PSYCHOMETRIKA—VOL. 88, NO. 1, 98–116
MARCH 2023
https://doi.org/10.1007/s11336-022-09891-5

A NOTE ON THE CONNECTION BETWEEN TREK RULES AND SEPARABLE NONLINEAR LEAST SQUARES IN LINEAR STRUCTURAL EQUATION MODELS

MAXIMILIAN S. ERNST

MAX PLANCK INSTITUTE FOR HUMAN DEVELOPMENT

HUMBOLDT-UNIVERSITÄT ZU BERLIN

AARON PEIKERT

MAX PLANCK INSTITUTE FOR HUMAN DEVELOPMENT

HUMBOLDT-UNIVERSITÄT ZU BERLIN

MAX PLANCK UCL CENTRE FOR COMPUTATIONAL PSYCHIATRY AND AGEING RESEARCH

ANDREAS M. BRANDMAIER

MAX PLANCK INSTITUTE FOR HUMAN DEVELOPMENT

MAX PLANCK UCL CENTRE FOR COMPUTATIONAL PSYCHIATRY AND AGEING RESEARCH

MSB MEDICAL SCHOOL BERLIN

YVES ROSSEEL

GHENT UNIVERSITY

We show that separable nonlinear least squares (SNLLS) estimation is applicable to all linear structural equation models (SEMs) that can be specified in RAM notation. SNLLS is an estimation technique that has successfully been applied to a wide range of models, for example neural networks and dynamic systems, often leading to improvements in convergence and computation time. It is applicable to models of a special form, where a subset of parameters enters the objective linearly. Recently, Kreiberg et al. (Struct Equ Model Multidiscip J 28(5):725–739, 2021. https://doi.org/10.1080/10705511.2020.1835484) have shown that this is also the case for factor analysis models. We generalize this result to all linear SEMs. To that end, we show that undirected effects (variances and covariances) and mean parameters enter the objective linearly, and therefore, in the least squares estimation of structural equation models, only the directed effects have to be obtained iteratively. For model classes without unknown directed effects, SNLLS can be used to analytically compute least squares estimates. To provide deeper insight into the nature of this result, we employ trek rules that link graphical representations of structural equation models to their covariance parametrization. We further give an efficient expression for the gradient, which is crucial to make a fast implementation possible. Results from our simulation indicate that SNLLS leads to improved convergence rates and a reduced number of iterations.

Key words: Gaussian graphical model, graph theory, numerical optimization, least squares estimation, RAM notation.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11336-022-09891-5.

This article is based on the master's thesis of Ernst (2022).

Correspondence should be made to Maximilian S. Ernst, Center for Lifespan Psychology, Max Planck Institute for Human Development, Lentzeallee 94, 14195 Berlin, Germany. Email: ernst@mpib-berlin.mpg.de

© 2022 The Author(s)

https://doi.org/10.1007/s11336-022-09891-5 Published online by Cambridge University Press

In the behavioral and social sciences, structural equation models (SEMs) have become widely accepted as a multivariate statistical tool for modeling the relation between latent and observed variables. Apart from maximum likelihood estimation, least squares (LS) estimation is a common approach for parameter estimation. In LS, parameters are estimated by minimizing a nonlinear function of the parameters and data. In practice, this problem is typically solved by applying generic nonlinear optimization techniques, such as Newton-type gradient descent approaches that iteratively minimize the objective function until convergence is reached. However, for some model classes, generic optimization algorithms can be adapted to make better use of the model structure and thus solve the problem more efficiently. For a particular type of models, the parameters *separate*, that is, one set of parameters enters the objective in a nonlinear way, while another set of parameters enters the objective linearly. For a vector of observations y and predictors x of size m, the objective is of the form

$$\sum_{i=1}^{m} \left[y_i - \sum_{j=1}^{n} \alpha_j \varphi_j(\beta, x_i) \right]^2 \tag{1}$$

where $\alpha \in \mathbb{R}^n$, $\beta \in \mathbb{R}^k$ are parameter vectors and the (nonlinear) functions φ_j are continuously differentiable w.r.t. β. Golub and Pereyra (1973) showed that this kind of objective allows for a reformulation of the optimization problem, such that only the parameters β have to be obtained iteratively, while the parameters α can be computed after the optimization in a single step. The procedure has been subsequently called separable nonlinear least squares (SNLLS). It has been successfully applied in many disciplines, and it has been observed that the reduced dimension of the parameter space can lead to reduced computation time, a reduced number of iterations and better convergence properties (Golub & Pereyra, 2003). Inspired by earlier work (Kreiberg et al., 2016, 2021) that showed that this procedure can also be applied to factor analysis models, we generalize their result to the entire class of linear structural equation models and give analytical gradients for the reduced optimization problem, which is central for an efficient implementation.

1. Review of Concepts

In the following, we briefly review the notation for structural equation models, the generalized least squares estimator and the trek rules used to derive the model-implied covariance matrix.

1.1. Linear Structural Equation Models

Linear structural equation models can be defined in RAM notation (reticular action model; McArdle & McDonald, 1984) as follows (we follow the notation from Drton, 2018): Let x, ε be random vectors with values in $\mathbb{R}^m$ and

$$x = \mathbf{\Lambda} x + \varepsilon \tag{2}$$

where $\mathbf{\Lambda} \in \mathbb{R}^{m \times m}$ is a matrix of constants or unknown (directed) parameters. Let $\mathbf{\Omega} \in \mathbb{R}^{m \times m}$ be the covariance matrix of ε and $\mathbf{I}$ the identity matrix. If $\mathbf{I} - \mathbf{\Lambda}$ is invertible, Eq. 2 can be solved by $x = (\mathbf{I} - \mathbf{\Lambda})^{-1} \varepsilon$ with covariance matrix

$$\mathbb{V}[x] = (\mathbf{I} - \mathbf{\Lambda})^{-1} \, \mathbf{\Omega} \, (\mathbf{I} - \mathbf{\Lambda})^{-T} \tag{3}$$

https://doi.org/10.1007/s11336-022-09891-5 Published online by Cambridge University Press

If x is partitioned into a part x_{obs} of m_{obs} observed variables and x_{lat} of m_{lat} latent variables, we can reorder x such that $x = \left(x_{\text{obs}}^T \; x_{\text{lat}}^T\right)^T$, and the covariance matrix of the observed variables is given by

$$\boldsymbol{\Sigma} := \mathbb{V}[x_{\text{obs}}] = \mathbf{F}(\mathbf{I} - \boldsymbol{\Lambda})^{-1}\boldsymbol{\Omega}(\mathbf{I} - \boldsymbol{\Lambda})^{-T}\mathbf{F}^T \tag{4}$$

where $\mathbf{F} = \left[\mathbf{I} \,\middle|\, \mathbf{0}\right] \in \mathbb{R}^{m_{\text{obs}} \times (m_{\text{obs}} + m_{\text{lat}})}$ is a rectangular filter matrix. We denote the parameters by $\theta = \left(\theta_\Lambda^T \Omega_\Lambda^T\right)^T \in \mathbb{R}^q$, partitioned into directed parameters from $\boldsymbol{\Lambda}$ and undirected parameters from $\boldsymbol{\Omega}$. (We call them directed or undirected parameters because they correspond to directed or undirected paths in the graph of the model.) If we want to stress that $\boldsymbol{\Sigma}$ is a function of the parameters, we write $\boldsymbol{\Sigma}(\theta)$. If we are also interested in the mean structure, we introduce a vector of (possibly zero) mean parameters $\gamma \in \mathbb{R}^m$ such that $x = \gamma + \boldsymbol{\Lambda}x + \varepsilon$ and obtain

$$\mu := \mathbb{E}[x_{obs}] = \mathbf{F}(\mathbf{I} - \boldsymbol{\Lambda})^{-1}\gamma \tag{5}$$

