0.278	0.882	0.245
	2	0.906	0.261	0.889	0.921	0.201	0.921	0.204	0.927	0.169
	3	0.874	0.237	0.820	0.888	0.179	0.887	0.182	0.893	0.149
	4	0.846	0.302	0.807	0.864	0.246	0.863	0.248	0.872	0.216
	5	0.175	0.926	0.888	0.235	0.913	0.233	0.913	0.266	0.904
	6	0.140	0.788	0.641	0.191	0.777	0.189	0.778	0.218	0.770
	7	0.082	0.760	0.584	0.132	0.753	0.130	0.753	0.157	0.748
	8	0.216	0.667	0.492	0.259	0.651	0.257	0.652	0.281	0.642
d_1, d_2, d_3		3.132	2.827	5.958	3.322	2.636	3.313	2.645	3.418	2.540
Quartimax criterion	6.981				7.016 (11)		7.016		7.007	
Varimax criterion	2.969				3.016		3.016 (25/0.5)		3.000	
Chisquare criterion	1.738				1.758		1.758		1.761 (10)	

Comments:

Quartimax: With $\gamma = 1$ monotonous convergence from below for $trace(\mathbf{A})$ and $x_Q(\mathbf{\Lambda})$ in 11 iterations.

Varimax: With $\gamma = 1$ monotonous convergence from below for $trace(\mathbf{A})$ and $x_V(\mathbf{\Lambda})$ in 273 iterations with $\gamma = 0.5$ monotonous convergence from below for $trace(\mathbf{A})$ and $x_V(\mathbf{\Lambda})$ in 25 iterations.

Chisquare: With $\gamma = 1$ monotonous convergence from below for $trace(\mathbf{A})$ and $x_C(\mathbf{\Lambda})$ in 10 iterations.

The quartimax and varimax solutions are nearly identical, but also the chisquare solution is not very different.

Table 5: Box problem of Thurstone (cf. Thurstone, 1947, p.371)

	before rotation					after rotation								
	Id.	loadings			c_i	Quartimax			Varimax			Chisquaremax		
Loading matrix	1	0.65	-0.67	0.33	0.98	0.60	-0.29	0.73	0.53	-0.35	0.76	0.58	-0.30	0.74
	2	0.74	0.53	0.37	0.97	0.77	0.61	-0.05	0.82	0.54	0.00	0.78	0.59	-0.04
	3	0.75	0.06	-0.64	0.98	0.76	-0.42	-0.47	0.74	-0.49	-0.43	0.76	-0.44	-0.45
	4	0.87	-0.04	0.48	0.99	0.86	0.27	0.42	0.85	0.18	0.48	0.86	0.25	0.44
	5	0.88	-0.40	-0.24	0.99	0.86	-0.49	0.14	0.80	-0.57	0.18	0.84	-0.51	0.16
	6	0.89	0.41	-0.20	1.00	0.92	0.13	-0.38	0.95	0.04	-0.32	0.93	0.10	-0.36
	7	0.84	-0.35	0.43	1.00	0.81	0.01	0.59	0.77	-0.08	0.64	0.80	-0.01	0.61
	8	0.86	0.22	0.43	0.97	0.87	0.42	0.21	0.89	0.33	0.27	0.87	0.40	0.22
	9	0.83	-0.55	-0.03	0.99	0.80	-0.45	0.39	0.72	-0.53	0.43	0.77	-0.47	0.41
	10	0.85	-0.26	-0.44	0.98	0.84	-0.52	-0.10	0.79	-0.60	-0.06	0.83	-0.54	-0.08
	11	0.86	0.49	-0.01	0.98	0.89	0.32	-0.30	0.93	0.23	-0.24	0.90	0.30	-0.28
	12	0.87	0.29	-0.38	0.99	0.89	-0.08	-0.43	0.90	-0.16	-0.38	0.90	-0.10	-0.41
	13	-0.07	-0.98	-0.09	0.97	-0.13	-0.77	0.60	-0.24	-0.76	0.58	-0.16	-0.77	0.60
	14	0.07	0.98	0.09	0.97	0.13	0.77	-0.60	0.24	0.76	-0.58	0.16	0.77	-0.60
	15	-0.05	-0.55	0.80	0.94	-0.09	0.15	0.96	-0.14	0.14	0.95	-0.11	0.15	0.95
	16	0.05	0.55	-0.80	0.94	0.09	-0.15	-0.96	0.14	-0.14	-0.95	0.11	-0.15	-0.95
	17	0.00	0.49	0.85	0.96	0.02	0.94	0.29	0.09	0.93	0.30	0.04	0.94	0.28
	18	0.00	-0.49	-0.85	0.96	-0.02	-0.94	-0.29	-0.09	-0.93	-0.30	-0.04	-0.94	-0.28
	19	0.86	0.05	0.48	0.97	0.86	0.33	0.36	0.86	0.24	0.42	0.86	0.31	0.38
	20	0.87	-0.39	-0.32	1.00	0.85	-0.54	0.07	0.79	-0.62	0.12	0.83	-0.56	0.10
	21	0.90	0.40	-0.19	1.00	0.92	0.13	-0.37	0.95	0.04	-0.31	0.94	0.10	-0.35
	22	0.85	0.05	0.47	0.95	0.85	0.33	0.35	0.85	0.24	0.41	0.85	0.31	0.37
	23	0.86	-0.34	-0.32	0.96	0.84	-0.50	0.04	0.79	-0.58	0.08	0.83	-0.52	0.06
	24	0.89	0.39	-0.16	0.97	0.91	0.14	-0.34	0.94	0.06	-0.28	0.92	0.12	-0.32
	25	0.99	-0.01	0.01	0.98	0.99	-0.04	0.06	0.97	-0.13	0.12	0.98	-0.06	0.09
	26	0.96	0.10	-0.02	0.93	0.96	0.02	-0.04	0.96	-0.07	0.02	0.97	-0.00	-0.01
d_1, d_2, d_3		14.69	5.53	5.15	25.37	14.69	5.52	5.16	14.52	5.64	5.21	14.66	5.53	5.18
Quartimax criterion	16.988					17.381 (83)			17.277			17.372		
Varimax criterion	6.011					6.398			6.417 (77)			6.407		
Chisquare criterion	1.757					1.822			1.820			1.823 (48)		

Comments:

With $\gamma = 1$ monotonous convergence from below for $trace(\mathbf{A})$ and the criteria x_Q, x_V, x_C for all three procedures.

The quartimax and chisquare solutions are nearly identical, but also the varimax solution is not very different.

Table 6: Twenty-four psychological tests (cf. Harman, 1976, p.215)

