PSYCHOMETRIKA—VOL. 61, NO. 2, 375–389
JUNE 1996

A MONOTONICALLY CONVERGENT ALGORITHM FOR ORTHOGONAL CONGRUENCE ROTATION

HENK A. L. KIERS

UNIVERSITY OF GRONINGEN

PATRICK GROENEN

UNIVERSITY OF LEIDEN

Brokken has proposed a method for orthogonal rotation of one matrix such that its columns have a maximal sum of congruences with the columns of a target matrix. This method employs an algorithm for which convergence from every starting point is not guaranteed. In the present paper, an iterative majorization algorithm is proposed which is guaranteed to converge from every starting point. Specifically, it is proven that the function value converges monotonically, and that the difference between subsequent iterates converges to zero. In addition to the better convergence properties, another advantage of the present algorithm over Brokken's one is that it is easier to program. The algorithms are compared on 80 simulated data sets, and it turned out that the new algorithm performed well in all cases, whereas Brokken's algorithm failed in almost half the cases. The derivation of the algorithm is given in full detail because it involves a series of inequalities that can be of use to derive similar algorithms in different contexts.

Key words: optimization, majorization, matching, Procrustes rotation.

In several situations researchers want to compare factors from one study to those obtained in a different study. Because factors and their loadings are determined up to a rotation only, it is often advocated to use this rotational freedom by rotating the one solution such that it maximally resembles the other (target) solution. One procedure for such a matching rotation was proposed by Green (1952), also see Cliff (1966). In this method (called "Procrustes rotation" henceforth) a loading matrix $A(n \times r)$ is rotated by an $(r \times r)$ orthonormal matrix T such that it optimally resembles a target loading matrix $B(n \times r)$ in the least squares sense. Procrustes methods involving translations and isotropic scalings in addition to rotations (e.g., see Goodall, 1991) are ignored in the present paper, because such transformations are not allowed in the comparison of factor loading matrices.

Brokken (1983) explains that the criterion used in Procrustes rotation is too restrictive for comparing factor loadings: Whereas proportionality of corresponding columns in AT and B is sufficient for inferring invariance of factors, the criterion used in Procrustes rotation is only 0 (implying a perfect match) when corresponding columns of AT and B are equal. For this reason, Brokken proposed an alternative procedure for orthogonal rotation of a loading matrix so as to optimally match a target loading matrix. His procedure aims at maximizing the average congruence between the columns of the rotated factor loading matrix and the target loading matrix. The congruence (Tucker, 1951) between two columns measures the degree of proportionality of two columns, and it has been advocated as a primary tool for comparison of factor structures (e.g., ten

This research has been made possible by a fellowship from the Royal Netherlands Academy of Arts and Sciences to the first author. The authors are obliged to Willem J. Heiser and Jos M. F. ten Berge for useful comments on an earlier version of this paper.

Requests for reprints should be sent to Henk A. L. Kiers, Department of Psychology (SPA), Grute Kruisstraat 2/1, 9712TS Groningen, THE NETHERLANDS.

0033-3123/96/0600-93154$00.75/0

Berge, 1986). Hence maximizing a sum of congruences seems preferable in cases where factor loading matrices are to be rotated towards each other.

Recently, Koschat and Swayne (1991) proposed an algorithm for weighted Procrustes analysis, and showed that their procedure could be employed in a procedure for maximizing the sum of *squared* congruences between corresponding columns of AT and B. They argue that "a change from $\phi = 0$ to $\phi = .3$ [ϕ is the congruence] means that there is no change in the assessment of 'no association', while a change from $\phi = .7$ to $\phi = 1$ means a change from 'mild association' to 'strongest possible association'. This suggests that instead of considering the sum of the ϕ_i, one ought to consider the self weighted sum of the ϕ_i" (p. 238). However, their argument seems somewhat subjective: One could just as well claim that a change from $\phi = 0$ to $\phi = .44$ is more substantial than a change from $\phi = .9$ to $\phi = 1$, whereas in terms of ϕ^2 the change is .19 in both cases. It seems that the choice for ϕ or ϕ^2 cannot be settled by arguments only. An empirical comparison of both methods, however, has so far been hampered by the complications of Brokken's algorithm. The present paper proposes an alternative algorithm for Brokken's method, and thus facilitates such a comparison.

As mentioned above, Brokken (1983) proposed to maximize the sum of congruences between corresponding columns of AT and B. Specifically, he proposed to maximize

$$\mathrm{f}(T) = \sum_{l=1}^{r} \phi(A\mathbf{t}_l, \mathbf{b}_l), \tag{1}$$

where ϕ denotes Tucker's (1951) coefficient of congruence, $\mathbf{t}_l$ denotes the l-th column of T, and $\mathbf{b}_l$ denotes the l-th column of B, $l = 1, \ldots, r$. The coefficient ϕ is defined as the normalized inner product between two columns; for cases where one of the vectors is zero, ϕ is defined here to be zero. We can rewrite (1) as

$$\mathrm{f}(T) = \sum_{l=1}^{r} \frac{\mathbf{t}_l' A' \mathbf{b}_l}{(\mathbf{t}_l' A' A \mathbf{t}_l)^{1/2} (\mathbf{b}_l' \mathbf{b}_l)^{1/2}}, \tag{2}$$

where it is to be kept in mind that a term is defined to be zero if the denominator is zero. It should be noted that the function (2) is insensitive to columnwise rescalings of AT and/or B.

Brokken (1983) proposed to solve the problem of maximizing $\mathrm{f}(T)$ subject to $T'T = I$ by solving the normal equation of the Lagrangian function $\mathrm{G}(T, \Theta) = \mathrm{f}(T) + \mathrm{tr}\,[\Theta(TT' - I)]$, and used a Newton-iteration maximization procedure for doing so (p. 344). A potential problem of Brokken's application of the Newton method (henceforth called "Brokken's algorithm") is that it does not always converge. That is, convergence of the Newton method is only guaranteed if the algorithm is started in the immediate vicinity of a (local) optimum or saddle-point, or when it lands in such a vicinity by coincidence. Hence, in order to use Brokken's algorithm, we need a good start for the rotation matrix. In cases where the columns of AT and B have nearly equal sums of squares, the Procrustes solution can be expected to furnish such a good start for T. This is because the Procrustes rotation maximizes the weighted sum of congruence coefficients (see Korth & Tucker, 1976, p. 533), and in case AT and B have the same column sums of squares the weights are all identical. In other cases, however, no such good start is available, and, as will be demonstrated below, Brokken's algorithm will fail to find the global maximum.

The purpose of the present paper is to develop an alternative algorithm for max-

imizing f(T) subject to $T'T = I$, that increases f monotonically, converges from every starting point, and is relatively easy to program. The monotonically convergent algorithm for maximizing f will be based on "majorization" (see, for instance, de Leeuw & Heiser, 1980; Meulman, 1986; Kiers, 1990; Groenen, 1993). Because majorization is used for minimization problems, we consider the problem of minimizing $h(T) \equiv -f(T)$, which is, obviously, equivalent to maximizing f(T). The majorization approach can be used for minimizing h(T), by iteratively decreasing h(T) as follows. Let the current T be denoted by T_c. Suppose $g(T, T_c)$ is a function of T and T_c such that $h(T) \leq g(T, T_c)$ for all T (i.e., $g(T, T_c)$ majorizes h(T)), and $h(T_c) = g(T_c, T_c)$. Also, suppose that we have a procedure for updating T_c by T_m such that $g(T_m, T_c) \leq g(T_c, T_c)$. Then this procedure decreases h, because $h(T_m) \leq g(T_m, T_c) \leq g(T_c, T_c) = h(T_c)$. Note that "decrease" is used for "not increase", for reasons of readability. Hence, if, for a given T_c, we can find a function g such that $h(T) \leq g(T, T_c)$ for all T (possibly subject to a constraint on T), and $h(T_c) = g(T_c, T_c)$, and if we know how to minimize g, we can construct an algorithm that monotonically decreases h(T). Therefore, the derivation of our algorithm will mainly consist of the derivation of a sequence of inequalities to establish a majorizing function $g(T, T_c)$.