1.2. Least Squares Estimation

The least squares objective function for θ is:

$$F_{\text{LS}} = (s - \sigma)^T \mathbf{V}(s - \sigma) \tag{6}$$

where $\sigma = \text{vech}(\boldsymbol{\Sigma})$ is the half-vectorization of $\boldsymbol{\Sigma}$, that is, the vector of non-duplicated elements of the model-implied covariance matrix, $s = \text{vech}(\mathbf{S})$ is the half-vectorization of the observed covariance matrix and $\mathbf{V}$ is a fixed symmetric positive definite weight matrix. Specific forms of the weight matrix $\mathbf{V}$ lead to commonly used special cases of this estimation technique: Generalized least squares estimation uses $\mathbf{V} = \frac{1}{2}\mathbf{D}^T\left(\mathbf{S}^{-1} \otimes \mathbf{S}^{-1}\right)\mathbf{D}$ (where $\mathbf{D}$ denotes the duplication matrix from Magnus and Neudecker (2019b)), asymptotic distribution-free estimation uses a consistent estimator of the asymptotic covariance matrix of s, and unweighted least squares estimation uses the identity matrix (Bollen, 1989; Browne, 1982, 1984).

1.3. Trek Rules

To show that in SEM undirected effects enter the least squares objective linearly, we employ trek rules (Drton, 2018), which are path tracing rules used to derive the model-implied covariance between any pair of variables in a SEM (Boker et al., 2002). Various authors have proposed rules to link the graph to the covariance parametrization of the model. Here, we give the rules as put forward by Drton (2018), which are based on *treks* as basic building blocks (for an overview of alternative formulations see Mulaik, 2009). A trek τ from a node i to j is a path connecting them, where directed edges can be traveled forwards and backwards, but it is not allowed to walk from one arrowhead into another (without *colliding arrowheads*). A *top node* of a trek is a node which has only outgoing edges.

To derive an expression for the model-implied covariance between any two variables i and j based on the postulated SEM, we follow 4 steps:

1. Find all treks **from** i **to** j.
2. For each trek, multiply all parameters along it.
3. If a trek does not contain a covariance parameter, factor in the variance of the top node.
4. Add all obtained *trek monomials* from the different treks together.

https://doi.org/10.1007/s11336-022-09891-5 Published online by Cambridge University Press

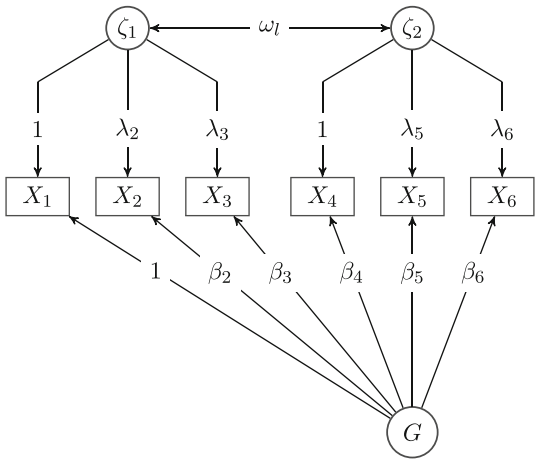


FIGURE 1.
Graph of a bi-factor model with one general factor and two specific factors. Circles represent latent variables, and rectangles represent observed variables. Variances are omitted in this representation.

Note that a trek is ordered in the sense that two treks containing exactly the same nodes and edges are considered different if they are traveled in a different order. In particular, each trek has a *source* (i) and a *target* (j), and a trek from j to i is considered to be a different trek, even if it contains exactly the same nodes and edges. Also note that variances are not considered to be edges in the mixed graph corresponding to the model (i.e., it is not allowed to travel variance edges). Therefore, all graphical representations of SEMs in this article omit variance edges, and it is required to factor them in according to rule 3 after the treks are collected.

1.3.1. Example To illustrate how the model-implied covariances can be derived using trek rules, we give an example based on the graph shown in the path diagram in Fig. 1. To find the model-implied covariance between nodes X_2 and X_6 in the model shown in Fig. 1, we first find all treks from X_2 to X_6

$$X_2 \overset{\lambda_2}{\longleftarrow} \zeta_1 \overset{\omega_l}{\longleftrightarrow} \zeta_2 \overset{\lambda_6}{\longrightarrow} X_6 \tag{7}$$

$$X_2 \overset{\beta_2}{\longleftarrow} G \overset{\beta_6}{\longrightarrow} X_6 \tag{8}$$

We now compute the trek monomials for each trek. The second trek does not contain a covariance parameter, so we need to factor in the variance of the top node. We find the trek's top node G and denote the variance parameter of G by ω_G. Finally, we add the resulting trek monomials and we find that the model-implied covariance between X_2 and X_6 can be expressed as follows:

$$\text{cov}(X_2, X_6) = \lambda_2 \omega_l \lambda_6 + \beta_2 \omega_G \beta_6 \tag{9}$$

As a second example, we derive the model-implied variance of X_3. Again, we first find all treks from X_3 to X_3:

$$X_3 \overset{\lambda_3}{\longleftarrow} \zeta_1 \overset{\lambda_3}{\longrightarrow} X_3 \tag{10}$$

https://doi.org/10.1007/s11336-022-09891-5 Published online by Cambridge University Press

$$X_3 \xleftarrow{\beta_3} G \xrightarrow{\beta_3} X_3 \tag{11}$$

$$X_3 \tag{12}$$

All treks do not contain a covariance parameter, so we need to factor in the variance of the respective top nodes ζ_1, G and X_3. We denote the variance parameters of ζ_1 and X_3 by ω_{ζ_1} and ω_3 and add the resulting trek monomials to obtain

$$\text{var}(X_3) = \lambda_3^2 \omega_{\zeta_1} + \beta_3^2 \omega_G + \omega_3 \tag{13}$$

1.3.2. Formal Definitions We denote the elements of $\mathbf{\Omega}$, the undirected effects between nodes i and j, by ω_{ij} and the elements of $\mathbf{\Lambda}$, the directed effects, by λ_{ij}. Drton (2018) defines a *trek monomial* of a trek τ without a covariance parameter as

$$\tau(\mathbf{\Lambda}, \mathbf{\Omega}) = \omega_{i_0 i_0} \prod_{k \to l \in \tau} \lambda_{lk} \tag{14}$$

where i_0 is the top node of the trek, and a trek monomial of a trek τ containing an undirected edge between i_0 and j_0 as

$$\tau(\mathbf{\Lambda}, \mathbf{\Omega}) = \omega_{i_0 j_0} \prod_{k \to l \in \tau} \lambda_{lk} \tag{15}$$

(notice the swapped indices of λ_{lk} compared to the formula in Drton because our $\mathbf{\Lambda}$ corresponds to his $\mathbf{\Lambda}^T$). With this, the elements of $\mathbf{\Sigma}(\theta)$ are represented as a summation over treks. He proves that

$$\mathbf{\Sigma}(\theta)_{ij} = \sum_{\tau \in \mathcal{T}(i,j)} \tau(\mathbf{\Lambda}, \mathbf{\Omega}) \tag{16}$$

where $\mathcal{T}(i, j)$ is the set of all treks **from** i **to** j. It follows that the model-implied covariance is a sum of monomials of parameters. Because covariances between the error terms are not transitive, exactly one undirected parameter (variance or covariance) is present in each monomial. Therefore, if all the directed parameters were fixed, the model-implied covariance would be a linear function of the undirected parameters. This is what makes the SNLLS procedure applicable to structural equation models.