	before rotation			after rotation		
	Id	loadings	c_i	Quartimax	Varimax	Chisquaremax
Loading matrix	1	0.601 0.019 0.388 0.221	0.561	0.730 -0.123 -0.029 -0.107	0.159 0.689 0.187 0.160	0.172 0.714 0.108 0.102
	2	0.372 -0.025 0.252 0.132	0.220	0.457 -0.060 -0.053 -0.070	0.117 0.436 0.083 0.097	0.124 0.447 0.033 0.060
	3	0.413 -0.117 0.388 0.144	0.356	0.551 -0.082 -0.190 -0.097	0.135 0.570 -0.019 0.109	0.143 0.569 -0.083 0.065
	4	0.487 -0.100 0.254 0.192	0.349	0.573 0.017 -0.068 -0.123	0.233 0.527 0.099 0.079	0.240 0.537 0.036 0.030
	5	0.691 -0.304 -0.279 0.035	0.649	0.532 0.591 0.123 0.033	0.739 0.185 0.214 0.150	0.750 0.211 0.184 0.089
	6	0.690 -0.409 -0.200 -0.076	0.689	0.547 0.614 -0.026 0.115	0.767 0.204 0.067 0.234	0.781 0.219 0.037 0.173
	7	0.677 -0.409 -0.292 0.084	0.718	0.525 0.661 0.063 -0.039	0.806 0.196 0.154 0.075	0.812 0.210 0.121 0.011
	8	0.674 -0.189 -0.099 0.122	0.515	0.598 0.380 0.112 -0.032	0.570 0.338 0.242 0.132	0.580 0.367 0.197 0.070
	9	0.697 -0.454 -0.212 -0.080	0.743	0.549 0.654 -0.050 0.110	0.806 0.200 0.042 0.227	0.819 0.212 0.011 0.165
	10	0.476 0.534 -0.486 0.092	0.756	0.246 0.118 0.816 0.127	0.169 -0.117 0.829 0.166	0.189 -0.014 0.837 0.141
	11	0.558 0.332 -0.142 -0.090	0.450	0.430 0.050 0.439 0.264	0.180 0.119 0.513 0.374	0.209 0.199 0.500 0.341
	12	0.472 0.508 -0.139 0.256	0.566	0.404 -0.119 0.622 -0.045	0.019 0.210 0.717 0.087	0.035 0.294 0.689 0.054
	13	0.602 0.244 0.028 0.295	0.510	0.599 -0.027 0.369 -0.120	0.188 0.437 0.526 0.082	0.200 0.496 0.472 0.029
	14	0.423 0.058 0.015 -0.415	0.355	0.323 0.107 0.040 0.487	0.198 0.050 0.081 0.554	0.232 0.093 0.082 0.535
	15	0.394 0.089 0.097 -0.362	0.304	0.336 0.017 0.018 0.436	0.122 0.116 0.074 0.519	0.154 0.156 0.068 0.501
	16	0.510 0.095 0.347 -0.249	0.452	0.556 -0.135 -0.077 0.345	0.068 0.409 0.062 0.526	0.101 0.446 0.023 0.492
	17	0.466 0.197 -0.004 -0.381	0.401	0.357 0.038 0.170 0.494	0.143 0.062 0.219 0.573	0.180 0.122 0.219 0.553
	18	0.515 0.312 0.152 -0.147	0.407	0.496 -0.149 0.223 0.300	0.026 0.293 0.336 0.456	0.058 0.358 0.309 0.425
	19	0.443 0.089 0.109 -0.150	0.239	0.419 0.007 0.070 0.242	0.149 0.239 0.162 0.365	0.173 0.278 0.138 0.335
	20	0.614 -0.118 0.126 -0.038	0.408	0.603 0.172 -0.024 0.120	0.377 0.402 0.118 0.300	0.397 0.428 0.072 0.249
	21	0.589 0.227 0.057 0.123	0.417	0.571 -0.026 0.297 0.038	0.174 0.380 0.438 0.222	0.194 0.439 0.394 0.176
	22	0.608 -0.107 0.127 -0.038	0.399	0.598 0.162 -0.018 0.121	0.366 0.399 0.123 0.301	0.386 0.426 0.077 0.250
	23	0.687 -0.044 0.138 0.098	0.503	0.694 0.126 0.070 0.019	0.369 0.500 0.244 0.238	0.387 0.536 0.185 0.178
	24	0.651 0.177 -0.212 -0.017	0.500	0.502 0.222 0.409 0.179	0.371 0.157 0.496 0.304	0.395 0.228 0.475 0.258
$d_1,\ldots,d_4$		7.645 1.681 1.228 0.911	11.464	6.511 1.967 1.813 1.173	3.649 2.870 2.657 2.288	3.886 3.359 2.358 1.861
Quartimax criterion		13.597		14.928 (47)	14.271	14.357
Varimax criterion		1.802		4.851	8.189 (10)	7.991
Chisquare criterion		1.465		2.190	2.419	2.441 (13)

Comments:

With $\gamma = 1$ monotonous convergence from below for $trace(\mathbf{A})$ and the criteria x_Q, x_V, x_C for all three procedures.

The chisquare solution is close to the varimax solution; the quartimax solution shows a clear tendency to a general factor ($d_1 = 6.511$ clearly larger than d_2, d_3, d_4).

Table 7: Thirteen psychological tests (cf. Harman, 1976, p.172)

	before rotation			after rotation		
	Id.	loadings	c_i	Quartimax	Varimax	Chisquaremax
Loading matrix	1	0.607 -0.060 -0.443	0.568	0.205 -0.219 -0.691	0.166 -0.240 -0.695	0.233 -0.214 -0.684
	2	0.355 0.038 -0.266	0.198	0.157 -0.065 -0.411	0.136 -0.079 -0.416	0.173 -0.062 -0.405
	3	0.418 0.148 -0.429	0.381	0.190 0.041 -0.585	0.166 0.022 -0.594	0.214 0.045 -0.577
	4	0.478 0.083 -0.287	0.318	0.260 -0.080 -0.494	0.235 -0.100 -0.503	0.280 -0.077 -0.483
	5	0.729 0.257 0.244	0.657	0.767 -0.198 -0.171	0.751 -0.231 -0.199	0.774 -0.196 -0.142
	6	0.707 0.354 0.167	0.653	0.771 -0.085 -0.227	0.756 -0.120 -0.258	0.779 -0.083 -0.197
	7	0.721 0.367 0.257	0.721	0.827 -0.105 -0.157	0.815 -0.141 -0.190	0.833 -0.104 -0.125
	8	0.705 0.197 0.062	0.540	0.637 -0.187 -0.313	0.615 -0.218 -0.337	0.650 -0.185 -0.289
	9	0.698 0.409 0.252	0.718	0.832 -0.058 -0.151	0.822 -0.094 -0.185	0.837 -0.056 -0.118
	10	0.455 -0.482 0.399	0.599	0.236 -0.728 0.115	0.212 -0.734 0.120	0.232 -0.728 0.119
	11	0.537 -0.390 0.145	0.461	0.233 -0.621 -0.146	0.202 -0.632 -0.143	0.239 -0.620 -0.141
	12	0.487 -0.553 0.033	0.544	0.059 -0.704 -0.214	0.022 -0.709 -0.202	0.068 -0.702 -0.216
	13	0.674 -0.368 -0.135	0.608	0.220 -0.592 -0.457	0.177 -0.609 -0.454	0.238 -0.589 -0.452
d_1, d_2, d_3		4.620 1.392 0.954	6.965	3.299 1.917 1.749	3.101 2.031 1.833	3.404 1.905 1.657
Quartimax criterion	7.298			9.799 (29)	9.773	9.791
Varimax criterion	0.947			5.343	5.384 (16)	5.290
Chisquare criterion	1.296			2.240	2.226	2.242 (16)

Comments:

With $\gamma = 1$ monotonous convergence from below for $trace(\mathbf{A})$ and the criteria x_Q, x_V, x_C for all three procedures.

The chisquare solution is close to the quartimax solution, but also the varimax solution is not very different.

Table 8: Veränderungsskalen (Hechenbichler, 1999, p.177)