In the first section, the inequalities that are to be used for establishing the majorizing function will be derived. In the second section, the majorizing function will be defined, and it will be shown how this can be minimized. In a third section, the resulting algorithm for maximizing f(T) is summarized schematically, especially for those who want to program the algorithm without studying the derivation. Finally, in the fourth section, some results on the performance of the algorithms for maximizing f are given.

Some Inequalities for Establishing the Majorizing Function

In the present section, the inequalities are derived that are needed to establish a function that majorizes $h(T) = -f(T)$. Some of these inequalities can be found in the literature, but all proofs will be given here for the sake of completeness. The inequalities are denoted as Lemmas 1 through 6.

Lemma 1. Let x and y be real positive numbers. Then

$$-y^{-1/2} \leq x^{-1}y^{1/2} - 2x^{-1/2}, \tag{3}$$

with equality iff $x = y$.

Proof. From $(x^{1/2} - y^{1/2})^2 \geq 0$ it follows that $2x^{1/2}y^{1/2} \leq x + y$. After division by $xy^{1/2}$, we obtain $2x^{-1/2} \leq y^{-1/2} + x^{-1}y^{1/2}$ from which (3) follows at once. We have equality iff $(x^{1/2} - y^{1/2})^2 = 0$, hence iff $x = y$. □

Lemma 2. Let x and y be real numbers such that $x > 0$ and $y \geq 0$. Then

$$y^{1/2} \leq \frac{1}{2}x^{1/2} + \frac{1}{2}x^{-1/2}y, \tag{4}$$

with equality iff $x = y$. (See, Groenen & Heiser, 1991, p. 14).

Proof. From $(x^{1/2} - y^{1/2})^2 x^{-1/2} \geq 0$ it follows at once that $y^{1/2} \leq \frac{1}{2}x^{1/2} + \frac{1}{2}x^{-1/2}y$, with equality iff $x^{1/2} = y^{1/2}$, hence $x = y$. □

Lemma 3. Let x and y be real positive numbers. Then

$$-y^{-1/2} \leq -\frac{3}{2}x^{-1/2} + \frac{1}{2}x^{-3/2}y, \tag{5}$$

with equality iff $x = y$.

Proof. Combining Lemma 1 and Lemma 2 by applying Lemma 2 to $y^{1/2}$ in the right hand side of (3), we have $-y^{-1/2} \leq x^{-1}y^{1/2} - 2x^{-1/2} \leq x^{-1}(\frac{1}{2}x^{1/2} + \frac{1}{2}x^{-1/2}y) - 2x^{-1/2} = -\frac{3}{2}x^{-1/2} + \frac{1}{2}x^{-3/2}y$, with equality iff $x = y$. □

Lemma 4. Let x_1, x_2, y_1 and y_2 be real numbers such that $x_1 > 0$, $x_2 > 0$, $y_1 \geq 0$ and $y_2 > 0$. Then

$$-y_1y_2^{-1} \leq x_1x_2^{-2}y_2 + x_1^{-1}x_2^{-1}y_1^2 - 3x_2^{-1}y_1, \tag{6}$$

with equality iff $x_1 = y_1$ and $x_2 = y_2$.

Proof. If $y_1 = 0$, (6) follows at once as a strict inequality. It remains to prove (6) for $y_1 > 0$. From $(y_1^{1/2}y_2^{-1/2} - x_1^{1/2}x_2^{-1}y_2^{1/2})^2 \geq 0$ it follows that $-y_1y_2^{-1} \leq x_1x_2^{-2}y_2 - 2x_1^{1/2}x_2^{-1}y_1^{1/2}$. From Lemma 3, we have $-y_1^{-1/2} \leq -\frac{3}{2}x_1^{-1/2} + \frac{1}{2}x_1^{-3/2}y_1$, hence $-2x_1^{1/2}x_2^{-1}y_1y_1^{-1/2} \leq (2x_1^{1/2}x_2^{-1}y_1)(-\frac{3}{2}x_1^{-1/2} + \frac{1}{2}x_1^{-3/2}y_1) = -3x_2^{-1}y_1 + x_1^{-1}x_2^{-1}y_1^2$, with equality iff $x_1 = y_1$. Combining these inequalities, we have (6) at once. We have equality in (6) iff $x_1 = y_1$ and $(y_1^{1/2}y_2^{-1/2} - x_1^{1/2}x_2^{-1}y_2^{1/2}) = 0$, hence iff $x_1 = y_1$ and $x_2 = y_2$. □

Lemma 5. Let x and y be real numbers, and define sgn (x) to be -1, 0, or 1, if x is negative, zero, or positive, respectively. Then

$$-|y| \leq -\text{sgn}\ (x) \cdot y, \tag{7}$$

with equality iff x and y have the same sign or $y = 0$.

Proof. If $y = 0$, the inequality follows trivially as an equality. If $y \neq 0$ and $x = 0$, then (7) follows trivially as a strict inequality. If $y \neq 0$, and $x \neq 0$, the inequality $(\text{sgn}\ (x) - \text{sgn}\ (y))^2 \geq 0$ yields sgn (x)sgn $(y) \leq 1$. Substituting sgn $(y) = |y|^{-1}y$, we obtain sgn $(x)y \leq |y|$, from which (7) follows at once, with equality iff sgn $(x) =$ sgn (y). □

Lemma 6. Let $\mathbf{x}$ and $\mathbf{y}$ be vectors of order r such that $\mathbf{x}'\mathbf{x} = \mathbf{y}'\mathbf{y} = 1$, let Z be a square matrix of order $r \times r$, and let λ be a positive value larger than or equal to the largest eigenvalue of $S \equiv \frac{1}{2}Z + \frac{1}{2}Z'$. Then

$$\mathbf{y}'Z\mathbf{y} \leq -\mathbf{x}'Z\mathbf{x} + 2\mathbf{x}'(S - \lambda I)\mathbf{y} + 2\lambda, \tag{8}$$

with equality for every choice of λ iff $\mathbf{x} = \mathbf{y}$. (See, Heiser, 1987, p. 345; also, see Kiers, 1990).

Proof. Define $\mathbf{e} \equiv \mathbf{y} - \mathbf{x}$, hence $\mathbf{y} = \mathbf{x} + \mathbf{e}$. Then $\mathbf{y}'Z\mathbf{y} = \mathbf{x}'Z\mathbf{x} + \mathbf{x}'(Z + Z')\mathbf{e} + \mathbf{e}'Z\mathbf{e} = \mathbf{x}'Z\mathbf{x} + 2\mathbf{x}'S\mathbf{e} + \mathbf{e}'Z\mathbf{e}$. Because $\mathbf{e}'Z\mathbf{e} = \mathbf{e}'S\mathbf{e}$ and $\mathbf{e}'S\mathbf{e} \leq \lambda\mathbf{e}'\mathbf{e}$, we have $\mathbf{y}'Z\mathbf{y} \leq \mathbf{x}'Z\mathbf{x} + 2\mathbf{x}'S\mathbf{e} + \lambda\mathbf{e}'\mathbf{e} = \mathbf{x}'Z\mathbf{x} + 2\mathbf{x}'S(\mathbf{y} - \mathbf{x}) + \lambda(\mathbf{y} - \mathbf{x})'(\mathbf{y} - \mathbf{x}) = -\mathbf{x}'Z\mathbf{x} + 2\mathbf{x}'S\mathbf{y} - 2\lambda\mathbf{x}'\mathbf{y} + \lambda\mathbf{y}'\mathbf{y} + \lambda\mathbf{x}'\mathbf{x}$, from which (7) follows at once. Equality in (7) for all choices of λ holds iff $\mathbf{x} = \mathbf{y}$.