For later use, we also note that Drton gives the following expression:

$$(\mathbf{I} - \mathbf{\Lambda})^{-1}_{ij} = \sum_{\tau \in \mathcal{P}(j,i)} \prod_{k \to l \in \tau} \lambda_{lk} \tag{17}$$

where $\mathcal{P}(j, i)$ is the set of directed paths from j to i. This is because we can write $(\mathbf{I} - \mathbf{\Lambda})^{-1} = \sum_{k=0}^{\infty} \mathbf{\Lambda}^k$, where the geometric series converges iff all eigenvalues of $\mathbf{\Lambda}$ lie in $(-1, 1)$. (Further explanations about this and an excellent account of the connections between matrix algebra and graphs can be found in Kepner & Gilbert (2011).)

https://doi.org/10.1007/s11336-022-09891-5 Published online by Cambridge University Press

2. Separable Nonlinear Least Squares for SEM

We first outline the proofs for the applicability of SNLLS to CFA as given by Golub and Pereyra (1973) and Kreiberg et al. (2021). Subsequently, we proof that SNLLS is applicable to linear structural equation models. We further extend the existing proofs to subsume models that contain a mean structure. Last, we derive analytic gradients that are central for efficient software implementations.

2.1. Outline of Previous Work

To minimize Eq. 1, Golub and Pereyra (1973) define the matrix function

$$\Phi_{ij} := \varphi_j(\beta, x_i) \tag{18}$$

such that Eq. 1 can be written as

$$\|y - \Phi(\beta)\alpha\|^2 \tag{19}$$

where $\|\cdot\|$ denotes the euclidean norm. For a fixed value of β, a solution for α can be obtained as $\alpha = \Phi^+(\beta)y$. They further proved that under the assumption that $\Phi(\beta)$ has constant rank near the solution, only the nonlinear parameters β have to be obtained iteratively by replacing α and minimizing the modified objective

$$\left\|y - \Phi(\beta)\Phi^+(\beta)y\right\|^2 \tag{20}$$

where Φ^+ denotes the Moore–Penrose generalized inverse. Afterward, the least squares solution for the linear parameters α can be obtained as the standard least squares estimator $\arg\min_{\alpha\in\mathbb{R}^n}\|\Phi(\hat{\beta})\alpha - y\| = \Phi^+(\hat{\beta})y$.

Kreiberg et al. (2021) showed that this procedure is applicable for CFA models (we reproduce their main results in our notation), as it is possible to rewrite the model-implied covariances σ as a product of a matrix-valued function $\mathbf{G}(\theta_{\mathbf{\Lambda}})$ (that depends only on the directed parameters) and the undirected parameters $\theta_{\mathbf{\Omega}}$, so the LS objective can be written as

$$\begin{aligned} F_{\text{LS}} &= (s-\sigma)^T\mathbf{V}(s-\sigma) && (21)\\ &= (s - \mathbf{G}(\theta_{\mathbf{\Lambda}})\theta_{\mathbf{\Omega}})^T\,\mathbf{V}\,(s - \mathbf{G}(\theta_{\mathbf{\Lambda}})\theta_{\mathbf{\Omega}}) && (22)\\ &= \|s - \mathbf{G}(\theta_{\mathbf{\Lambda}})\theta_{\mathbf{\Omega}}\|^2_{\mathbf{V}} && (23)\end{aligned}$$

They further stated that if $\theta_{\mathbf{\Lambda}}$ is fixed, we know from standard linear least squares estimation that the minimizer for the undirected effects can be obtained as

$$\hat{\theta}_{\mathbf{\Omega}} = \left(\mathbf{G}(\theta_{\mathbf{\Lambda}})^T\mathbf{V}\mathbf{G}(\theta_{\mathbf{\Lambda}})\right)^{-1}\mathbf{G}(\theta_{\mathbf{\Lambda}})^T\mathbf{V}s \tag{24}$$

Inserting Eq. 24 into Eq. 22 and simplifying, they obtained a new objective to be minimized:

$$\begin{aligned} \hat{\theta}_{\mathbf{\Lambda}} &= \arg\min_{\theta_{\mathbf{\Lambda}}}\left[s^T\mathbf{V}s - s^T\mathbf{V}\mathbf{G}(\theta_{\mathbf{\Lambda}})\left(\mathbf{G}(\theta_{\mathbf{\Lambda}})^T\mathbf{V}\mathbf{G}(\theta_{\mathbf{\Lambda}})\right)^{-1}\mathbf{G}(\theta_{\mathbf{\Lambda}})^T\mathbf{V}s\right] && (25)\\ &= \arg\min_{\theta_{\mathbf{\Lambda}}} F_{\text{SNLLS}}(\theta_{\mathbf{\Lambda}}) && (26)\end{aligned}$$

https://doi.org/10.1007/s11336-022-09891-5 Published online by Cambridge University Press

This objective only depends on the directed parameters $\theta_{\mathbf{\Lambda}}$. After minimizing it to obtain a LS estimate $\hat{\theta}_{\mathbf{\Lambda}}$, Eq. 24 can be used to obtain the LS estimate of $\theta_{\mathbf{\Omega}}$. We would like to note they assume that $\mathbf{G}$ has full rank, which is not a necessary assumption and can be relaxed using alternative formulations of Eqs. 24 and 25. To extend the method to general structural equation models, we have to derive $\mathbf{G}(\theta_{\mathbf{\Lambda}})$. We do that in the following for all models formulated in the RAM notation.