	before rotation						after rotation											
	Id	loadings				c_i	Quartimax				Varimax				Chisquaremax			
Loading matrix	1	-0.022	0.192	0.296	-0.210	0.169	0.107	-0.010	-0.161	0.363	0.103	-0.005	-0.162	0.364	0.098	0.000	-0.184	0.354
	2	0.342	-0.301	0.128	-0.350	0.347	0.570	-0.089	-0.085	-0.081	0.570	-0.089	-0.090	-0.076	0.565	-0.093	-0.110	-0.083
	3	0.522	-0.144	0.069	0.022	0.298	0.442	0.096	0.300	-0.059	0.446	0.105	0.291	-0.059	0.457	0.089	0.281	-0.046
	4	0.364	-0.384	-0.065	0.232	0.338	0.264	-0.096	0.399	-0.316	0.271	-0.091	0.398	-0.312	0.283	-0.109	0.400	-0.293
	5	-0.079	0.423	0.359	0.049	0.317	-0.130	0.066	0.023	0.543	-0.133	0.078	0.022	0.541	-0.130	0.080	0.003	0.542
	6	0.337	0.434	-0.290	0.077	0.392	-0.019	0.606	0.155	0.021	-0.021	0.610	0.140	0.007	-0.007	0.605	0.161	0.013
	7	0.471	-0.267	-0.139	0.033	0.313	0.384	0.097	0.263	-0.296	0.388	0.099	0.255	-0.296	0.398	0.084	0.259	-0.285
	8	0.350	-0.116	0.054	0.412	0.309	0.102	0.005	0.542	-0.067	0.110	0.018	0.540	-0.066	0.129	-0.003	0.539	-0.040
	9	0.377	0.391	-0.413	-0.141	0.485	0.103	0.682	-0.025	-0.093	0.098	0.680	-0.044	-0.107	0.106	0.679	-0.019	-0.111
	10	0.396	-0.344	0.019	-0.137	0.295	0.486	-0.059	0.110	-0.207	0.489	-0.057	0.105	-0.203	0.491	-0.068	0.095	-0.200
	11	-0.176	0.295	0.359	-0.024	0.247	-0.120	-0.061	-0.078	0.472	-0.123	-0.053	-0.075	0.473	-0.125	-0.047	-0.096	0.469
	12	0.332	-0.151	-0.033	0.359	0.263	0.104	0.024	0.478	-0.154	0.111	0.033	0.475	-0.154	0.128	0.015	0.479	-0.130
	13	-0.145	0.486	0.467	0.021	0.476	-0.156	0.026	-0.023	0.671	-0.160	0.039	-0.022	0.670	-0.158	0.044	-0.048	0.669
	14	0.651	-0.191	0.193	-0.249	0.560	0.727	0.084	0.156	0.019	0.728	0.093	0.144	0.021	0.734	0.078	0.119	0.024
	15	0.376	-0.064	-0.041	0.465	0.364	0.051	0.099	0.582	-0.113	0.059	0.111	0.578	-0.115	0.081	0.090	0.585	-0.086
	16	0.498	0.596	-0.199	-0.121	0.658	0.171	0.765	0.067	0.195	0.165	0.772	0.045	0.179	0.178	0.768	0.057	0.179
	17	0.612	-0.135	0.056	0.005	0.396	0.511	0.152	0.328	-0.066	0.515	0.162	0.317	-0.067	0.528	0.144	0.306	-0.054
	18	0.212	0.149	0.298	0.102	0.166	0.135	0.031	0.225	0.310	0.136	0.045	0.222	0.310	0.146	0.036	0.202	0.320
	19	0.136	-0.064	0.246	0.290	0.167	0.042	-0.146	0.356	0.132	0.047	-0.133	0.359	0.136	0.058	-0.146	0.345	0.153
	20	0.623	-0.001	-0.096	0.133	0.415	0.366	0.319	0.408	-0.111	0.370	0.330	0.395	-0.116	0.388	0.311	0.398	-0.098
	21	0.500	0.075	0.146	0.222	0.326	0.270	0.178	0.453	0.126	0.274	0.194	0.445	0.124	0.293	0.175	0.434	0.144
	22	0.412	0.540	-0.210	0.031	0.506	0.046	0.677	0.157	0.148	0.043	0.684	0.139	0.133	0.057	0.678	0.154	0.138
	23	0.226	0.574	-0.070	-0.096	0.394	-0.000	0.555	-0.025	0.293	-0.006	0.560	-0.040	0.281	0.002	0.562	-0.034	0.278
	24	0.416	0.270	0.335	0.098	0.368	0.257	0.190	0.319	0.406	0.258	0.209	0.310	0.402	0.273	0.196	0.287	0.416
	25	0.724	-0.189	0.217	-0.036	0.609	0.678	0.083	0.377	0.024	0.682	0.098	0.365	0.025	0.696	0.076	0.341	0.040
	26	-0.069	0.383	0.187	-0.045	0.189	-0.107	0.144	-0.075	0.388	-0.111	0.150	-0.077	0.384	-0.111	0.155	-0.087	0.380
	27	0.414	0.061	0.097	0.179	0.216	0.220	0.160	0.368	0.085	0.223	0.173	0.360	0.082	0.239	0.157	0.353	0.099
	28	0.284	0.352	0.017	0.072	0.210	0.057	0.360	0.177	0.214	0.056	0.370	0.167	0.206	0.067	0.363	0.167	0.213
	29	0.088	0.545	0.244	-0.052	0.367	-0.028	0.300	-0.007	0.525	-0.033	0.311	-0.014	0.518	-0.028	0.313	-0.029	0.517
	30	0.679	-0.278	0.059	-0.284	0.623	0.758	0.110	0.127	-0.140	0.760	0.115	0.113	-0.139	0.765	0.101	0.097	-0.136
	31	-0.092	0.233	0.140	0.082	0.089	-0.151	0.039	0.026	0.253	-0.152	0.045	0.027	0.251	-0.149	0.047	0.022	0.253
	32	0.282	0.139	-0.076	0.253	0.168	0.010	0.239	0.333	-0.000	0.013	0.248	0.326	-0.006	0.028	0.236	0.334	0.011
$d_1,\ldots,d_4$		5.069	3.264	1.468	1.238	11.040	3.311	2.808	2.510	2.411	3.350	2.893	2.423	2.373	3.481	2.821	2.384	2.353
Quartimax criterion	16.663						20.960 (74)				20.950				20.927			
Varimax criterion	6.958						12.859				12.909 (14)				12.867			
Chisquare criterion	1.857						2.725				2.725				2.731 (20)			

Comments:

With $\gamma = 1$ monotonous convergence from below for $trace(\mathbf{A})$ and the criteria x_Q, x_V, x_C for all three procedures.

The results for all three procedures are very similar.

Table 9: Eifersuchtsdaten (Hechenbichler, 1999, p.200)

	Id	initial loadings								c_i	after quartimax rotation							
Loading matrix	1	0.422	0.099	0.211	0.325	0.314	0.054	-0.163	-0.199	0.505	0.393	-0.151	0.141	0.084	0.113	0.211	0.493	0.017
	2	0.088	0.330	0.365	-0.042	0.183	-0.090	0.012	-0.239	0.350	0.438	0.161	-0.162	-0.166	-0.051	0.128	0.204	-0.133
	3	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	1.000	0.509	0.082	0.004	0.072	-0.148	0.090	0.057	-0.834
	4	-0.308	-0.193	-0.288	0.306	-0.009	0.148	0.001	0.090	0.339	-0.410	-0.371	-0.133	0.081	0.054	0.033	-0.040	0.056
	5	0.338	0.394	0.389	0.126	0.036	-0.182	0.116	-0.187	0.519	0.683	0.033	-0.038	-0.043	-0.125	0.145	0.108	-0.005
	6	0.217	0.227	0.264	-0.547	-0.133	0.037	-0.130	-0.141	0.523	0.257	0.666	0.051	0.021	0.004	0.034	-0.029	-0.090
	7	0.807	-0.459	0.051	-0.110	0.050	-0.002	-0.029	-0.015	0.880	0.156	0.102	0.880	0.139	-0.009	0.034	0.209	0.080
	8	0.164	0.065	0.147	0.107	-0.151	0.161	0.265	-0.174	0.213	0.194	-0.036	0.079	0.031	-0.112	0.369	-0.135	-0.008
	9	-0.327	-0.045	-0.228	-0.252	0.468	-0.197	0.311	-0.075	0.584	-0.241	-0.018	-0.092	-0.707	-0.001	-0.118	0.034	0.052
	10	0.195	0.207	0.154	-0.538	-0.028	0.262	-0.158	-0.011	0.489	0.121	0.618	0.070	0.032	0.264	0.089	-0.042	-0.087
	11	0.476	0.236	0.305	-0.140	0.021	-0.295	0.009	0.134	0.500	0.612	0.178	0.227	-0.011	0.010	-0.206	0.006	0.002
	12	0.135	-0.011	-0.186	0.393	0.094	-0.038	-0.152	0.006	0.241	0.020	-0.321	0.012	0.167	0.061	-0.019	0.237	0.222
	13	-0.323	-0.222	-0.113	-0.126	0.307	0.195	0.095	0.005	0.323	-0.407	-0.048	-0.027	-0.311	0.157	0.064	0.045	-0.163
	14	0.319	0.277	0.276	0.026	-0.265	0.131	0.254	0.176	0.438	0.474	0.031	0.107	0.154	0.086	0.183	-0.366	-0.045
	15	-0.192	-0.330	-0.117	0.251	0.277	0.124	0.023	0.188	0.350	-0.329	-0.410	0.080	-0.084	0.174	-0.039	0.101	-0.136
	16	0.210	0.432	0.147	-0.254	0.129	0.028	-0.000	0.253	0.398	0.407	0.253	-0.042	-0.097	0.365	-0.107	-0.112	0.004
	17	0.202	0.323	0.465	-0.113	0.238	-0.008	0.163	-0.190	0.494	0.536	0.183	-0.024	-0.258	0.024	0.217	0.124	-0.207
	18	0.072	0.295	0.333	0.138	0.112	-0.123	0.260	0.030	0.318	0.477	-0.139	-0.102	-0.182	-0.016	0.068	-0.070	-0.132
	19	0.067	-0.104	-0.017	0.191	-0.119	0.014	-0.293	-0.044	0.154	-0.057	-0.062	0.013	0.329	-0.051	-0.029	0.186	0.027
	20	0.324	0.340	0.349	0.065	-0.039	-0.054	-0.040	0.146	0.374	0.571	0.051	0.012	0.150	0.123	-0.027	-0.022	-0.078
	21	0.534	0.185	0.277	0.198	-0.312	-0.147	0.102	0.038	0.566	0.607	-0.047	0.233	0.296	-0.156	0.064	-0.131	0.086
	22	0.192	0.343	-0.061	-0.056	-0.287	0.058	0.056	-0.040	0.251	0.243	0.185	-0.083	0.157	0.008	0.131	-0.215	0.251
	23	0.417	0.361	0.209	0.107	0.225	0.103	0.047	-0.177	0.454	0.525	0.045	0.046	-0.057	0.169	0.291	0.224	0.086
	24	-0.158	-0.350	-0.171	0.535	0.016	0.004	0.110	0.048	0.477	-0.275	-0.610	0.049	0.085	-0.122	0.051	0.039	0.015
	25	0.376	0.308	0.330	0.062	0.006	-0.509	-0.048	0.120	0.625	0.697	0.012	0.058	0.026	-0.129	-0.334	0.080	0.026
	26	0.511	0.417	0.392	0.117	-0.186	0.013	-0.066	-0.081	0.648	0.705	0.146	0.044	0.300	0.008	0.189	0.039	0.023
	27	0.422	0.522	0.140	0.024	0.233	0.040	-0.004	0.086	0.535	0.599	0.081	-0.013	-0.051	0.357	0.071	0.103	0.153
	28	0.545	0.195	0.118	0.201	-0.229	0.303	0.008	0.136	0.552	0.405	-0.034	0.243	0.409	0.236	0.274	-0.129	0.117
	29	0.115	0.450	0.175	-0.004	0.251	0.141	0.145	0.162	0.376	0.408	0.002	-0.139	-0.182	0.375	0.108	-0.061	-0.037
	30	-0.153	-0.453	-0.122	-0.009	-0.272	-0.085	0.111	-0.324	0.442	-0.392	-0.013	0.142	0.028	-0.503	0.113	-0.039	0.007
	31	-0.213	-0.293	-0.146	0.065	0.074	0.024	0.258	0.127	0.246	-0.308	-0.269	0.096	-0.194	-0.023	-0.017	-0.163	-0.065
	32	0.230	0.119	0.166	0.073	-0.005	0.158	0.109	-0.331	0.247	0.239	0.072	0.053	0.007	-0.086	0.401	0.113	0.020
	33	0.325	0.331	0.290	0.128	0.102	0.155	0.280	0.163	0.456	0.538	-0.113	0.083	-0.061	0.258	0.221	-0.151	-0.068
	34	0.140	0.268	0.147	0.081	-0.052	0.286	0.137	-0.162	0.249	0.246	0.050	-0.078	0.049	0.090	0.408	-0.054	0.007
	35	0.277	0.208	0.177	0.143	0.066	0.259	0.136	-0.269	0.335	0.312	0.007	0.039	0.007	0.058	0.469	0.110	0.027
	36	0.677	-0.516	0.123	-0.060	0.018	0.007	0.198	0.066	0.787	0.133	-0.041	0.869	0.038	-0.061	0.072	0.019	-0.043
	37	0.176	0.177	0.019	-0.103	0.289	0.316	-0.197	0.240	0.353	0.091	0.113	0.047	0.031	0.559	0.017	0.122	-0.042
	38	0.437	-0.088	0.101	0.293	0.354	-0.022	-0.361	-0.066	0.555	0.231	-0.160	0.258	0.170	0.165	-0.018	0.594	0.029
	39	-0.143	-0.060	0.232	-0.139	0.045	0.086	-0.024	0.114	0.120	-0.033	0.087	-0.023	-0.035	0.079	-0.041	-0.051	-0.315
$d_1,\ldots,d_8$		4.388	3.442	3.043	1.853	1.491	1.101	1.000	0.950	17.267	6.609	2.041	1.966	1.459	1.453	1.362	1.241	1.137
Quartimax criterion	12.081										19.798 (82)							
Varimax criterion	6.193										12.076							
Chisquare criterion	2.377										4.003							