The inequalities derived here can now be used to establish an algorithm for maximizing f(T).

Maximization of f(T) via Majorization

In the present section, an algorithm is derived for minimizing $h(T) = -f(T)$ over orthonormal matrices T. Before doing so, we will first simplify the notation by defining $\mathbf{w}_l \equiv A'\mathbf{b}_l(\mathbf{b}_l'\mathbf{b}_l)^{-1/2}$, and $C \equiv A'A$. Note that, if $\mathbf{b}_l$ is a zero vector, we take $\mathbf{w}_l$ equal to a zero vector as well, because the corresponding value of ϕ is zero by definition. Then $h(T)$ can be written as

$$h(T) = -\sum_{l=1}^{r} \mathbf{w}_l'\mathbf{t}_l(\mathbf{t}_l'C\mathbf{t}_l)^{-1/2}, \tag{9}$$

where $\mathbf{w}_l'\mathbf{t}_l(\mathbf{t}_l'C\mathbf{t}_l)^{-1/2}$ is defined to be zero if $(\mathbf{t}_l'C\mathbf{t}_l) = 0$. Rather than focusing on how we can monotonically decrease $h(T)$, we will focus on decreasing the function

$$\tilde{h}(T) = -\sum_{l=1}^{r} |\mathbf{w}_l'\mathbf{t}_l|(\mathbf{t}_l'C\mathbf{t}_l)^{-1/2}. \tag{10}$$

As will be shown, the procedure for decreasing $\tilde{h}(T)$ can easily be adjusted such that it decreases $h(T)$ as well.

We will now derive a procedure for updating a current matrix T, denoted as T_c (with columns $\mathbf{t}_1^c, \ldots, \mathbf{t}_r^c$) such that $\tilde{h}(T) \leq \tilde{h}(T_c)$. We will first find a function $g(T, T_c)$ such that $g(T, T_c) \geq \tilde{h}(T)$ and $g(T_c, T_c) = \tilde{h}(T_c)$, next show that the T that minimizes $g(T, T_c)$ decreases $\tilde{h}(T)$, and finally show how the function $g(T, T_c)$ can be minimized. To find such a function $g(T, T_c)$, we will find functions that majorize each of the terms of $\tilde{h}(T)$. First, we consider the terms for which $|\mathbf{w}_l'\mathbf{t}_l^c| \neq 0$; after that we will consider the terms for which $|\mathbf{w}_l'\mathbf{t}_l^c| = 0$ (which are very unlikely to be encountered in practice).

To find a useful majorizing function for $\tilde{h}(T)$ we have to find majorizations of the terms $-|\mathbf{w}_l'\mathbf{t}_l|(\mathbf{t}_l'C\mathbf{t}_l)^{-1/2}$, $l = 1, \ldots, r$, that are (relatively) simple in terms of $\mathbf{t}_l$. Clearly, then, we need a majorization that no longer employs the complicating term $(\mathbf{t}_l'C\mathbf{t}_l)^{-1/2}$. It can be seen that Lemma 4, with $y_1 = |\mathbf{w}_l'\mathbf{t}_l|$ and $y_2 = (\mathbf{t}_l'C\mathbf{t}_l)^{1/2}$ yields such a majorization of $-|\mathbf{w}_l'\mathbf{t}_l|(\mathbf{t}_l'C\mathbf{t}_l)^{-1/2}$. Specifically, we apply Lemma 4 to each term of $\tilde{h}(T)$, for which $|\mathbf{w}_l'\mathbf{t}_l^c| \neq 0$. We define the fixed scalar $p_l \equiv (\mathbf{t}_l^{c\prime}C\mathbf{t}_l^c)$, which is positive because $p_l = 0$ implies $\mathbf{w}_l'\mathbf{t}_l^c = 0$, and apply Lemma 4 to $x_1 = |\mathbf{w}_l'\mathbf{t}_l^c|$, $x_2 = p_l^{1/2}$, $y_1 = |\mathbf{w}_l'\mathbf{t}_l|$, and $y_2 = (\mathbf{t}_l'C\mathbf{t}_l)^{1/2}$. It should be noted that Lemma 4 requires that $x_1 > 0$, $x_2 > 0$, $y_1 \geq 0$ and $y_2 > 0$. Hence Lemma 4 only holds if we make the additional assumption that $y_2 = (\mathbf{t}_l'C\mathbf{t}_l)^{1/2} > 0$; the case where $\mathbf{t}_l'C\mathbf{t}_l = 0$ will be treated separately. According to Lemma 4, see (6), we have

$$-|\mathbf{w}_l'\mathbf{t}_l|(\mathbf{t}_l'C\mathbf{t}_l)^{-1/2} \leq p_l^{-1}|\mathbf{w}_l'\mathbf{t}_l^c|(\mathbf{t}_l'C\mathbf{t}_l)^{1/2} + p_l^{-1/2}|\mathbf{w}_l'\mathbf{t}_l^c|^{-1}(\mathbf{w}_l'\mathbf{t}_l)^2 - 3p_l^{-1/2}|\mathbf{w}_l'\mathbf{t}_l| \equiv h_1(\mathbf{t}_l) + h_2(\mathbf{t}_l) + h_3(\mathbf{t}_l), \tag{11}$$

where h_1, h_2 and h_3 are defined implicitly in (11). Note that, when $\mathbf{t}_l = \mathbf{t}_l^c$, we have equality in (11), as we require of the majorization functions we use. If $\mathbf{t}_l'C\mathbf{t}_l = 0$, the derivation of (11) no longer holds, but (11) does hold: The left hand side is zero by definition; the right hand side is zero because $\mathbf{t}_l'C\mathbf{t}_l = 0$ implies that $\mathbf{w}_l'\mathbf{t}_l = 0$. Hence, we no longer have to distinguish between cases for $\mathbf{t}_l'C\mathbf{t}_l \neq 0$ and $\mathbf{t}_l'C\mathbf{t}_l = 0$. We will now majorize the functions h_1, h_2, and h_3 separately.

To majorize h_1, we apply Lemma 2 to $(\mathbf{t}_l'C\mathbf{t}_l)$. By setting $x = p_l$ ($x > 0$) and $y = \mathbf{t}_l'C\mathbf{t}_l$ ($y \geq 0$), we have, see (4),

$$(\mathbf{t}_l'C\mathbf{t}_l)^{1/2} \leq \frac{1}{2}\, p_l^{1/2} + \frac{1}{2}\, p_l^{-1/2}(\mathbf{t}_l'C\mathbf{t}_l). \tag{12}$$

Next, applying Lemma 6 to $(\mathbf{t}_l'C\mathbf{t}_l)$, we find, taking $\mathbf{x} = \mathbf{t}_l^c$, $\mathbf{y} = \mathbf{t}_l$, $Z = S = C$, and $\lambda = \rho$ (which denotes the first eigenvalue of C),

$$(\mathbf{t}_l'C\mathbf{t}_l) \leq -\mathbf{t}_l^{c\prime}C\mathbf{t}_l^c + 2\mathbf{t}_l^{c\prime}(C - \rho I)\mathbf{t}_l + 2\rho. \tag{13}$$

Combining (12) and (13) we obtain

$$\begin{aligned}(\mathbf{t}_l'C\mathbf{t}_l)^{1/2} &\leq \frac{1}{2}\, p_l^{1/2} - \frac{1}{2}\, p_l^{-1/2}\mathbf{t}_l^{c\prime}C\mathbf{t}_l^c + p_l^{-1/2}\mathbf{t}_l^{c\prime}(C - \rho I)\mathbf{t}_l + \rho p_l^{-1/2} \\ &= \frac{1}{2}\, p_l^{1/2} - \frac{1}{2}\, p_l^{-1/2}p_l + p_l^{-1/2}\mathbf{t}_l^{c\prime}(C - \rho I)\mathbf{t}_l + \rho p_l^{-1/2} \\ &= p_l^{-1/2}\mathbf{t}_l^{c\prime}(C - \rho I)\mathbf{t}_l + \rho p_l^{-1/2}. \end{aligned} \tag{14}$$

Because $p_l^{-1}|\mathbf{w}_l'\mathbf{t}_l^c| > 0$, it follows from (14) that

$$\mathrm{h}_1(\mathbf{t}_l) = p_l^{-1}|\mathbf{w}_l'\mathbf{t}_l^c|(\mathbf{t}_l'C\mathbf{t}_l)^{1/2} \leq c_1 + p_l^{-3/2}|\mathbf{w}_l'\mathbf{t}_l^c|\mathbf{t}_l^{c\prime}(C - \rho I)\mathbf{t}_l \equiv \mathrm{g}_1(\mathbf{t}_l), \tag{15}$$

where c_1 is a constant.