2.2. *Derivation of* $\mathbf{G}(\theta_{\mathbf{\Lambda}})$

Since $\mathbf{F} = \left[\mathbf{I} \,\middle|\, \mathbf{0}\right]$ with $\mathbf{0} \in \mathbb{R}^{m_{\text{obs}} \times m_{\text{lat}}}$, the product $\mathbf{FMF}^T$ for any $\mathbf{M} \in \mathbb{R}^{m \times m}$ is equal to just deleting the last m_{lat} rows and columns of $\mathbf{M}$. We also note that for any matrices $\mathbf{M}, \mathbf{D} \in \mathbb{R}^{n \times n}$ we can write

$$\left(\mathbf{MDM}^T\right)_{ij} = \sum_{k=1}^{n} \sum_{l=1}^{n} m_{il}\, d_{lk}\, m_{jk} \tag{27}$$

With this in mind, we can rewrite the model-implied covariance matrix $\mathbf{\Sigma}(\theta)$ as

$$\begin{aligned}
\mathbf{\Sigma}(\theta)_{ij} &= \Big(\mathbf{F} \quad (\mathbf{I} - \mathbf{\Lambda})^{-1} \quad \mathbf{\Omega} \quad (\mathbf{I} - \mathbf{\Lambda})^{-T} \quad \mathbf{F}^T\Big)_{ij} && (28)\\
&= \Big(\quad (\mathbf{I} - \mathbf{\Lambda})^{-1} \quad \mathbf{\Omega} \quad (\mathbf{I} - \mathbf{\Lambda})^{-T} \quad\Big)_{ij} && (29)\\
&= \sum_{k=1}^{m} \sum_{l=1}^{m} (\mathbf{I} - \mathbf{\Lambda})^{-1}_{il} \quad \omega_{lk} \quad (\mathbf{I} - \mathbf{\Lambda})^{-1}_{jk} && (30)\\
&= \sum_{k=1}^{m} \sum_{l=1}^{m} \Big(\sum_{\tau \in \mathcal{P}(l,i)} \prod_{r \to s \in \tau} \lambda_{sr}\Big) \quad \omega_{lk} \quad \Big(\sum_{\tau \in \mathcal{P}(k,j)} \prod_{r \to s \in \tau} \lambda_{sr}\Big) && (31)
\end{aligned}$$

with $i, j \in \{1, \ldots, m_{\text{obs}}\}$. We now immediately see that each entry of $\mathbf{\Sigma}$ is a sum of products of entries of $(\mathbf{I} - \mathbf{\Lambda})^{-1}$ and $\mathbf{\Omega}$. More importantly, exactly one entry of $\mathbf{\Omega}$ enters each term of the sum; if we keep all entries of $\mathbf{\Lambda}$ fixed, each element in $\mathbf{\Sigma}$ is a linear function of the entries of $\mathbf{\Omega}$ and is therefore a linear function of the undirected parameters in $\mathbf{\Omega}$ (under the assumption that $\mathbf{\Omega}$ is linearly parameterized). As a result, the parameter vector θ is separable in two parts, $\theta_{\mathbf{\Lambda}}$ from $\mathbf{\Lambda}$ and $\theta_{\mathbf{\Omega}}$ from $\mathbf{\Omega}$, and $\theta_{\mathbf{\Omega}}$ enters the computation of the model-implied covariance linearly. As stated before, this is the reason why we will be able to apply separable nonlinear least squares estimation to our problem. Before we proceed, we would like to introduce some notation. If $\mathcal{F}$ and $\mathcal{G}$ are tuples of length n and m, and f and g are functions, we define a column vector of length n as

$$\left(\left[f(i)\right]_{i \in \mathcal{F}}\right) = \begin{pmatrix} f(\mathcal{F}_1) \\ f(\mathcal{F}_2) \\ \cdots \\ f(\mathcal{F}_n) \end{pmatrix} \tag{32}$$

https://doi.org/10.1007/s11336-022-09891-5 Published online by Cambridge University Press

and a matrix of size $n \times m$ as

$$\left(\left[g(i,\, j)\right]_{i\in\mathcal{F},\; j\in\mathcal{G}}\right) = \begin{pmatrix} g(\mathcal{F}_1, \mathcal{G}_1) & g(\mathcal{F}_1, \mathcal{G}_2) & \dots & g(\mathcal{F}_1, \mathcal{G}_m) \\ g(\mathcal{F}_2, \mathcal{G}_1) & g(\mathcal{F}_2, \mathcal{G}_2) & \dots & g(\mathcal{F}_2, \mathcal{G}_m) \\ \dots & \dots & \dots & \dots \\ g(\mathcal{F}_n, \mathcal{G}_1) & \dots & \dots & g(\mathcal{F}_n, \mathcal{G}_m) \end{pmatrix} \tag{33}$$

To make the subsequent steps easier to follow, we assume that there are no equality constraints between parameters in $\mathbf{\Omega}$ and no constant terms in $\mathbf{\Omega}$ different from 0. In Appendices A and B, we show how to lift those assumptions. We now further simplify Eq. 30: Since only nonzero entries of $\mathbf{\Omega}$ (the parameters $\theta_{\mathbf{\Omega}}$) contribute to the sum, we define $\mathcal{C}$ as the lower triangular indices of $\theta_{\mathbf{\Omega}}$ in $\mathbf{\Omega}$, i.e., $\mathcal{C}_i = (l, k) \in \mathbb{N} \times \mathbb{N}$ with $(\theta_{\mathbf{\Omega}})_i = \omega_{lk}$ and $l \geq k$. We now rewrite Eq. 30 by omitting all zero terms:

$$\mathbf{\Sigma}(\theta)_{ij} = \sum_{(l,k)\in\mathcal{C}} \left[(\mathbf{I}-\mathbf{\Lambda})^{-1}_{il}\omega_{lk}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{jk} + \delta_{k\neq l}\,(\mathbf{I}-\mathbf{\Lambda})^{-1}_{ik}\omega_{lk}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{jl}\right] \tag{34}$$

$$= \left(\left[(\mathbf{I}-\mathbf{\Lambda})^{-1}_{il}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{jk} + \delta_{k\neq l}\,(\mathbf{I}-\mathbf{\Lambda})^{-1}_{ik}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{jl}\right]_{(l,k)\in\mathcal{C}}\right)^{T} \theta_{\mathbf{\Omega}} \tag{35}$$

where $\delta_{k\neq l}$ is an indicator function that takes the value 1 if $k \neq l$ and 0 otherwise. Since we are only interested in the non-duplicated elements σ of $\mathbf{\Sigma}$, we define another index tuple $\mathcal{D}$ that denotes the indices of the original position of σ_k in $\mathbf{\Sigma}$, i.e., $\mathcal{D}_k = (i,\, j)$ such that $\sigma_k = \mathbf{\Sigma}_{ij}$. This allows us to stack the expression we just found for $\mathbf{\Sigma}_{ij}$ rowwise to get

$$\sigma = \left(\left[\mathbf{\Sigma}_{ij}\right]_{(i,j)\in\mathcal{D}}\right) \tag{36}$$

$$= \left(\left[(\mathbf{I}-\mathbf{\Lambda})^{-1}_{il}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{jk} + \delta_{k\neq l}\,(\mathbf{I}-\mathbf{\Lambda})^{-1}_{ik}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{jl}\right]_{(i,j)\in\mathcal{D},\;(l,k)\in\mathcal{C}}\right)\theta_{\mathbf{\Omega}} \tag{37}$$

$$= \mathbf{G}(\theta_{\mathbf{\Lambda}})\;\theta_{\mathbf{\Omega}} \tag{38}$$

where $\mathbf{G}(\theta_{\mathbf{\Lambda}}) \in \mathbb{R}^{\dim(\sigma)\times\dim(\theta_{\mathbf{\Omega}})}$. (We let $\dim(\cdot)$ of a vector denote its number of elements, i.e., the dimension of the underlying (finite-dimensional) vector space.)