Table 9: Continued

	Id	after varimax rotation								after chisquare rotation							
Loading matrix	1	0.216	0.347	-0.067	0.179	0.046	0.151	0.527	-0.003	0.227	0.328	-0.081	0.126	0.125	0.067	0.548	0.055
	2	0.345	0.303	0.230	0.100	-0.125	-0.142	0.145	0.139	0.345	0.297	0.190	0.061	-0.153	-0.145	0.149	0.192
	3	0.427	0.267	0.136	-0.006	0.135	0.015	0.057	0.840	0.342	0.205	0.115	0.033	0.015	0.087	0.026	0.905
	4	-0.333	-0.112	-0.427	-0.079	0.009	-0.149	-0.010	-0.068	-0.389	-0.108	-0.378	-0.052	-0.138	0.039	-0.007	-0.099
	5	0.560	0.395	0.140	0.112	0.080	-0.016	0.104	0.011	0.586	0.377	0.057	0.051	-0.037	0.040	0.113	0.107
	6	0.105	0.088	0.691	0.056	0.067	0.063	-0.053	0.112	0.205	0.089	0.670	0.040	0.058	0.055	-0.075	0.103
	7	0.068	0.064	0.131	-0.036	0.129	0.876	0.253	-0.069	0.134	0.046	0.121	-0.040	0.864	0.151	0.268	-0.055
	8	0.059	0.407	-0.019	-0.057	0.127	0.083	-0.132	0.005	0.075	0.397	-0.041	-0.068	0.087	0.135	-0.128	0.042
	9	-0.038	-0.110	-0.054	0.015	-0.727	-0.073	-0.166	-0.078	-0.116	-0.060	-0.044	0.027	-0.056	-0.729	-0.129	-0.115
	10	-0.120	0.071	0.620	0.244	0.062	0.086	-0.044	0.115	-0.012	0.085	0.636	0.242	0.080	0.078	-0.053	0.059
	11	0.551	0.015	0.269	0.219	0.112	0.249	0.033	0.012	0.609	0.005	0.192	0.184	0.223	0.055	0.042	0.064
	12	0.019	-0.008	-0.293	0.046	0.109	0.004	0.304	-0.219	0.019	-0.014	-0.295	0.014	-0.011	0.124	0.317	-0.193
	13	-0.359	-0.057	-0.115	0.031	-0.388	-0.023	-0.051	0.151	-0.427	-0.032	-0.048	0.076	-0.004	-0.352	-0.034	0.082
	14	0.254	0.288	0.078	0.233	0.361	0.126	-0.285	0.056	0.318	0.277	0.029	0.225	0.114	0.332	-0.281	0.084
	15	-0.253	-0.134	-0.454	0.060	-0.167	0.074	0.099	0.125	-0.343	-0.130	-0.396	0.104	0.082	-0.140	0.119	0.085
	16	0.221	0.010	0.304	0.497	0.025	-0.008	-0.092	0.016	0.296	0.032	0.275	0.477	-0.031	-0.002	-0.073	-0.007
	17	0.373	0.416	0.256	0.205	-0.159	0.007	0.051	0.214	0.378	0.414	0.213	0.175	-0.004	-0.179	0.067	0.260
	18	0.424	0.257	-0.074	0.183	-0.047	-0.078	-0.093	0.128	0.394	0.252	-0.130	0.164	-0.088	-0.089	-0.076	0.184
	19	-0.058	-0.067	-0.056	-0.118	0.243	-0.006	0.265	-0.018	-0.048	-0.092	-0.044	-0.127	-0.013	0.256	0.245	-0.001
	20	0.405	0.147	0.134	0.293	0.268	0.031	0.055	0.097	0.454	0.130	0.080	0.262	0.006	0.230	0.057	0.142
	21	0.486	0.243	0.040	0.031	0.453	0.238	-0.023	-0.076	0.551	0.209	-0.044	-0.010	0.217	0.410	-0.028	0.014
	22	0.095	0.169	0.215	0.084	0.267	-0.077	-0.162	-0.239	0.192	0.176	0.175	0.045	-0.086	0.257	-0.166	-0.222
	23	0.266	0.456	0.132	0.307	0.019	0.072	0.228	-0.071	0.322	0.459	0.095	0.247	0.048	0.026	0.256	-0.027
	24	-0.133	-0.023	-0.641	-0.200	0.026	0.028	0.073	-0.036	-0.231	-0.039	-0.617	-0.177	0.038	0.043	0.082	-0.015
	25	0.741	-0.052	0.127	0.137	0.134	0.075	0.118	-0.018	0.766	-0.073	0.024	0.086	0.045	0.053	0.119	0.076
	26	0.440	0.384	0.252	0.193	0.429	0.059	0.138	0.002	0.533	0.355	0.182	0.133	0.030	0.403	0.132	0.080
	27	0.333	0.257	0.176	0.534	0.067	0.022	0.141	-0.130	0.418	0.273	0.129	0.472	-0.012	0.054	0.176	-0.109
	28	0.075	0.320	0.020	0.292	0.535	0.251	0.028	-0.093	0.181	0.305	-0.003	0.264	0.227	0.547	0.036	-0.073
	29	0.188	0.233	0.055	0.514	-0.055	-0.102	-0.061	0.049	0.217	0.255	0.035	0.491	-0.121	-0.067	-0.028	0.041
	30	-0.185	-0.002	-0.073	-0.615	-0.050	0.109	-0.096	-0.030	-0.233	-0.020	-0.061	-0.589	0.140	-0.034	-0.123	-0.008
	31	-0.177	-0.100	-0.325	-0.086	-0.195	0.093	-0.208	0.044	-0.251	-0.089	-0.297	-0.040	0.111	-0.188	-0.193	0.015
	32	0.069	0.463	0.108	-0.038	0.045	0.059	0.095	-0.016	0.097	0.453	0.091	-0.071	0.057	0.066	0.100	0.025
	33	0.286	0.375	-0.046	0.429	0.122	0.116	-0.112	0.078	0.316	0.378	-0.086	0.410	0.097	0.105	-0.080	0.102
	34	0.003	0.447	0.079	0.141	0.133	-0.065	-0.037	0.002	0.046	0.447	0.070	0.116	-0.070	0.152	-0.029	0.018
	35	0.059	0.542	0.053	0.120	0.069	0.052	0.110	-0.019	0.096	0.538	0.040	0.082	0.044	0.096	0.126	0.016
	36	0.087	0.102	-0.029	-0.064	0.083	0.868	0.042	0.042	0.108	0.083	-0.040	-0.043	0.866	0.093	0.062	0.058
	37	-0.157	0.001	0.127	0.522	0.025	0.069	0.173	0.068	-0.100	0.022	0.166	0.521	0.048	0.054	0.197	0.002
	38	0.135	0.069	-0.091	0.162	0.057	0.257	0.654	-0.012	0.144	0.048	-0.087	0.123	0.229	0.081	0.671	0.023
	39	-0.045	-0.057	0.069	0.060	-0.038	-0.019	-0.064	0.317	-0.077	-0.064	0.092	0.097	-0.013	-0.044	-0.073	0.291
$d_1,\ldots,d_8$		3.340	2.562	2.481	2.404	1.970	1.969	1.399	1.143	4.138	2.435	2.137	2.056	1.888	1.861	1.473	1.278
Quartimax criterion		19.194								19.093							
Varimax criterion		13.733 (90)								13.390							
Chisquare criterion		4.138								4.244 (49/ $\gamma = 0.5$)							