To majorize h_2, we apply Lemma 6 to $(\mathbf{w}_l'\mathbf{t}_l)^2 = \mathbf{t}_l'\mathbf{w}_l\mathbf{w}_l'\mathbf{t}_l$. By setting $\mathbf{x} = \mathbf{t}_l^c$, $\mathbf{y} = \mathbf{t}_l$, $Z = S = \mathbf{w}_l\mathbf{w}_l'$, and $\lambda = \mathbf{w}_l'\mathbf{w}_l$ (which is the first eigenvalue of $\mathbf{w}_l\mathbf{w}_l'$), we obtain

$$(\mathbf{w}_l'\mathbf{t}_l)^2 \leq -\mathbf{t}_l^{c\prime}\mathbf{w}_l\mathbf{w}_l'\mathbf{t}_l^c + 2\mathbf{t}_l^{c\prime}(\mathbf{w}_l\mathbf{w}_l' - \mathbf{w}_l'\mathbf{w}_l I)\mathbf{t}_l + 2\mathbf{w}_l'\mathbf{w}_l. \tag{16}$$

Hence, because $p_l^{-1/2}|\mathbf{w}_l'\mathbf{t}_l^c| > 0$,

$$\begin{aligned}\mathrm{h}_2(\mathbf{t}_l) = p_l^{-1/2}|\mathbf{w}_l'\mathbf{t}_l^c|^{-1}(\mathbf{w}_l'\mathbf{t}_l)^2 &\leq -p_l^{-1/2}|\mathbf{w}_l'\mathbf{t}_l^c|^{-1}\mathbf{t}_l^{c\prime}\mathbf{w}_l\mathbf{w}_l'\mathbf{t}_l^c \\ &\quad + 2p_l^{-1/2}|\mathbf{w}_l'\mathbf{t}_l^c|^{-1}\mathbf{t}_l^{c\prime}(\mathbf{w}_l\mathbf{w}_l' - \mathbf{w}_l'\mathbf{w}_l I)\mathbf{t}_l + 2p_l^{-1/2}|\mathbf{w}_l'\mathbf{t}_l^c|^{-1}\mathbf{w}_l'\mathbf{w}_l \\ &= c_2 + 2p_l^{-1/2}|\mathbf{w}_l'\mathbf{t}_l^c|^{-1}\mathbf{t}_l^{c\prime}(\mathbf{w}_l\mathbf{w}_l' - \mathbf{w}_l'\mathbf{w}_l I)\mathbf{t}_l \equiv \mathrm{g}_2(\mathbf{t}_l), \end{aligned} \tag{17}$$

where c_2 is a constant.

Finally, to majorize h_3, we apply Lemma 5 to $|\mathbf{w}_l'\mathbf{t}_l|$. Taking $x = \mathbf{w}_l'\mathbf{t}_l^c$, and $y = \mathbf{w}_l'\mathbf{t}_l$, we obtain

$$-|\mathbf{w}_l'\mathbf{t}_l| \leq -\mathrm{sgn}\,(\mathbf{w}_l'\mathbf{t}_l^c)\cdot\mathbf{w}_l'\mathbf{t}_l. \tag{18}$$

Hence, because $3p_l^{-1/2} > 0$,

$$\mathrm{h}_3(\mathbf{t}_l) = -3p_l^{-1/2}|\mathbf{w}_l'\mathbf{t}_l| \leq -3p_l^{-1/2}\cdot\mathrm{sgn}\,(\mathbf{w}_l'\mathbf{t}_l^c)\cdot\mathbf{w}_l'\mathbf{t}_l \equiv \mathrm{g}_3(\mathbf{t}_l). \tag{19}$$

Above, we have found functions that majorize h_1, h_2, and h_3, respectively, and hence jointly majorize the l-th term of $\tilde{\mathrm{h}}(T)$ if $|\mathbf{w}_l'\mathbf{t}_l^c| \neq 0$. The terms for which $|\mathbf{w}_l'\mathbf{t}_l^c| = 0$, on the other hand, must be majorized in a different way. A simple way is to use the constant function $\mathrm{k}(\mathbf{t}_l) = 0$, with equality if $\mathbf{t}_l = \mathbf{t}_l^c$. This follows at once from the fact that $-|\mathbf{w}_l'\mathbf{t}_l|(\mathbf{t}_l'C\mathbf{t}_l)^{-1/2} \leq 0$. Hence, upon combination of the latter result with (11), (15), (17), and (19), it follows that

$$\tilde{h}(T) = -\sum_{l=1}^{r} |\mathbf{w}_l'\mathbf{t}_l|(\mathbf{t}_l'C\mathbf{t}_l)^{-1/2} \leq \sum_{l\in A} h_1(\mathbf{t}_l) + \sum_{l\in A} h_2(\mathbf{t}_l) + \sum_{l\in A} h_3(\mathbf{t}_l) \leq \sum_{l\in A} g_1(\mathbf{t}_l)$$

$$+ \sum_{l\in A} g_2(\mathbf{t}_l) + \sum_{l\in A} g_3(\mathbf{t}_l) \equiv g(T, T_c), \quad (20)$$

where $\Sigma_{l\in A}$ denotes that the summation is only over the values of l for which $|\mathbf{w}_l'\mathbf{t}_l^c| \neq 0$. Thus, we have established a function $g(T, T_c)$ that majorizes $\tilde{h}(T)$.

It has been mentioned above that, in order to be useful for updating T, the majorizing function should be such that $g(T_c, T_c) = \tilde{h}(T_c)$. That this is indeed the case can be verified for the terms for which $|\mathbf{w}_l'\mathbf{t}_l^c| \neq 0$, by noting that the inequalities (11), (12), (13), (16) and (18), which were used in the derivation of (20), turn into equalities if we substitute $\mathbf{t}_l = \mathbf{t}_l^c$; for the terms for which $|\mathbf{w}_l'\mathbf{t}_l^c| = 0$, equality (with both sides zero) follows trivially. Therefore, if we find the update T as the orthonormal matrix that minimizes $g(T, T_c)$, we have $\tilde{h}(T) \leq g(T, T_c) \leq g(T_c, T_c) = \tilde{h}(T_c)$. Hence this T decreases the function $\tilde{h}$. We will now show how this T can be found. Next, it will be shown how this T can be adjusted such that it decreases the original function h as well.