Even though this expression may appear involved, it is in fact easy to compute. Before the optimization procedure starts, we store $\mathcal{C}$ by looking up the positions of the parameters in $\mathbf{\Omega}$ and also store $\mathcal{D}$. At each step of the optimization procedure, to compute $\mathbf{G}(\theta_{\mathbf{\Lambda}})$, we now compute $(\mathbf{I}-\mathbf{\Lambda})^{-1}$ first and then loop through the entries $\mathcal{C}$ and $\mathcal{D}$ to compute each entry of $\mathbf{G}(\theta_{\mathbf{\Lambda}})$ according to Eq. 37. We note that $\mathbf{G}$ will typically be sparse; therefore, it is advisable to analyze its sparsity pattern previous to the optimization, and only loop through nonzero values.

In Appendix D, we present a different way of obtaining $\mathbf{G}(\theta_{\mathbf{\Lambda}})$ and the gradients, which mimics the approach of Kreiberg et al. (2021). However, the expressions obtained here are computationally more efficient, as the ones in the appendix contain very large Kronecker products.

2.3. Mean Structures

If the model contains mean parameters, we partition the parameter vector θ into three parts: $\theta_{\mathbf{\Lambda}}$ and $\theta_{\mathbf{\Omega}}$ as before, and θ_{γ} from the mean vector γ. From Eq. 5, we directly see that the model-implied mean vector $\mu(\theta)$ is a linear function of θ_{γ}. If we let $\mathcal{A}$ denote the indices of the parameters θ_{γ} in γ, i.e., for $i = \mathcal{A}_j$ we have $(\theta_{\gamma})_j = \gamma_i$, we obtain the formula

https://doi.org/10.1007/s11336-022-09891-5 Published online by Cambridge University Press

$$\mu = \left(\left[(\mathbf{I} - \mathbf{\Lambda})^{-1}_{ij} \right]_{i \in (1,\ldots,m_{\text{obs}}),\ j \in \mathcal{A}} \right) \theta_\gamma \tag{39}$$

We now make a slight change in notation: For the previously obtained $\mathbf{G}(\theta)$-matrix, we write $\mathbf{G}_\sigma$ instead and define $\mathbf{G}_\mu := \left(\left[(\mathbf{I} - \mathbf{\Lambda})^{-1}_{ij} \right]_{i \in (1,\ldots,m_{\text{obs}}),\ j \in \mathcal{A}} \right)$. Using a formulation of the least squares objective that also includes a mean structure, we see that

$$\begin{aligned} F_{\text{LS}} &= \left[\begin{pmatrix} s \\ m \end{pmatrix} - \begin{pmatrix} \sigma \\ \mu \end{pmatrix} \right]^T \mathbf{V} \left[\begin{pmatrix} s \\ m \end{pmatrix} - \begin{pmatrix} \sigma \\ \mu \end{pmatrix} \right] \\ &= \left[\begin{pmatrix} s \\ m \end{pmatrix} - \begin{pmatrix} \mathbf{G}_\sigma \theta_{\mathbf{\Omega}} \\ \mathbf{G}_\mu \theta_\gamma \end{pmatrix} \right]^T \mathbf{V} \left[\begin{pmatrix} s \\ m \end{pmatrix} - \begin{pmatrix} \mathbf{G}_\sigma \theta_{\mathbf{\Omega}} \\ \mathbf{G}_\mu \theta_\gamma \end{pmatrix} \right] \\ &= \left[\begin{pmatrix} s \\ m \end{pmatrix} - \mathbf{G} \begin{pmatrix} \theta_{\mathbf{\Omega}} \\ \theta_\gamma \end{pmatrix} \right]^T \mathbf{V} \left[\begin{pmatrix} s \\ m \end{pmatrix} - \mathbf{G} \begin{pmatrix} \theta_{\mathbf{\Omega}} \\ \theta_\gamma \end{pmatrix} \right] \end{aligned} \tag{40}$$

with

$$\mathbf{G} := \begin{bmatrix} \mathbf{G}_\sigma & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_\mu \end{bmatrix} \tag{41}$$

It follows that in addition to the undirected parameters, the mean parameters also do not have to be optimized iteratively but can instead be computed analytically after the iterative optimization is completed.

2.4. Gradient of the SNLLS Objective

There are computationally efficient expression to compute the SNLLS objective and its gradient analytically (Kaufman, 1975; O'Leary & Rust, 2013). Because numerical approximations of the gradient are often slow and may become numerically unstable, we derive an analytical expression for the part of the gradient that is specific to SEMs. We use the notation and methods from Magnus and Neudecker (2019a) and denote the differential by d and the Jacobian by D. The Jacobian of a matrix function $\mathbf{M}$ with respect to a vector x is defined as $\mathsf{D}\,\mathbf{M} = \frac{\partial \operatorname{vec} \mathbf{M}}{\partial x^T}$. In the approaches by Kaufman (1975) and O'Leary and Rust (2013), the gradient of the SNLSS objective is expressed in terms of the partial derivatives of the entries of $\mathbf{G}$ w.r.t the nonlinear parameters, i.e., $\mathsf{D}\,\mathbf{G}$. In order to be able to implement such efficient approaches in practice, we derive $\mathsf{D}\,\mathbf{G}$ here. We also give the full gradient of Eq. 25 for completeness in Appendix C, although in practice, a more efficient expression from the cited literature can be used (which also does not assume $\mathbf{G}$ to have full rank). For reasons of clarity, we here only consider the case without mean structure, e.g., $\mathbf{G} = \mathbf{G}_\sigma$. This is because the derivative of $\mathbf{G}_\mu$ is similar to obtain and we do not want to make the derivation unnecessarily technical.

Let $\mathcal{E}$ denote the indices of $\theta_{\mathbf{\Lambda}}$ in $\mathbf{\Lambda}$, i.e., $\mathcal{E}_k = (i, j)$ such that $\mathbf{\Lambda}_{ij} = (\theta_{\mathbf{\Lambda}})_k$. We note that

$$\frac{\partial (\mathbf{I} - \mathbf{\Lambda})^{-1}_{kl}}{\partial \mathbf{\Lambda}_{ij}} = (\mathbf{I} - \mathbf{\Lambda})^{-1}_{ki} (\mathbf{I} - \mathbf{\Lambda})^{-1}_{jl} \tag{42}$$

https://doi.org/10.1007/s11336-022-09891-5 Published online by Cambridge University Press