Comment: For $\gamma = 1$ the chisquare algorithm finally flutters between $(x_1, x_2) = (4.271..., 3.881...)$ and $(x_1, x_2) = (4.146..., 3.989...)$; for $\gamma = 0.5$ the algorithm converges monotonously within 49 iterations.

Appendix

A1 Maximum of the quartimax criterion

According to (3) the quartimax criterion is given by

$$x_Q = \sum_{ir} f_{ir}^2$$

where

$$f_{ir} = \lambda_{ir}^2 > 0 \quad \text{and} \quad \sum_r f_{ir} = c_i,\ c_i \text{ fixed}.$$

From Lemma 1 below we have

(7) $\quad \sum_r f_{ir}^2 \le c_i^2$ for $i = 1, \ldots, p$,

and so

$$x_Q = \sum_{i,r} f_{ir}^2 = \sum_i \left(\sum_r f_{ir}^2 \right) \le \sum_i c_i^2 .$$

The equality sign in (7) holds true if and only if just one of the values $f_{i1}, \ldots, f_{ik}$ is positive, and so the quartimax criterion x_Q attains the maximum value

$$\max x_Q = \sum_i c_i^2$$

if and only if in each row of the matrix $\mathbf{F} = (f_{ir})$ there is only one nonzero element. Note that these nonzero elements may all be in the same column.

Lemma 1:

Let $y_1, \ldots, y_k$ be real numbers with $y_r \ge 0$ for $r = 1, \ldots, k$ and $\sum y_r = c$, c constant. Then

$$\sum y_r^2 \le c^2 \quad \text{and} \quad \sum y_r^2 = c^2 \text{ if and only if just one of the values } y_r \text{ is positive}.$$

Proof:

We have

$$c^2 = \left(\sum_r y_r\right)^2 = \sum_r y_r^2 + \sum_{r \ne s} y_r y_s$$

and so

$$\sum_r y_r^2 = c^2 - \sum_{r \ne s} y_r y_s \le c^2 .$$

If more than one of the *n* values y_r were positive, then the sum $\sum_{r \ne s} y_r y_s$ were positive, but if only one of the *y*-values is positive this sum is zero.

A2 Maximum of the varimax criterion

According to (4) the varimax criterion is given by

(8) $$x_V = \sum_{r=1}^{k} \left(\sum_{i=1}^{p} f_{ir}^2 - d_r^2 / p \right) = \sum_{i,r} f_{ir}^2 - \frac{1}{p} \sum_{r=1}^{k} d_r^2 = x_Q - \frac{1}{p} \sum_{r=1}^{k} d_r^2$$

where

$$f_{ir} = \lambda_{ir}^2 > 0 \quad \text{and} \quad d_r = \sum_i f_{ir} .$$

Now x_Q becomes maximal if in every row of the loading matrix there is just one nonzero element, and from Lemma 2 below we see that the last sum in (8) becomes minimal for $d_1 = \ldots = d_k$. Note that $d_1 + \cdots + d_k = \sum f_{ir} = S$ (= sum of all communalities) is fixed. So the varimax criterion becomes maximal if in every row of the loading matrix there is just one nonzero element, and if all column sums $d_1, \ldots, d_k$ are equal. The reverse is not necessarily true. As the row sums (communalities) are fixed it may not be possible to arrange the loadings such that all column sums are equal.

Lemma 2:
Let $d_1, \ldots, d_k$ be real numbers with $d_r \geq 0$ for $r = 1, \ldots, k$ and $\sum d_r = S$, S constant. Then

$$\sum (d_r - \bar{d})^2 = \sum d_r^2 - k\bar{d}^2 = \sum d_r^2 - S^2 / k \geq 0,$$

and so $\sum d_r^2 \geq S^2 / k$; the equality sign holds true if and only if $d_1 = \cdots = d_k (= \bar{d})$.

A3 Maximum of the chisquare criterion

Let $f_{ir}, i = 1, \ldots, p, r = 1, \ldots, k, p \geq k$, be real numbers with $f_{ir} \geq 0$, and denote by $f_{i\bullet}$ the row sums and by $f_{\bullet r}$ the colum sums. We assume that all row and colum sums are positive. Then

$$x_C = \sum_{i,r} \frac{f_{ir}^2}{f_{i\bullet} f_{\bullet r}} \leq k$$

and

$x_C = k$ if and only if each row of $\mathbf{F} = (f_{ij})$ contains only one element different from zero .

Proof (see Cramér, 1945, p 282 and p 443)

1° We have $f_{ir} \leq f_{i\bullet}$ and so

(9) $$\frac{f_{ir}^2}{f_{i\bullet} f_{\bullet r}} \leq \frac{f_{ir}}{f_{\bullet r}} \quad \text{and} \quad x_C = \sum_{i,r} \frac{f_{ir}^2}{f_{i\bullet} f_{\bullet r}} \leq \sum_{i,r} \frac{f_{ir}}{f_{\bullet r}} = \sum_{r=1}^{k} \underbrace{\left(\sum_{i=1}^{p} \frac{f_{ir}}{f_{\bullet r}} \right)}_{=1} = k .$$

2° If $x_C = p$ then the equal sign must hold true in (9) and so

(10) $$\frac{f_{ir}^2}{f_{i\bullet} f_{\bullet r}} = \frac{f_{ir}}{f_{\bullet r}} \quad \text{for all } i, r ;$$

thus either $f_{ir} = 0$ or $f_{ir} = f_{i\bullet}$ which means that each row of $\mathbf{F} = (f_{ir})$ contains only one element different from zero.