It will now be described how we obtain the minimum of $g(T, T_c)$ over orthonormal T. The function $g(T, T_c)$ can be written as

$$g(T, T_c) = c + \sum_{l\in A} p_l^{-3/2}|\mathbf{w}_l'\mathbf{t}_l^c|\mathbf{t}_l^{c\prime}(C - \rho I)\mathbf{t}_l + 2\sum_{l\in A} p_l^{-1/2}|\mathbf{w}_l'\mathbf{t}_l^c|^{-1}\mathbf{t}_l^{c\prime}(\mathbf{w}_l\mathbf{w}_l'$$

$$- \mathbf{w}_l'\mathbf{w}_l I)\mathbf{t}_l - 3\sum_{l\in A} p_l^{-1/2} \cdot \operatorname{sgn}(\mathbf{w}_l'\mathbf{t}_l^c) \cdot \mathbf{w}_l'\mathbf{t}_l = c + \sum_{l=1}^{r} \mathbf{u}_l'\mathbf{t}_l$$

$$= c + \operatorname{tr} U'T, \quad (21)$$

with $\mathbf{u}_l$, the l-th column of U, defined as

$$\mathbf{u}_l \equiv p_l^{-3/2}|\mathbf{w}_l'\mathbf{t}_l^c|(C\mathbf{t}_l^c - \rho\mathbf{t}_l^c) + 2p_l^{-1/2}|\mathbf{w}_l'\mathbf{t}_l^c|^{-1}(\mathbf{w}_l\mathbf{w}_l'\mathbf{t}_l^c - \mathbf{w}_l'\mathbf{w}_l\mathbf{t}_l^c)$$

$$- 3p_l^{-1/2} \cdot \operatorname{sgn}(\mathbf{w}_l'\mathbf{t}_l^c) \cdot \mathbf{w}_l, \quad (22)$$

if $|\mathbf{w}_l'\mathbf{t}_l^c| \neq 0$, and $\mathbf{u}_l = \mathbf{0}$ if $|\mathbf{w}_l'\mathbf{t}_l^c| = 0$, $l = 1, \ldots, r$. Note that the choice $\mathbf{u}_l = \mathbf{0}$ implies that $\mathbf{t}_l$ can be chosen arbitrarily, as long as it does not affect the orthonormality of T. As explained below, we will always arrange that $\mathbf{w}_l'\mathbf{t}_l^c \geq 0$, $l = 1, \ldots, r$. Using this, we can write (22) as

$$\mathbf{u}_l \equiv p_l^{-3/2}(\mathbf{w}_l'\mathbf{t}_l^c)(C\mathbf{t}_l^c - \rho\mathbf{t}_l^c) - 2p_l^{-1/2}(\mathbf{w}_l'\mathbf{t}_l^c)^{-1}\mathbf{w}_l'\mathbf{w}_l\mathbf{t}_l^c - p_l^{-1/2}\mathbf{w}_l. \quad (23)$$

The problem of minimizing (21) has been solved by Cliff (1966). The matrix T that minimizes $g(T, T_c)$ subject to the constraint that $T'T = I$, can be obtained from the singular value decomposition (SVD) of U, $U = PDQ'$ as $T = -PQ'$. Hence, by updating T in this way, we will decrease $\tilde{h}(T)$.

Rather than decreasing $\tilde{h}(T)$ monotonically, we wish to decrease $h(T)$ monotonically. This can be arranged as follows. We can always start the procedure of updating T_c by reflecting all columns of T_c for which $\mathbf{w}_l'\mathbf{t}_l^c < 0$, because this does not affect the value of $\tilde{h}(T_c)$, and can only *decrease* the value of $h(T_c)$. Then this reflection makes sure that $\tilde{h}(T_c) = h(T_c)$. Next, we update T_c by the procedure described above, such that, for the update T, we have $\tilde{h}(T) \leq \tilde{h}(T_c) = h(T_c)$. If for the update T we have $\mathbf{w}_l'\mathbf{t}_l \geq 0$ for $l = 1, \ldots, r$, we have $\tilde{h}(T) = h(T)$ and hence $h(T) \leq h(T_c)$. If $\mathbf{w}_l'\mathbf{t}_l < 0$ for

certain columns of T, we reflect these columns, and for the resulting T_r we have $h(T_r) < \tilde{h}(T) \leq \tilde{h}(T_c) = h(T_c)$. Thus, by reflecting the relevant columns of the update T we can always make sure that the original function h is decreased. By iteratively updating T in the above described way, we decrease h monotonically. Because the function is bounded below, the algorithm must converge to a stable function value. In the Appendix it is proven, under mild assumptions, that, $(T^{i+1} - T^i)$ converges to 0 and that T satisfies the stationary equations at convergence.

For the above algorithm, it has been proven that $(T^{i+1} - T^i)$ converges to 0, and that this implies that either T converges to a continuum or to a local maximum, a local minimum or a saddle point. However, the fact that the algorithm monotonically increases $f(T)$ causes that it is very unlikely that the algorithm stops in a local minimum or even a saddle point. Convergence to a continuum of points seems rather unlikely as well, and has not been encountered in our test analyses. Hence, the algorithm will usually converge to at least a local maximum. Nothing guarantees that this local maximum will be the global maximum of the function f. By using different starting configurations, it is hoped that the global maximum of f will be found. In a later section we report some results based on simulated data which indicate that the algorithm finds the global maximum very often indeed. First, however, we give a schematic overview of the above derived algorithm.

Schematic Overview of the Algorithm

The basic step in the present algorithm is described in (23) and below. The other steps are merely initializations and definitions. The algorithm can be summarized as follows:

Step 1. $W := A'B(\text{Diag}\,(B'B))^{-1/2}$ (with columns $\mathbf{w}_l$).
Step 2. $C := A'A$.
Step 3. $\rho :=$ largest eigenvalue of C.
Step 4. Choose T_c (as an orthonormal initialization of T); if $W'T_c$ has negative diagonal elements, multiply the corresponding columns of T_c by -1.
Step 5. $f := \text{tr}W'T_c(\text{Diag}\,(T_c'CT_c))^{-1/2}$.
Step 6. $f^{\text{old}} := f$.
for $l := 1$ to r
Step 7a. $p_l := \mathbf{t}_l^{c\prime} C \mathbf{t}_l^c$;
Step 7b. $q_l := \mathbf{w}_l' \mathbf{t}_l^c$;
Step 7c. if $q_l \neq 0$, $\mathbf{u}_l := p_l^{-3/2} q_l (C\mathbf{t}_l^c - \rho \mathbf{t}_l^c) - 2p_l^{-1/2} q_l^{-1} \mathbf{w}_l' \mathbf{w}_l \mathbf{t}_l^c - p_l^{-1/2} \mathbf{w}_l$;
if $q_l = 0$, $\mathbf{u}_l := \mathbf{0}$;
Step 8. Obtain P and Q from the SVD $U = PDQ'$.
Step 9. $T := -PQ'$.
Step 10. If $W'T$ has negative diagonal elements, multiply the corresponding columns of T by -1.
Step 11. $f := \text{tr}W'T(\text{Diag}\,(T'CT))^{-1/2}$.
Step 12. If $f < f^{\text{old}} + \varepsilon_1^* |f|$ and, if desired, $\|T - T_c\| < \varepsilon_2$, where ε_1 and ε_2 are small positive constants, consider the algorithm converged; else $T_c := T$ and go to Step 6.

Performance of the Algorithm

The algorithm has been tested on 80 simulated data sets, each consisting of a matrix A and a matrix B. We constructed these data by taking B equal to a random

$n \times r$ matrix and A was computed as $A = BDT' + \delta N$, where D was a fixed diagonal matrix (to be described below), T was a random orthonormal matrix, δ was zero in half the conditions (the perfect match conditions) and $\delta = 1$ in the other (imperfect match) conditions, and, finally, N denotes a random $n \times r$ matrix. The random matrices were constructed from elements drawn from the uniform distribution on the interval $(-.5, .5)$; in case of T, the obtained random matrix was orthonormalized. The matrices A, B and N were of two different orders, 10×3, and 20×6. We used essentially two different choices for D, one with values 1, 1.5 and 2, and the other with values 1, 3, and 6; for the $r = 6$ cases each of these values occurred twice in D. According to this design, we created 80 data sets (10 replications in each condition). We applied the algorithm described above to each data set, using 20 different random starting configurations, as well as the "rational" start obtained from the Procrustes rotation. So for each data set the algorithm was run 21 times. We considered the algorithm converged if changes of the function value dropped below .0001% ($\varepsilon_1 = 10^{-6}$), which can be considered a fairly strict convergence criterion. For the perfectly matching data, the global maximum is known in advance; for the imperfect case, the best of the 21 runs was considered to yield the global maximum. Function values smaller than the (alleged) global maximum minus .001 were considered local optima. Our main interest is in studying how often the algorithm misses the global maximum. In addition, we wanted to gain some insight in computation times and numbers of iterations.