With this, we derive the partial derivatives of each entry of $\mathbf{G}$ in terms of the matrix $(\mathbf{I}-\mathbf{\Lambda})^{-1}$ as

$$\frac{\partial \mathbf{G}_{r,s}}{\partial(\theta_{\mathbf{\Lambda}})_n} = \frac{\partial}{\partial(\theta_{\mathbf{\Lambda}})_n}\left[(\mathbf{I}-\mathbf{\Lambda})^{-1}_{il}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{jk} + \delta_{k\neq l}\,(\mathbf{I}-\mathbf{\Lambda})^{-1}_{ik}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{jl}\right] \tag{43}$$

$$= \left[\frac{\partial}{\partial(\theta_{\mathbf{\Lambda}})_n}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{il}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{jk} + (\mathbf{I}-\mathbf{\Lambda})^{-1}_{il}\frac{\partial}{\partial(\theta_{\mathbf{\Lambda}})_n}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{jk}\right]$$
$$+ \delta_{k\neq l}\left[\frac{\partial}{\partial(\theta_{\mathbf{\Lambda}})_n}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{ik}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{jl} + (\mathbf{I}-\mathbf{\Lambda})^{-1}_{ik}\frac{\partial}{\partial(\theta_{\mathbf{\Lambda}})_n}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{jl}\right] \tag{44}$$

$$= \left[(\mathbf{I}-\mathbf{\Lambda})^{-1}_{iu}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{vl}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{jk} + (\mathbf{I}-\mathbf{\Lambda})^{-1}_{il}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{ju}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{vk}\right]$$
$$+ \delta_{k\neq l}\left[(\mathbf{I}-\mathbf{\Lambda})^{-1}_{iu}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{vk}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{jl} + (\mathbf{I}-\mathbf{\Lambda})^{-1}_{ik}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{ju}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{vl}\right] \tag{45}$$

with $(i, j) = \mathcal{D}_r$, $(l, k) = \mathcal{C}_s$, and $(u, v) = \mathcal{E}_n$. Since $\mathbf{G}$ is of dimension $\dim(\sigma) \times \dim(\theta_{\mathbf{\Omega}})$, with $k = \dim(\sigma)$ we have

$$\operatorname{vec}(\mathbf{G})_t = \mathbf{G}_{t-k\lfloor(t-1)/k)\rfloor,\,\lceil t/k\rceil} \tag{46}$$

and we obtain $\mathsf{D}\,\mathbf{G} \in \mathbb{R}^{\dim(\sigma)\dim(\theta_{\mathbf{\Omega}})\times\dim(\theta_{\mathbf{\Lambda}})}$ as

$$\mathsf{D}\,\mathbf{G} = \frac{\partial \operatorname{vec}\mathbf{G}}{\partial\theta^T_{\mathbf{\Lambda}}} = \left(\left[\frac{\partial \mathbf{G}_{t-k\lfloor(t-1)/k)\rfloor,\,\lceil t/k\rceil}}{\partial(\theta_{\mathbf{\Lambda}})_n}\right]_{t\in(1,\ldots,\dim(\sigma)\dim(\theta_{\mathbf{\Omega}})),\; n\in(1,\ldots,\dim(\theta_{\mathbf{\Lambda}}))}\right) \tag{47}$$

To facilitate software implementation, we give a way to compute $\mathsf{D}\,\mathbf{G}$ in pseudocode in Algorithm 1. In practice, $\mathsf{D}\,\mathbf{G}$ will typically contain many zero values. Therefore, it is advisable to analyze the sparsity pattern of $\mathsf{D}\,\mathbf{G}$ before the optimization procedure begins and to only compute the nonzero values of $\mathsf{D}\,\mathbf{G}$ at each iteration. Also note that the entries of $\mathsf{D}\,\mathbf{G}$ are continuous w.r.t $\theta_{\mathbf{\Lambda}}$, since they are sums of products of entries of the inverse $(\mathbf{I}-\mathbf{\Lambda})^{-1}$, which is continuous.

Algorithm 1
Compute $\mathsf{D}\,\mathbf{G}$

for $s \in (1, \ldots, \dim(\theta_{\mathbf{\Omega}}))$ **do**
  $(l, k) \leftarrow \mathcal{C}_s$
  for $r \in (1, \ldots, \dim(\sigma))$ **do**
    $(i, j) \leftarrow \mathcal{D}_r$
    $t \leftarrow (s-1)\dim(\sigma) + r$
    for $n \in (1, \ldots, \dim(\theta_{\mathbf{\Lambda}}))$ **do**
      $(u, v) \leftarrow \mathcal{E}_n$
      $\mathsf{D}\,\mathbf{G}_{t,n} \leftarrow \left[(\mathbf{I}-\mathbf{\Lambda})^{-1}_{iu}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{vl}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{jk} + (\mathbf{I}-\mathbf{\Lambda})^{-1}_{il}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{ju}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{vk}\right]$
        $+\delta_{k\neq l}\left[(\mathbf{I}-\mathbf{\Lambda})^{-1}_{iu}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{vk}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{jl} + (\mathbf{I}-\mathbf{\Lambda})^{-1}_{ik}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{ju}(\mathbf{I}-\mathbf{\Lambda})^{-1}_{vl}\right]$
    end for
  end for
end for

https://doi.org/10.1007/s11336-022-09891-5 Published online by Cambridge University Press

3. Discussion

We have shown that separable nonlinear least squares is applicable to generalized least squares estimation of structural equation models formulated in the RAM notation. We have also shown a connection to path tracing rules in the form of trek rules. Note that when the same weight matrix is used, the point estimates obtained by SNLLS and LS are identical. Therefore, standard errors and test statistics are obtained using the same methods available for regular least squares estimation. In the following, we would like to discuss the two major benefits of using SNLLS for SEM: better convergence properties and a reduction in the computation time for parameter estimation.

3.1. Convergence

An important issue in SEM is convergence problems, especially in small samples (De Jonckere & Rosseel, 2022). If the optimizer fails to converge, no parameter estimates can be obtained. Using the SNLLS objective should lead to fewer convergence problems than LS, since only the directed parameters need to be estimated iteratively. Therefore, only the subset of directed parameters requires starting values. In many models, most of the directed parameters are factor loadings, and we can obtain very good starting values for them with the FABIN 3 estimator (Hägglund, 1982). Also, Ruhe and Wedin (1980) and Golub and Pereyra (2003) give additional proofs and reasons for why the reduced optimization problem of SNLLS should in principle be better behaved than the full LS problem. Additionally, for the class of models without unknown directed parameters, convergence problems should be eliminated altogether, as the estimator of the mean and (co)variance parameters can be computed analytically. Most prominently, this features many types of latent growth curve models.

To investigate the convergence properties of SNLLS in SEM, we ran a small simulation. We used the model in Fig. 2 to draw 1000 random data sets for varying sample sizes (N = 10 to N = 100) under the assumption of multivariate normality with zero expectation and the model-implied covariance induced by the parameters. The sample size and the factor loadings are deliberately chosen to be small to achieve a setting where non-convergence often occurs. We fitted the true model to each sample with generalized least squares (GLS; Bollen, 1989) and SNLLS estimation. All analyses were done in the programming language R (R Core Team, 2021). For GLS estimation, we used lavaan (Rosseel, 2012). The plots were created with ggplot2 (Wickham, 2016), and the data were prepared with dplyr (Wickham et al., 2021). In Fig. 3 we report the number of converged models for each sample size. In Fig. 4, we report the median number of iterations needed until convergence for each sample size. Using SNLLS effectively halved the median number of iterations until convergence for most sample sizes and more than halved the number of non-converged models for most sample sizes. This indicates that SNLLS might be a useful alternative for applied researchers to consider if they encounter convergence problems.