3° If each row of $\mathbf{F} = (f_{ir})$ contains only one element different from zero then we have either $f_{ir} = 0$ or $f_{ir} = f_{i\bullet}$ for all i, r and so (10) holds true and thus

$$x_C = \sum_{i,r} \frac{f_{ir}^2}{f_{i\bullet} f_{\bullet r}} = \sum_{i,r} \frac{f_{ir}}{f_{\bullet r}} = k .$$

So our proof is complete. Note that we have tacitly assumed that all column sums $f_{\bullet r}$ are positive; if one column sum were zero the maximum would be reduced from k to $k-1$.

A4 Maple worksheets to compute the quartimax, varimax and chisquaremax solution

Note that in Table 4 to 9 the colums of the loading matrices are ordered such that $d_1 \geq d_2 \geq \cdots \geq d_k$. The loading matrices resulting from the following Maple worksheets are unordered.

a) Maple worksheet to compute the quartimax solution

```
> #
  # Factor Analysis
  # Quartimax as rotation criterion
  # by Leo Knüsel, University of Munich, October 2006
  #
> # Here: Simple example demonstrating the convergence problem
> restart;
> Digits := 15;
> with(LinearAlgebra):
> p := 5;
  k := 3;
  LAM0 := Matrix(p,k,[[0.5,0.5,0],[0.9,0,0.3],[0,1,0],[0,0,1],[1,0,0]]);
> printf("%6.3f\n",LAM0);
> LAM1 := Matrix(p,k):
  LAM2 := Matrix(p,k):
  M1 := Matrix(k,k,shape=identity):
  A1 := Matrix(k,k):
  B1 := Matrix(k,k):
  C1 := Matrix(p,k):
> # compute row sums (c1,...,cp) for LAM0
  c := Vector[row](1..p):
  for i from 1 to p do
    ci := 0;
    for r from 1 to k do
      ci := ci+LAM0[i,r]^2;
    end do;
    c[i] := ci;
    printf("%6.3f\n",c[i]);
  end do:
  printf("\n%6.3f\n",add(c[i],i=1..p));
> # compute column sums (d1,...,dp) for LAM0
  d := Vector[row](1..k):
  for r from 1 to k do
    dr := 0;
    for i from 1 to p do
      dr := dr+LAM0[i,r]^2;
    end do;
    d[r] := dr;
  end do:
  d;
  ss := add(d[r],r=1..k);
  printf("%6.3f\n",d);
> # Kaiser normalization?
  normalize := true;
  if normalize then
    for i from 1 to p do
      for r from 1 to k do
        LAM0[i,r] := LAM0[i,r]/sqrt(c[i]);
      end do;
    end do:
  end if;
> # compute quartimax criterion for LAM0
  x := 0:
  for r from 1 to k do
    for i from 1 to p do
      x := x + LAM0[i,r]^4;
    end do:
  end do:
  x;
> ###
  ### Iteration procedure to determine optimal rotation
  ###
  eps := 1e-9;
  gam := 1.0;
  itmax := 500;
  x1_old := 0:
  x2_old := x:
  LAM1 := LAM0.M1:
  for ii from 1 to itmax do
    # compute matrix C1 = (cij)
    for i from 1 to p do
      for r from 1 to k do
        C1[i,r]:= LAM1[i,r]^3;
      end do;
    end do;
    # determine singular value decomposition of B1
    B1 := Transpose(LAM0).C1;
    U, S, Vt := SingularValues(B1, output=['U', 'S', 'Vt']):
    DD := DiagonalMatrix(S[1..k]);
```

```
    A1 := Transpose(Vt).DD.Vt;
    M2 := U.Vt;
    LAM2 := LAM0.M2;
    x1 := Trace(A1);
    # compute x2 = quartimax criterion for LAM2
    x2 := 0;
    for r from 1 to k do
      for i from 1 to p do
        x2 := x2 + LAM2[i,r]^4;
      end do;
    end do:
    printf("%5d trace=%11.9f quartimax=%11.9f \n",ii,x1,x2);
    # break off iteration if desired relative accuracy is achieved
    if (abs(x1-x1_old)/x1 < eps and abs(x2-x2_old)/x2 < eps and abs(x1-x2)/x1 < eps) then
      break
    end if;
    x1_old := x1;
    x2_old := x2;
    M1 := M2;
    LAM1 := gam*LAM2 + (1-gam)*LAM1;
  end do:
> iter := ii;
  if(ii > itmax) then
    printf("\n!!!\n!!! No convergence with itmax = %d !!!\n!!!\n",itmax);
  else
    printf("\n!! Iteration procedure converged in %d iteration!!\n\n",iter);
  end if;
> M2;               # M2 = optimal rotation matrix
  M2.Transpose(M2); # Is M2 orthogonal?
  M2-M1;            # M2-M1 should be small!
> LAM2;             # optimal rotated loading matrix
  LAM2-LAM1;        # LAM2-LAM1 should be small!
> # compute column sums d1,...,dk for LAM2
  for r from 1 to k do
    dr := 0;
    for i from 1 to p do
      dr := dr+LAM2[i,r]^2;
    end do;
    d[r] := dr;
  end do:
  d;
> # compute varimax criterion for LAM2
  x := 0:
  for r from 1 to k do
    for i from 1 to p do
      x := x + LAM2[i,r]^4;
    end do;
    x := x - d[r]^2/p;
  end do:
  x;
> # restore original row sums c1,...,ck
  if normalize then
    for i from 1 to p do
      for r from 1 to k do
        LAM2[i,r] := LAM2[i,r]*sqrt(c[i]);
      end do;
    end do:
  end if;
> printf("%6.3f\n",LAM2);
> # compute column sums d1,...,dk for LAM2
  for r from 1 to k do
    dr := 0:
    for i from 1 to p do
      dr := dr+LAM2[i,r]^2;
    end do;
    d[r] := dr;
  end do:
  d;
  ss := add(d[r],r=1..k);
  printf("%6.3f\n",d);
> # compute chisquare criterion for LAM2
  x := 0:
  for i from 1 to p do
    for r from 1 to k do
      x := x + LAM2[i,r]^4/(c[i]*d[r]);
    end do;
  end do;
  x;
>
```