The algorithm has been programmed in PCMATLAB and tested on a personal computer with 486-intel processor. It was first studied how often the algorithm found the global maximum. In Table 1, we report the percentage of runs that hit the global maximum, both for the randomly and rationally started runs. These "hit rates" have been reported separately for each condition, as well as averaged across conditions. First of all, it can be seen that 79 of the 80 rationally started runs converged to the global maximum, indicating that it is indeed rational to use the "rational" start. That the rational starts are very good indeed can be seen from Table 2, where average maximal function values and average function values at the rational start are reported for all conditions. It can also be seen that, in the perfect match condition, the rational start always led to the known maximal function value of 3 when $r = 3$, and 6 when $r = 6$, respectively. In all conditions more than 50% of the randomly started runs converged to the global maximum. The hit rates for the smaller data were considerably larger than those for the larger data ($\chi^2 = 17.6$, $p < .001$). No other significant differences were found, hence there is little or no reason to expect that data with certain particularly unfavorable properties (other than large size) lead to large numbers of local minima.

The results for average time and average numbers of iterations are of course closely related. We give both (see Table 1) because computation times may not be informative for use at other machines or with other programming languages. As far as computation time is concerned it can be concluded that rationally started runs are considerably faster than randomly started runs ($F = 53.8$, $p < .001$), small data sets could be analyzed much faster than large data sets ($F = 915.2$, $p < .001$), and data where the columns of AT and B had widely different importances (D with values 1, 3, 6) converged much more slowly than those with mildly differing importances ($F = 300.3$, $p < .001$). No significant difference was found between "perfect" and "imperfect" data, which is convenient, since in practice we have no possibilities whatsoever to manipulate this aspect of the data.

It can be concluded that the method is well behaved in that it almost always finds the global maximum when it is started rationally, and finds the global maximum at least every other time when it is started randomly. Moreover, computation times are fairly small considering that a 20×6 problem with mildly differing importances took less than

TABLE 1

Hit Rates, Average Computation Times (in seconds) and Average Numbers of Iterations for Simulated Data

			Random Start			Rational Start		
Data Type	$n \times r$	D	Hits	Time	Iter.	Hits	Time	Iter.
Perfect	10×3	(1,1½,2)	60.0%	1.8	58.7	100%	0.7	20.6
	10×3	(1,3,6)	63.5%	2.6	84.6	100%	1.3	42.0
	20×6	(1,1½,2)	56.0%	7.1	127.9	100%	1.5	23.6
	20×6	(1,3,6)	54.5%	18.0	323.6	100%	3.6	61.4
Imperfect	10×3	(1,1½,2)	75.0%	1.7	48.5	90%	0.7	17.8
	10×3	(1,3,6)	57.0%	3.8	115.8	100%	1.4	39.5
	20×6	(1,1½,2)	51.0%	8.1	134.9	100%	1.8	26.2
	20×6	(1,3,6)	51.5%	17.8	284.9	100%	4.8	67.9
Average			58.6%	7.6	147.4	99%	2.0	37.4

two seconds on the average, when started rationally, and less than 8 seconds when started randomly. A series of 21 runs (which is usually more than enough) still would take less than three minutes.

We also programmed Brokken's Newton based algorithm, and applied it to all 80 data sets. It was soon observed that using a random start made the algorithm require very many iterations before it landed in a stationary point. Moreover, in our test runs this point never was the global maximum and often not even a local maximum. Therefore, we only studied the algorithm systematically using the rational start. When started rationally, Brokken's algorithm found the global maximum in 38 (out of 40) of the cases with little difference in importances, but only in 4 (out of 40) of the cases with large differences in importances. This can be explained by the extremely good fit of the rational starts in the conditions with little difference in importances and the somewhat poorer starting fits in the other conditions (see Table 2). Whenever Brokken's algorithm found the global maximum, it found it in very few iterations (2 to 7). This does not imply that Brokken's algorithm is faster than the majorization based one, because the iterations in Brokken's algorithm are considerably more time consuming, especially for large data sets. In our comparison, Brokken's algorithm was slightly faster than the majorization based one in the first 10×3 condition, but it was slower in the other conditions, even though it converged in fewer iterations. However, as mentioned before, comparisons in computation time may depend heavily on the programming language, so these results can only serve to give a rough indication. Moreover, a compar-

TABLE 2

Average Maximal Function Values and Average Function Values at the Rational Start for Simulated Data

			Function Value	
Data Type	$n \times r$	D	Maximum	At Rational Start
Perfect	10×3	$(1,1\frac{1}{2},2)$	3.00	2.99
	10×3	(1,3,6)	3.00	2.85
	20×6	$(1,1\frac{1}{2},2)$	6.00	5.97
	20×6	(1,3,6)	6.00	5.58
Imperfect	10×3	$(1,1\frac{1}{2},2)$	2.49	2.48
	10×3	(1,3,6)	2.76	2.68
	20×6	$(1,1\frac{1}{2},2)$	5.13	5.12
	20×6	(1,3,6)	5.37	5.22

ison of computation time is not our main interest. Nonconvergence, or failure to find the global maximum is more important. It turned out that, if Brokken's algorithm missed the global maximum the algorithm used very many iterations before it found a stationary point, and in 9 cases it had not found a stationary point after 1000 iterations (which was the maximum we used). To sum up, Brokken's algorithm works well in cases with extremely good starts, although even then it is not necessarily faster than the majorization based algorithm. In cases where the rational start is relatively poor, the Newton procedure is very unreliable, and the majorization based algorithm is to be preferred.

Discussion

The majorization based algorithm proposed in the present paper turned out to perform well in all cases considered. It was found that the rational start almost always led to the global optimum. Indeed, in many cases the rational start already was quite good, especially in the cases with relatively small differences in importances. In the latter cases, Brokken's Newton based procedure was unproblematic, and converged in a few iterations, although not necessarily faster than the majorization based algorithm. For such cases one might prefer to use the majorization based algorithm for other reasons, like the ease of programming, or the fact that it can be started fruitfully from different starts and hence can be used for checking for local optimality. In cases where

importances differed much (1, 3, 6), the Newton procedure hardly ever found the global maximum. Hence, it can be concluded that only the majorization based algorithm can be advocated for general use. Moreover, the algorithm is easy to implement in any matrix language.

The present paper provides a useful alternative *algorithm* for Brokken's method. The present paper is not concerned with the usefulness of Brokken's *criterion*. By offering a better algorithm, however, it facilitates a systematic comparison between, for instance, Brokken's criterion and Koschat and Swayne's criterion.

Brokken (1985) has extended his method to the case where more than two matrices are rotated simultaneously to maximal congruence. Let $A_1, \ldots, A_p$ denote p matrices of order $n \times r$. Then his method aims at maximizing

$$\mathrm{F}(T_1, \ldots, T_p) = \sum_{i=1}^{p} \sum_{j>i}^{p} \sum_{l=1}^{r} \phi(A_i \mathbf{t}_{il}, A_j \mathbf{t}_{jl}), \tag{24}$$

over $T_1, \ldots, T_p$, which are arbitrary orthonormal matrices. His algorithm for maximizing F is based on alternately updating one of the orthonormal matrices, T_i say, considering the other orthonormal matrices fixed. As is readily verified (also, see Brokken, 1985), the latter problem reduces to the problem of maximizing (1) with A replaced by A_i and B by $\Sigma_{j \neq i}^{p} A_j T_j$. Rather than using the Brokken (1983) procedure, we propose to use our new procedure for finding each T_i. This procedure will monotonically increase F, because each update of T_i increases F over T_i considering the $T_j (j = 1, \ldots, p, j \neq i)$ fixed.