3.2. Computation Time

The benefits of SNLLS estimation, specifically the reduced dimensionality of the parameter space, better starting values and fewer iterations to convergence, could lead to reduced computation times. However, the computation of the SNLLS objective function and gradient is also more costly, so the cost per iteration can be higher. In sum, the question whether SNLLS estimation is faster in actual time spent in the optimization hinges upon several aspects, such as the actual implementation of the gradient, meta-parameters of the optimizer and model complexity.

Kreiberg et al. (2021) stated that estimation by SNLLS will typically be multiple times faster than LS when the reduced parameter space is much smaller than the original one. They conducted a simulation study, where they fitted a number of CFA models and concluded that the estimation time is bigger for LS than for SNLLS as the number of estimated parameters increases. Even

https://doi.org/10.1007/s11336-022-09891-5 Published online by Cambridge University Press

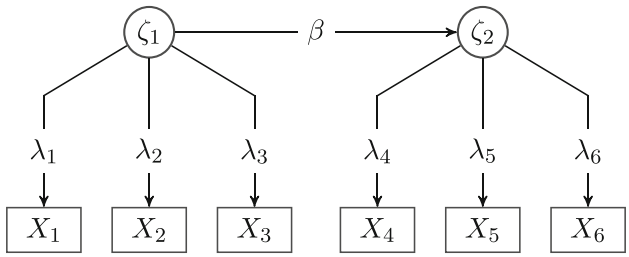


FIGURE 2.
The structural equation model used to compare convergence properties of SNLLS and GLS estimation, with two latent variables, ζ_1 and ζ_2. Variances are omitted in this representation. The population values are the same as in De Jonckere and Rosseel (2022): $\lambda_1 = \lambda_4 = 1$, $\lambda_2 = \lambda_5 = 0.8$, $\lambda_3 = \lambda_6 = 0.6$, $\beta = 0.25$, and all error variances are set to 1

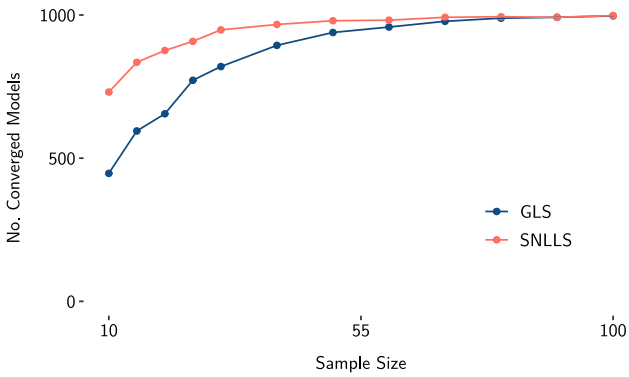


FIGURE 3.
Simulation results—number of converged replications out of 1000. GLS, generalized least squares; SNLLS, separable nonlinear least squares

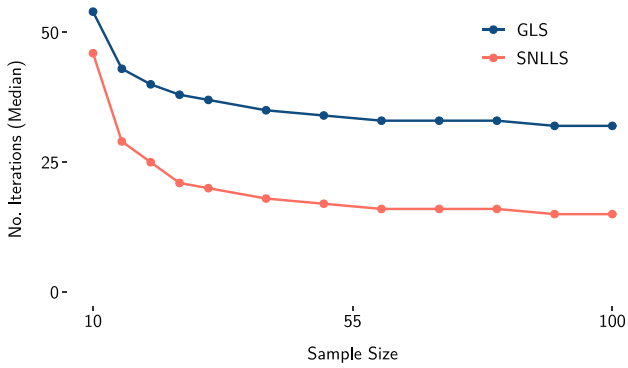


FIGURE 4.
Simulation results—median number of iterations by sample size. GLS, generalized least squares; SNLLS, separable nonlinear least squares

https://doi.org/10.1007/s11336-022-09891-5 Published online by Cambridge University Press

though their simulation is useful to illustrate the potential benefits of SNLLS, it seems unfit to us to make a case for a general reduction in computation time when using SNLLS in modern software. The gradient computation in the simulation was based on a finite difference approximation in both the LS and the SNLLS condition. In existing software (Rosseel, 2012; von Oertzen et al., 2015), analytic gradients are implemented for LS estimation, so the authors compare against a straw man that would not be used in practice if computational efficiency is important. In addition, centered finite differences takes $2q$ calls to the objective function per computation of the gradient, where q is the number of parameters. Since SNLLS results in a smaller parameter space, their method of differentiation favors the SNLLS procedure.

It remains to implement a competitive version of SNLLS optimization for SEM using the analytic gradients derived in this paper to be able to do a realistic simulation to investigate whether SNLLS outperforms the LS estimator in practice. However, there is a large body of research concerning the efficient implementation of SNLLS (see, for example, Kaufman, 1975; O'Leary and Rust, 2013); writing competitive software for SNLLS in SEMs would be a research topic on its own. Therefore, we only give simulation results concerning the improvement of convergence rates and the number of iterations in this paper. As noted previously, for the class of models without unknown directed parameters, the estimator of the mean and (co)variance parameters can be computed in a single step. As a result, those models should especially benefit from lower computation times.

3.3. An Outlook on Maximum Likelihood Estimation

If the assumption of multivariate normality is tenable, another method of obtaining parameter estimates is maximum likelihood estimation. Here, we briefly discuss to what extent our results may have an impact on maximum likelihood optimization of SEMs. In least squares estimation with a fixed weight matrix, we saw that the undirected parameters $\theta_{\mathbf{\Omega}}$ and the mean parameters θ_γ enter the objective linearly. For maximum likelihood estimation, we believe it is not possible to factor out the undirected parameters (for most models used in practice). This is because the likelihood of the normal distribution

$$\phi(x) = \left((2\pi)^{m_{\text{obs}}} \det \mathbf{\Sigma}\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^T \mathbf{\Sigma}^{-1}(x-\mu)\right) \tag{48}$$

depends on the inverse of the model-implied covariance matrix. For the simplistic example model depicted in Fig. 5, we derive the model-implied covariance matrix as

$$\mathbf{\Sigma} = \begin{pmatrix} \omega_l + \omega_1 & \omega_l \\ \omega_l & \omega_l + \omega_2 \end{pmatrix} \tag{49}$$

and the inverse can be computed as

$$\mathbf{\Sigma}^{-1} = (\det \mathbf{\Sigma})^{-1} \operatorname{adj} \mathbf{\Sigma} \tag{50}$$

where adj refers to the adjugate matrix, so in our example,

$$\det \mathbf{\Sigma} = (\omega_l + \omega_1)(\omega_l + \omega_2) - \omega_l^2 = \omega_1\omega_l + \omega_2\omega_l + \omega_1\omega_2 \tag{51}$$

and

$$\mathbf{\Sigma}^{-1} = (\omega_1\omega_l + \omega_2\omega_l + \omega_1\omega_2)^{-1} \begin{pmatrix} \omega_l + \omega_2 & -\omega_l \\ -\omega_l & \omega_l + \omega_1 \end{pmatrix} \tag{52}$$

https://doi.org/10.1007/s11336-022-09891-5 Published online by Cambridge University Press