b) Maple worksheet to compute the varimax solution

```
> #
  # Factor Analysis
  # Varimax as rotation criterion
  # by Leo Knüsel, University of Munich, October 2006
  #
> # Here: Simple example demonstrating the convergence problem
> restart;
> Digits := 15;
> with(LinearAlgebra):
> p := 5;
  k := 3;
  LAM0 := Matrix(p,k,[[0.5,0.5,0],[0.9,0,0.3],[0,1,0],[0,0,1],[1,0,0]]);
> printf("%6.3f\n",LAM0);
> LAM1 := Matrix(p,k):
  LAM2 := Matrix(p,k):
  M1 := Matrix(k,k,shape=identity):
  A1 := Matrix(k,k):
  B1 := Matrix(k,k):
  C1 := Matrix(p,k):
> # compute row sums (c1,...,cp) for LAM0
  c := Vector[row](1..p):
  for i from 1 to p do
    ci := 0;
    for r from 1 to k do
      ci := ci+LAM0[i,r]^2;
    end do;
    c[i] := ci;
  end do:
  c;
  ss := add(c[i],i=1..p);
> # compute column sums (d1,...,dp) for LAM0
  d := Vector[row](1..k):
  for r from 1 to k do
    dr := 0;
    for i from 1 to p do
      dr := dr+LAM0[i,r]^2;
    end do;
    d[r] := dr;
  end do:
  d;
  ss := add(d[r],r=1..k);
  printf("%6.3f\n",d);
> # Kaiser normalization?
  normalize := true;
  if normalize then
    for i from 1 to p do
      for r from 1 to k do
        LAM0[i,r] := LAM0[i,r]/sqrt(c[i]);
      end do;
    end do:
  end if;
> # compute varimax criterion for LAM0
  # (d1,...,dp) for normalized LAM0
  for r from 1 to k do
    dr := 0;
    for i from 1 to p do
      dr := dr+LAM0[i,r]^2;
    end do;
    d[r] := dr;
  end do:
  # varimax criterion for normalized LAM0
  x := 0:
  for r from 1 to k do
    for i from 1 to p do
      x := x + LAM0[i,r]^4;
    end do;
    x := x - d[r]^2/p;
  end do:
  x;
> ###
  ### Iteration procedure to determine optimal rotation
  ###
  eps := 1e-9;
  gam := 1.0;
  itmax := 500;
  x1_old := 0:
  x2_old := x:
  LAM1 := LAM0.M1:
  for ii from 1 to itmax do
    # compute (d1,...,dk) for LAM1
    for r from 1 to k do
      dr := 0;
      for i from 1 to p do
        dr := dr+LAM1[i,r]^2;
      end do;
      d[r] := dr;
    end do:
    # compute matrix C1 = (cij)
    for i from 1 to p do
      for r from 1 to k do
        C1[i,r]:= LAM1[i,r]^3-d[r]*LAM1[i,r]/p;
```

```
      end do;
    end do;
    # determine singular value decomposition of B1
    B1 := Transpose(LAM0).C1;
    U, S, Vt := SingularValues(B1, output=['U', 'S', 'Vt']):
    DD := DiagonalMatrix(S[1..k]);
    A1 := Transpose(Vt).DD.Vt;
    M2 := U.Vt;
    LAM2 := LAM0.M2;
    x1 := Trace(A1);
    # compute column sums d1,...,dk for LAM2
    for r from 1 to k do
      dr := 0;
      for i from 1 to p do
        dr := dr+LAM2[i,r]^2;
      end do;
      d[r] := dr;
    end do:
    # compute x2 = varimax criterion for LAM2
    x2 := 0;
    for r from 1 to k do
      for i from 1 to p do
        x2 := x2 + LAM2[i,r]^4;
      end do;
      x2 := x2 - d[r]^2/p;
    end do:
    printf("%5d trace=%11.9f varimax=%11.9f \n",ii,x1,x2);
    # break off iteration if desired relative accuracy is achieved
    if (abs(x1-x1_old)/x1 < eps and abs(x2-x2_old)/x2 < eps and abs(x1-x2)/x1 < eps) then
      break
    end if;
    x1_old := x1;
    x2_old := x2;
    M1 := M2;
    LAM1 := gam*LAM2 + (1-gam)*LAM1;
  end do:
> iter := ii;
  if(ii > itmax) then
    printf("\n!!!\n!!! No convergence with itmax = %d !!!\n!!!\n",itmax);
  else
    printf("\n!! Iteration procedure converged in %d iteration!!\n\n",iter);
  end if;
> M2;               # M2 = optimal rotation matrix
  M2.Transpose(M2); # Is M2 orthogonal?
  M2-M1;            # M2-M1 should be small!
> LAM2;             # optimal rotated loading matrix
  LAM2-LAM1;        # LAM2-LAM1 should be small!
> # compute quartimax criterion for LAM2
  x := 0:
  for r from 1 to k do
    for i from 1 to p do
      x := x + LAM2[i,r]^4;
    end do;
  end do:
  x;
> # restore original row sums c1,...,ck
  if normalize then
    for i from 1 to p do
      for r from 1 to k do
        LAM2[i,r] := LAM2[i,r]*sqrt(c[i]);
      end do;
    end do:
  end if;
> printf("%6.3f\n",LAM2);
> # compute column sums d1,...,dk for LAM2
  for r from 1 to k do
    dr := 0;
    for i from 1 to p do
      dr := dr+LAM2[i,r]^2;
    end do;
    d[r] := dr;
  end do:
  d;
  ss := add(d[r],r=1..k);
  printf("%6.3f\n",d);
> # compute chisquare criterion for LAM2
  x := 0:
  for i from 1 to p do
    for r from 1 to k do
      x := x + LAM2[i,r]^4/(c[i]*d[r]);
    end do;
  end do;
  x;
>
```

c) Maple worksheet to compute the chisquare solution

```
> #
  # Factor Analysis
  # Chisquare as rotation criterion
  # by Leo Knüsel, University of Munich, October 2006
  #
> # Here: Simple example demonstrating the convergence problem
> restart;
> Digits := 15;
> with(LinearAlgebra):
> p := 5;
  k := 3;
  LAM0 := Matrix(p,k,[[0.5,0.5,0],[0.9,0,0.3],[0,1,0],[0,0,1],[1,0,0]]);
> printf("%6.3f\n",LAM0);
> LAM1 := Matrix(p,k):
  LAM2 := Matrix(p,k):
  M1 := Matrix(k,k,shape=identity):
  A1 := Matrix(k,k):
  B1 := Matrix(k,k):
  C1 := Matrix(p,k):
> # compute row sums (c1,...,cp) for LAM0
  c := Vector[row](1..p):
  for i from 1 to p do
    ci := 0;
    for r from 1 to k do
      ci := ci+LAM0[i,r]^2;
    end do;
    c[i] := ci;
  end do:
  c;
  ss := add(c[i],i=1..p);
> # compute column sums (d1,...,dp) for LAM0
  d := Vector[row](1..k):
  for r from 1 to k do
    dr := 0;
    for i from 1 to p do
      dr := dr+LAM0[i,r]^2;
    end do;
    d[r] := dr;
  end do:
  d;
  ss := add(d[r],r=1..k);
  printf ("%6.3f\n",d);
> # compute chisquare criterion for LAM0
  x := 0:
  for i from 1 to p do
    for r from 1 to k do
      x := x + LAM0[i,r]^4/(c[i]*d[r]);
    end do;
  end do;
  x;
> ###
  ### Iteration procedure to determine optimal rotation
  ###
  eps := 1e-9;
  gam := 1.0;
  itmax := 500;
  x1_old := 0:
  x2_old := x:
  LAM1 := LAM0.M1:
  for ii from 1 to itmax do
    # compute matrix C1 = (cij)
    for i from 1 to p do
      for r from 1 to k do
        dr := 0;
        er := 0;
        for j from 1 to p do
          dr := dr + LAM1[j,r]^2;
          er := er + LAM1[j,r]^4/c[j];
        end do;
        C1[i,r]:= LAM1[i,r]^3/(c[i]*dr)-LAM1[i,r]*er/(2*dr^2);
      end do;
    end do;
    # determine singular value decomposition of B1
    B1 := Transpose(LAM0).C1;
    U, S, Vt := SingularValues(B1, output=['U', 'S', 'Vt']):
    DD := DiagonalMatrix(S[1..k]);
    A1 := Transpose(Vt).DD.Vt;
    M2 := U.Vt;
    LAM2 := LAM0.M2;
    x1 := 2*Trace(A1);
    # compute (d1,...,dk) for LAM2
    for r from 1 to k do
      dr := 0;
      for i from 1 to p do
        dr := dr+LAM2[i,r]^2;
      end do;
      d[r] := dr;
    end do:
    # compute x2 = chisquare criterion for LAM2
    x2 := 0;
    for i from 1 to p do
```

```
      for r from 1 to k do
        x2 := x2 + LAM2[i,r]^4/(c[i]*d[r]);
      end do;
    end do;
    printf("%5d 2*trace=%11.9f chi2max=%11.9f \n",ii,x1,x2);
    # break off iteration if desired relative accuracy is achieved
    if (abs(x1-x1_old)/x1 < eps and abs(x2-x2_old)/x2 < eps and abs(x1-x2)/x1 < eps) then
      break
    end if;
    x1_old := x1;
    x2_old := x2;
    M1 := M2;
    LAM1 := gam*LAM2 + (1-gam)*LAM1;
  end do:
> iter := ii;
  if(ii > itmax) then
    printf("\n!!!\n!!! No convergence with itmax = %d !!!\n!!!\n",itmax);
  else
    printf("\n!! Iteration procedure converged in %d iteration!!\n\n",iter);
  end if;
> M2;               # M2 = optimal rotation matrix
  M2.Transpose(M2); # Is M2 orthogonal?
  M2-M1;            # M2-M1 should be small!
> LAM2;                # optimal rotated loading matrix
  LAM2-LAM1;           # LAM2-LAM1 should be small!
  printf("%6.3f\n",LAM2);
  printf("\n%6.3f\n",d); # column sums d1,...,dk for LAM2
> #
  # compute of quartimax and varimax criterion for LAM2 (in normalized form)
  #
> # normalize LAM2 (Kaiser normalization)
  for i from 1 to p do
    for r from 1 to k do
      LAM2[i,r] := LAM2[i,r]/sqrt(c[i]);
    end do;
  end do:
> # compute column sums d1,...,dk for LAM2
  for r from 1 to k do
    dr := 0;
    for i from 1 to p do
      dr := dr+LAM2[i,r]^2;
    end do;
    d[r] := dr;
  end do:
  d;
> # compute quartimax criterion for LAM2
  x := 0:
  for r from 1 to k do
    for i from 1 to p do
      x := x + LAM2[i,r]^4;
    end do;
  end do:
  x;
> # compute varimax criterion for LAM2
  x := 0:
  for r from 1 to k do
    for i from 1 to p do
      x := x + LAM2[i,r]^4;
    end do;
    x := x - d[r]^2/p;
  end do:
  x;
>
```