The six Lemmas mentioned here have proven most useful in deriving a majorizing function for the present problem. Some of these have been used in other contexts (Groenen & Heiser, 1991; Groenen, 1993; Kiers, 1995) as well. It seems likely that these Lemmas may be of use to derive majorizing functions for other minimization problems, which was one reason for stating and proving them extensively. The present paper has demonstrated that the combination of such inequalities can lead to powerful results for the construction of majorization based algorithms.

Appendix

In the present paper, an algorithm has been proposed that decreases the function $\tilde{\mathrm{h}}$ monotonically, and that, because $\tilde{\mathrm{h}}$ is bounded below, converges to a stable function value. Let T^i denote the matrix T at iteration i, then this result implies that $\mathrm{Lim}_{i \to \infty} \tilde{\mathrm{h}}(T^i) = \mathrm{h}^*$, for a certain value h^*. In the present appendix, it will be shown that, under some mild assumptions, this result also implies that the difference between $(T^i - T^{i+1})$ converges to zero, and that T^i satisfies the stationary equations as $i \to \infty$. One of these assumptions is that $\mathrm{Lim}_{i \to \infty} (\mathbf{w}_l' \mathbf{t}_l^i) \neq 0$ for all l. The other will be mentioned where it is used.

To prove that $(T^i - T^{i+1})$ converges to zero, we use the monotonicity of the algorithm (that is, $\mathrm{h}(T^{i+1}) \leq \mathrm{h}(T^i)$) and the facts that $\mathrm{k}(T, T^i) \equiv \Sigma_l (\mathrm{h}_1(\mathbf{t}_l) + \mathrm{h}_2(\mathbf{t}_l) + \mathrm{h}_3(\mathbf{t}_l))$ majorizes $\tilde{\mathrm{h}}(T)$, see (20), and that $\mathrm{k}(T, T^i)$ in turn is majorized by $\mathrm{g}(T, T^i) \equiv \Sigma_l (\mathrm{g}_1(\mathbf{t}_l) + \mathrm{g}_2(\mathbf{t}_l) + \mathrm{g}_3(\mathbf{t}_l))$, with equality for $T = T^i$. Note that summations are over all l because $\mathbf{w}_l' \mathbf{t}_l^i \neq 0$ for all l by assumption. Thus we have the sequence

$$\tilde{\mathrm{h}}(T^{i+1}) \leq \mathrm{k}(T^{i+1}, T^i) \leq \mathrm{g}(T^{i+1}, T^i) \leq \mathrm{g}(T^i, T^i) = \tilde{\mathrm{h}}(T^i). \tag{25}$$

From $\mathrm{Lim}_{i \to \infty} \tilde{\mathrm{h}}(T^i) = \mathrm{h}^*$ it follows that $\mathrm{Lim}_{i \to \infty} (\tilde{\mathrm{h}}(T^{i+1}) - \tilde{\mathrm{h}}(T^i)) = 0$, hence

$$\lim_{i\to\infty} (k(T^{i+1}, T^i) - g(T^{i+1}, T^i)) = 0, \tag{26}$$

and because $h_1(t_l^{i+1}) - g_1(t_l^{i+1}) \leq 0$, $h_2(t_l^{i+1}) - g_2(t_l^{i+1}) \leq 0$, and $h_3(t_l^{i+1}) - g_3(t_l^{i+1}) \leq 0$, for all l, it follows that, among others,

$$\lim_{i\to\infty} (h_1(t_l^{i+1}) - g_1(t_l^{i+1})) = 0. \tag{27}$$

From (14), multiplied by $p_l^{-1}|w_l't_l^i|$, and the definition of h_1 and g_1 in (15), we have, with $p_l \equiv (t_l^{i\prime}Ct_l^i)$,

$$\begin{aligned} h_1(t_l^{i+1}) - g_1(t_l^{i+1}) &= p_l^{-1}|w_l't_l^i|((t_l^{i+1\prime}Ct_l^{i+1})^{1/2} - p_l^{-1/2}t_l^{i\prime}(C-\rho I)t_l^{i+1} - \rho p_l^{-1/2}) \\ &= \frac{1}{2}p_l^{-3/2}|w_l't_l^i|(-(p_l^{1/2} - (t_l^{i+1\prime}Ct_l^{i+1})^{1/2})^2 \\ &\quad + (t_l^{i+1} - t_l^i)'(C-\rho I)(t_l^{i+1} - t_l^i)). \end{aligned} \tag{28}$$

Clearly, $p_l^{-3/2}$ is strictly nonzero and does not tend to zero, because this would require p_l to tend to ∞. Because, in addition, $\lim_{i\to\infty} |w_l't_l^i| \neq 0$, and the two additive terms in (28) are both nonpositive, it follows from (27) that $\lim_{i\to\infty} (p_l^{1/2} - (t_l^{i+1\prime}Ct_l^{i+1})^{1/2}) = 0$ and $\lim_{i\to\infty} (t_l^{i+1} - t_l^i)'(C-\rho I)(t_l^{i+1} - t_l^i) = 0$. From the latter it follows that $\lim_{i\to\infty} (t_l^{i+1} - t_l^i) = \mathbf{0}$, or $\lim_{i\to\infty} (t_l^{i+1} - t_l^i)$ is in the null space of $(C - \rho I)$.

Similarly, it follows from (26) that

$$\lim_{i\to\infty} (h_2(t_l^{i+1}) - g_2(t_l^{i+1})) = 0. \tag{29}$$

From (17) we have

$$\begin{aligned} h_2(t_l^{i+1}) - g_2(t_l^{i+1}) &= p_l^{-1/2}|w_l't_l^i|^{-1}((w_l't_l^{i+1})^2 + t_l^{i\prime}w_lw_l't_l^i \\ &\quad - 2t_l^{i\prime}(w_lw_l' - w_l'w_lI)t_l^{i+1} - 2w_l'w_l) \\ &= p_l^{-1/2}|w_l't_l^i|^{-1}((t_l^{i+1} - t_l^i)'(w_lw_l' - (w_l'w_l)I)(t_l^{i+1} - t_l^i)). \end{aligned} \tag{30}$$

Because $p_l^{-1/2}$ and $|w_l't_l^i|^{-1}$ are strictly nonzero and do not tend to zero, it follows from (29) and (30) that

$$\lim_{i\to\infty} (t_l^{i+1} - t_l^i)'(w_lw_l' - (w_l'w_l)I)(t_l^{i+1} - t_l^i) = 0. \tag{31}$$

Hence $\lim_{i\to\infty} (t_l^{i+1} - t_l^i) = \mathbf{0}$, or $\lim_{i\to\infty} (t_l^{i+1} - t_l^i)$ is in the null space of $(w_lw_l' - (w_l'w_l)I)$ and therefore proportional to w_l.

In practice, the combination of these results implies that $\lim_{i\to\infty} (t_l^{i+1} - t_l^i) = \mathbf{0}$. The alternatives could hold only when w_l happens to be in the null space of $C - \rho I$, which can safely be assumed not to be the case in practice. Thus it has been proven that, under our assumptions, $\lim_{i\to\infty} (T_l^{i+1} - t_l^i) = \mathbf{0}$ for all l.