A5 Programs in R to compute the quartimax, varimax and chisquaremax solution

Note that in Table 4 to 9 the colums of the loading matrices are ordered such that $d_1 \geq d_2 \geq \cdots \geq d_k$. The loading matrices resulting from the following R programs are unordered.

a) Original R function to compute the varimax solution

Help on varimax:

```
varimax(x, normalize=TRUE, eps=1e-5)
  x = (p x k)  = loading matrix
  normalize: If Kaiser-normalization is required the rows of x are rescaled to unit length before
    rotation and scaled back afterwards;
  eps = tolerance for stopping: the relative change in the sum of singular values;
value: A list with components
  loadings: The rotated loadings matrix, x*rotmax;
  rotmax: The rotation matrix
```

Definition of varimax function:

```
function(x, normalize=TRUE, eps=1e-5)
{
  nc <– ncol(x)
  if (nc < 2)
    return x;
  if(normalize) {
    sc <– sqrt(drop(apply(x, 1, function(x) sum(x^2))))
    x <– x/sc
  }
  p <– nrow(x)
  TT <– diag(nc)
  d <– 0
  for (i in 1:1000) {
    z <– x %*% TT
    B <– t(x) %*% (z^3 – z %*% diag(drop(rep(1,p) %*% z^2))/p)
    sB <– La-svd(B)
    TT <– sB$u %*% sB$vt
    dpast <– d
    d <– sum(sB$d)
    if (d < dpast*(1+eps))
      break
  }
  z <– x %*% TT
  if (normalize)
    z <– z * sc
  dimnames(z) <– dimnames(x)
  class(z) <– “loadings”
  list(loadings = z, rotmat = TT)
}
```

Note that here the condition `(d < dpast*(1+eps))` for breaking off the iteration assumes that the trace *d* is monotonously increasing. This is not necessarily true (see section 6).

b) R function to compute the quartimax solution

```
# Definition of quartimax function
quartimax <-
function (x, normalize=TRUE, eps=1e-09, gamma=1)
{
  nc <- ncol(x)
  if (nc < 2)
    return(x)
  if (normalize) {
    sc <- sqrt(drop(apply(x, 1, function(x) sum(x^2))))
    x <- x/sc
  }
  p <- nrow(x)
  TT <- diag(nc)
  d <- 0
  q <- sum(x^4)
  z <- x
  for (i in 1:1000) {
    B <- t(x) %*% z^3
    sB <- La.svd(B)
    TT <- sB$u %*% sB$vt
    dpast <- d
    d <- sum(sB$d)
    zpast <- z
    z <- x %*% TT
    qpast <- q
    q <- sum(z^4)
    cat("i=",i," trace=",d," quartimax=",q,"\n")
    if (abs(d-dpast)/d < eps && abs(q-qpast)/q < eps  && abs(d-q)/d < eps)
```

```
      break
    z <- gamma*z + (1-gamma)*zpast  # can improve convergence
  }
  if (normalize)
    z <- z * sc
  dimnames(z) <- dimnames(x)
  cat("\n")
  list(iterations = i, rotmat = TT, loadings = z, SS_columns = rep(1,p)%*%z^2)
}

# Data matrix
x <- matrix(c(
 0.5,0.5,0,
 0.9,0,0.3,
 0,1,0,
 0,0,1,
 1,0,0
), nrow = 5, ncol=3, byrow=TRUE)
quartimax(x)
```

Note that here the condition

```
   (abs(d-dpast)/d < eps && abs(q-qpast)/q < eps  && abs(d-q)/d < eps)
```

for breaking off the iteration not only requires the convergence of d $(= trace(A))$ and q $(=$ quartimax criterion), but additionally that d converges to q. This is sensible in view of the findings in section 6.

c) R function to compute the varimax solution

```
# Definition of varimax-function
varimax <-
function (x, normalize=TRUE, eps=1e-09, gamma=1)
{
    nc <- ncol(x)
    if (nc < 2)
        return(x)
    if (normalize) {
        sc <- sqrt(drop(apply(x, 1, function(x) sum(x^2))))
        x <- x/sc
    }
    p <- nrow(x)
    TT <- diag(nc)
    d <- 0
    v <- sum(x^4) - sum((rep(1, p) %*% x^2)^2)/p
    z <- x
    for (i in 1:1000) {
        B <- t(x) %*% (z^3 - z %*% diag(drop(rep(1, p) %*% z^2))/p)
        sB <- La.svd(B)
        TT <- sB$u %*% sB$vt
        dpast <- d
        d <- sum(sB$d)
        zpast <- z
        z <- x %*% TT
        vpast <- v
        v <- sum(z^4) - sum((rep(1, p) %*% z^2)^2)/p
        cat("i=",i," trace=",d," varimax=",v,"\n")
        if (abs(d-dpast)/d < eps && abs(v-vpast)/v < eps  && abs(d-v)/d < eps)
            break
        z <- gamma*z + (1-gamma)*zpast  # can improve convergence
    }
    z <- x %*% TT
    if (normalize)
        z <- z * sc
    dimnames(z) <- dimnames(x)
    #class(z) <- "loadings"
    cat("\n")
    list(iterations = i, rotmat = TT, loadings = z, SS_columns = rep(1,p)%*%z^2)
}
# Data matrix
 x <- matrix(c(
 0.5,0.5,0,
 0.9,0,0.3,
 0,1,0,
 0,0,1,
 1,0,0
), nrow = 5, ncol=3, byrow=TRUE)
varimax(x)
```

d) R function to compute the chisquaremax solution

```
# Definition of chi2max-function
chi2max <-
function(x,eps=1e-9,gamma=1)
{
    nc <- ncol(x)
    if (nc < 2)
        return(x)
    p <- nrow(x)
    k <- ncol(x)
    TT <- diag(nc)
    c <- 0
    d <- 0
    z <- x
    H <- drop(z^2 %*% rep(1, k))    # SS rows (communalities)
    for (i in 1:1000) {
        D <- drop(rep(1, p) %*% z^2) # SS columns
        E <- drop((1/H) %*% z^4)
        C <- diag(1/H) %*% z^3 %*% diag(1/D) - (1/2) * z %*% diag(E) %*% diag(1/D^2)
        B <- t(x) %*% C
        sB <- La.svd(B)
        TT <- sB$u %*% sB$vt
        dpast <- d
        d <- sum(sB$d)
        zpast <- z
        z <- x %*% TT
        D <- drop(rep(1, p) %*% z^2) # SS columns
        cpast <- c
        c <- sum(diag(1/H) %*% z^4 %*% diag(1/D))
        cat("i=",i," 2*d=",2*d," c=",c,"\n")
        if (abs(d-dpast)/d < eps && abs(c-cpast)/c < eps  && abs(d-c/2)/d < eps)
            break
        z <- gamma*z + (1-gamma)*zpast  # can improve convergence
    }
    dimnames(z) <- dimnames(x)
    #class(z) <- "loadings"
    cat("\n")
    list(iterations = i, rotmat = TT, loadings = z, SS_columns = rep(1,p)%*%z^2)
}

# Data matrix
x <- matrix(c(
 0.5,0.5,0,
 0.9,0,0.3,
 0,1,0,
 0,0,1,
 1,0,0
 ), nrow = 5, ncol=3, byrow=TRUE)
chi2max(x,gamma=1.0)
```

References

Cramér, Harald (1945). Mathematical Methods of Statistics. 575 pp. Princeton University Press.

Cureton, E.E. and Mulaik, S.A. (1975). The Weighted Varimax Rotation and the Promax Rotation. Psychometrika, 40, 183-195.

Harman, Harry H. (1976). Modern Factor Analysis. Third edition, revised. 487 pp. University of Chicago Press, Chicago.

Hechenbichler, Klaus (1999). Zu den Rotationsverfahren der Faktorenanalyse. 216 pp. Diploma thesis, University of Munich. A copy of the paper can be found at URL *www.stat.uni-muenchen.de/~knuesel*

Lawley, D.N. and Maxwell, A.E. (1971). Factor Analysis as a Statistical Method. Second edition. 153 pp. Butterworths London.

Maple (2006). Waterloo Maple Inc. URL: *www.maplesoft.com*

R Development Core Team (2006). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL: *www.R-project.org*.

Thurstone, L.L. (1947). Multiple-Factor Analysis. 535 pp. University of Chicago Press, Chicago.