It has now been proven that $\lim_{i\to\infty} (T^{i+1} - T^i) = 0$. According to Ostrowski (1969, p. 203) this implies that T converges to a single limit point or a continuum of limit points. It will next be proven that for this point (or this continuum of points) the stationary equation for the problem of minimizing $h(T)$ subject to $T'T = I$ is satisfied. In other words, it will be proven that the stationary equation is satisfied as $i \to \infty$. From

the fact that T^{i+1} minimizes $g(T, T^i)$ in (21), it follows that, writing U^i for the matrix U based on T^i,

$$U^i = T^{i+1}\Theta^i \tag{32}$$

for a negative semidefinite matrix Θ^i. The equation for the l-th columns of the left- and right-hand side of (32) is given by

$$(\mathbf{t}_l^{i\prime}C\mathbf{t}_l^i)^{-3/2}(\mathbf{w}_l'\mathbf{t}_l^i)(C\mathbf{t}_l^i - \rho\mathbf{t}_l^i) - 2(\mathbf{t}_l^{i\prime}C\mathbf{t}_l^i)^{-1/2}(\mathbf{w}_l'\mathbf{t}_l^i)^{-1}\mathbf{w}_l'\mathbf{w}_l\mathbf{t}_l^i - (\mathbf{t}_l^{i\prime}C\mathbf{t}_l^i)^{-1/2}\mathbf{w}_l = T^{i+1}\boldsymbol{\theta}_l^i, \tag{33}$$

where $\boldsymbol{\theta}_l^i$ is the l-th column of Θ^i. Because $\operatorname{Lim}_{i\to\infty}(\mathbf{t}_l^{i+1} - \mathbf{t}_l^i) = \mathbf{0}$, we have

$$\begin{aligned}\operatorname*{Lim}_{i\to\infty} &((\mathbf{t}_l^{i\prime}C\mathbf{t}_l^i)^{-3/2}(\mathbf{w}_l'\mathbf{t}_l^i)(C\mathbf{t}_l^i - \rho\mathbf{t}_l^i) - 2(\mathbf{t}_l^{i\prime}C\mathbf{t}_l^i)^{-1/2}(\mathbf{w}_l'\mathbf{t}_l^i)^{-1}\mathbf{w}_l'\mathbf{w}_l\mathbf{t}_l^i \\ &- (\mathbf{t}_l^{i\prime}C\mathbf{t}_l^i)^{-1/2}\mathbf{w}_l - T^i\boldsymbol{\theta}_l^i) \\ = \operatorname*{Lim}_{i\to\infty} &(-(\mathbf{t}_l^{i\prime}C\mathbf{t}_l^i)^{-1/2}\mathbf{w}_l + (\mathbf{t}_l^{i\prime}C\mathbf{t}_l^i)^{-3/2}(\mathbf{w}_l'\mathbf{t}_l^i)C\mathbf{t}_l^i - \rho(\mathbf{t}_l^{i\prime}C\mathbf{t}_l^i)^{-3/2}(\mathbf{w}_l'\mathbf{t}_l^i)\mathbf{t}_l^i \\ &- 2(\mathbf{t}_l^{i\prime}C\mathbf{t}_l^i)^{-1/2}(\mathbf{w}_l'\mathbf{t}_l^i)^{-1}\mathbf{w}_l'\mathbf{w}_l\mathbf{t}_l^i - T^i\boldsymbol{\theta}_l^i) = \mathbf{0}.\end{aligned} \tag{34}$$

The stationary equations for $h(T)$ are given by $T'T = I$ (which is satisfied for all T^i) and

$$-(\mathbf{t}_l'C\mathbf{t}_l)^{-1/2}\mathbf{w}_l + (\mathbf{w}_l'\mathbf{t}_l)(\mathbf{t}_l'C\mathbf{t}_l)^{-3/2}C\mathbf{t}_l - T\boldsymbol{\phi}_l = \mathbf{0}, \tag{35}$$

$l = 1, \ldots, r$, where $\boldsymbol{\phi}_l$ is the l-th column of a symmetric Lagrange multiplier matrix Φ. Upon defining the symmetric matrix Φ^i with l-th column

$$\boldsymbol{\phi}_l^i \equiv \boldsymbol{\theta}_l^i + (\rho(\mathbf{t}_l^{i\prime}C\mathbf{t}_l^i)^{-3/2}(\mathbf{w}_l'\mathbf{t}_l^i) + 2(\mathbf{t}_l^{i\prime}C\mathbf{t}_l^i)^{-1/2}(\mathbf{w}_l'\mathbf{t}_l^i)^{-1}\mathbf{w}_l'\mathbf{w}_l)\mathbf{e}_l, \tag{36}$$

where $\mathbf{e}_l$ is the l-th column of the identity matrix, we have from (34)

$$\operatorname*{Lim}_{i\to\infty}(-(\mathbf{t}_l^{i\prime}C\mathbf{t}_l^i)^{-1/2}\mathbf{w}_l + (\mathbf{t}_l^{i\prime}C\mathbf{t}_l^i)^{-3/2}(\mathbf{w}_l'\mathbf{t}_l^i)C\mathbf{t}_l^i - T^i\boldsymbol{\phi}_l^i) = \mathbf{0}. \tag{37}$$

This implies that, in the limit where $i \to \infty$, the stationary equation (35) is satisfied. Thus it has been proven that the stationary equations are satisfied at convergence. It follows that, if the algorithm converges to a single point (rather than a continuum), then this accumulation point must be a local maximum, a local minimum, or a saddle point.

The present proof relies on the assumption that $\mathbf{w}_l'\mathbf{t}_l^i \neq 0$ for all l. In practice, we have seen no violations of this assumption. If it would be violated, we could use an alternative algorithm that avoids this problem, but this alternative is much slower than the one presented in the paper. Because there seems no practical urge to use this algorithm, we have ignored it in the present paper.

References

Brokken, F.B. (1983). Orthogonal Procrustes rotation maximizing congruence. *Psychometrika, 48*, 343–352.

Brokken, F.B. (1985). The simultaneous maximization of congruence for two or more matrices under orthogonal rotation. *Psychometrika, 50*, 51–56.

Cliff, N. (1966). Orthogonal rotation to congruence. *Psychometrika, 31*, 33–42.

de Leeuw, J., & Heiser, W. (1980). Multidimensional scaling with restrictions on the configuration. In P. R. Krishnaiah (Ed.), *Multivariate Analysis V* (pp. 501–522). Amsterdam: North Holland.

Goodall, C. (1991). Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society, Series B, 53*, 285–339.

Green, B.F. (1952). The orthogonal approximation of an oblique structure in factor analysis. *Psychometrika, 17*, 429–440.

Groenen, P.J.F. (1993). *The majorization approach to multidimensional scaling: Some problems and extensions*. Leiden: DSWO Press.

Groenen, P.J.F., & Heiser, W.J. (1991). *An improved tunnelling function for finding a decreasing series of local minima in MDS* (Research Report RR-91-06). Leiden: Department of Data Theory.

Heiser, W.J. (1987). Correspondence Analysis with least absolute residuals. *Computational Statistics and Data Analysis, 5*, 337–356.

Kiers, H.A.L. (1990). Majorization as a tool for optimizing a class of matrix functions. *Psychometrika, 55*, 417–428.

Kiers, H.A.L. (1995). Maximization of sums of quotients of quadratic forms and some generalizations. *Psychometrika, 60*, 221–245.

Korth, B., & Tucker, L.R. (1976). Procrustes matching by congruence coefficients. *Psychometrika, 41*, 531–535.

Koschat, M.A., & Swayne, D.F. (1991). A weighted Procrustes criterion. *Psychometrika, 56*, 229–239.

Meulman, J.J. (1986). *A distance approach to nonlinear multivariate analysis*. Leiden: DSWO Press.

Ostrowski, A.M. (1969). *Solutions of equations and systems of equations*. New York: Academic Press.

ten Berge, J.M.F. (1986). Some relationships between descriptive comparisons of components from different studies. *Multivariate Behavioral Research, 21*, 29–40.

Tucker, L.R. (1951). *A method for synthesis of factor analysis studies* (Personnel Research Section Report No. 984). Washington DC: Department of the Army.

Manuscript received 11/8/93
Final version received 1/19/95