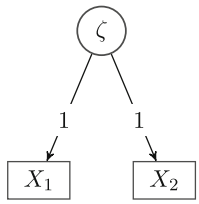


FIGURE 5.
Graph of a simplistic example model with one latent variable, measured by two indicators. The model contains no unknown directed effects and only two observed variables to allow for an easily traceable computation of the inverse of the model-implied covariance matrix. All variances are treated as unknown parameters

We see that $\theta_{\mathbf{\Omega}}$ enters the determinant and therefore the inverse of $\mathbf{\Sigma}$ in a nonlinear way. In general, the Leibniz Formula for the determinant gives

$$\det \mathbf{\Sigma} = \sum_{\pi \in \mathcal{S}_{m_{\text{obs}}}} \operatorname{sgn}(\pi) \prod_{i=1}^{m_{\text{obs}}} \mathbf{\Sigma}_{i,\pi(i)} \tag{53}$$

where $\mathcal{S}_{m_{\text{obs}}}$ denotes the symmetric group. Since this formula multiplies entries of $\mathbf{\Sigma}$, and we saw in Eq. 30 that the entries of $\mathbf{\Sigma}$ depend on the undirected parameters, it is very likely that those form a product and enter the objective in a nonlinear way. However, for the mean parameters, the picture may be different and we leave this for future work. If the model is saturated (e.g., has zero degrees of freedom), the least squares estimates are the same as the maximum likelihood estimates, since $\mathbf{S} = \mathbf{\Sigma}(\hat{\theta}_{\text{ML}}) = \mathbf{\Sigma}(\hat{\theta}_{\text{LS}})$. Also, Lee and Jennrich (1979) showed that maximum likelihood estimation can be obtained as a form of iteratively reweighted least squares if $\mathbf{V}$ is a function of the parameters:

$$\mathbf{V} = \frac{1}{2}\mathbf{D}^T \left(\mathbf{\Sigma}^{-1} \otimes \mathbf{\Sigma}^{-1}\right) \mathbf{D} \tag{54}$$

where $\mathbf{D}$ denotes the duplication matrix from Magnus and Neudecker (2019b). Another way of obtaining ML estimates with SNLLS would therefore be to minimize the SNLLS objective and use the obtained $\mathbf{\Sigma}$ to update the weight matrix $\mathbf{V}$ as given in Eq. 54. SNLLS could then be rerun with the updated weight matrix, and the weight matrix be updated again, until $\mathbf{\Sigma}$ converges to $\mathbf{\Sigma}(\hat{\theta}_{\text{ML}})$. However, we would like to note that this procedure is probably computationally quite inefficient.

3.4. Conclusion

We generalized separable nonlinear least squares estimation to all linear structural equation models that can be specified in the RAM notation, particularly those including a mean structure. We explained this result with the help of trek rules and the non-transitivity of the covariances of the error terms, providing deeper insight into the algebraic relations between the parameters of SEMs. We further derived analytic gradients and explained why they are of central importance to obtain a competitive implementation. Our simulation indicates that SNLLS leads to improvements in convergence rate and number of iterations. It remains for future research to investigate the computational costs empirically. We also showed why it is unlikely that undirected parameters enter the maximum likelihood objective linearly. Thus, another line of research could be concerned with the applicability of SNLLS to the mean parameters in maximum likelihood estimation and

https://doi.org/10.1007/s11336-022-09891-5 Published online by Cambridge University Press

the relationship of SNLLS to other decomposition methods for maximum likelihood estimation (Pritikin et al., 2017, 2018). Further research might also examine whether SNLLS is applicable to multilevel models. SNLLS promises better convergence rates for least squares parameter estimation in SEM and, with an efficient implementation, also reduced computation times. This result is important in its own right but may as well serve as a first step for generating starting values for subsequent ML estimation.

Funding Information Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest We have no conflicts of interest to disclose.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Appendix A: Equality Constraints

Kreiberg et al. (2021) showed how to incorporate equality constraints in CFA models. Because their proof follows a different approach, we show how to incorporate equality constraints in our expressions. Since the SNLLS objective only depends on $\theta_{\mathbf{\Lambda}}$, constraints in $\theta_{\mathbf{\Omega}}$ and θ_{γ} can be difficult to implement. However, simple equality constraints (e.g., $\theta_j = \theta_i$) are feasible under SNLLS. Since $\sigma = \mathbf{G}\,\theta_{\mathbf{\Omega},\gamma}$, we see that if two (or more) parameters in $\theta_{\mathbf{\Omega},\gamma}$ are equal, we can delete all but one occurrence from the parameter vector and add the relevant columns in $\mathbf{G}$ together, e.g.,

$$\begin{pmatrix} a & b & c \end{pmatrix} \begin{pmatrix} d \\ e \\ d \end{pmatrix} = ad + be + cd = (a + c)d + be = \begin{pmatrix} a + c & b \end{pmatrix} \begin{pmatrix} d \\ e \end{pmatrix} \tag{A1}$$

Or, put differently, if we allow the index tuples $\mathcal{C}$ and $\mathcal{A}$ to have sets of indices as entries, i.e., $\mathcal{C}_i = \{(k, l) \in \mathbb{N} \times \mathbb{N} \mid \theta_{\mathbf{\Omega}_i} = \omega_{kl} \wedge k \geq l\}$, we obtain

$$\mathbf{G}_{\sigma} = \left(\left[\sum_{(l,k) \in c} (\mathbf{I} - \mathbf{\Lambda})^{-1}_{il} (\mathbf{I} - \mathbf{\Lambda})^{-1}_{jk} + \delta_{k \neq l} \, (\mathbf{I} - \mathbf{\Lambda})^{-1}_{ik} (\mathbf{I} - \mathbf{\Lambda})^{-1}_{jl} \right]_{(i,j) \in \mathcal{D},\ c \in \mathcal{C}} \right) \tag{A2}$$

An expression for $\mathbf{G}_{\mu}$ can be obtained in a similar way.

https://doi.org/10.1007/s11336-022-09891-5 Published online by Cambridge University Press

Appendix B: Constants in $\mathbf{\Omega}$

To handle constants in $\mathbf{\Omega}$ different from zero, we introduce c as the vector of constant nonzero entries in $\mathbf{\Omega}$ and $\mathcal{E}$ as the lower triangular indices of c in $\mathbf{\Omega}$. Further, define

$$\mathbf{G}_c := \left(\left[(\mathbf{I}-\mathbf{\Lambda})^{-1}_{il} (\mathbf{I}-\mathbf{\Lambda})^{-1}_{jk} + \delta_{k\neq l} \, (\mathbf{I}-\mathbf{\Lambda})^{-1}_{ik} (\mathbf{I}-\mathbf{\Lambda})^{-1}_{jl} \right]_{(i,j)\in\mathcal{D},\ (l,k)\in\mathcal{E}} \right) \tag{B1}$$

with $\mathcal{D}$ defined as in Eq. 36, i.e., the indices of the original position of σ_k in $\mathbf{\Sigma}$. This allows us to modify Eq. 38 to

$$\sigma = \mathbf{G}(\theta_{\mathbf{\Lambda}}) \, \theta_{\mathbf{\Omega}} \; + \; \mathbf{G}_c \, c \tag{B2}$$

and reformulate the least squares objective as

$$\begin{aligned} F_{\mathrm{LS}} &= (s-\sigma)^T \mathbf{V} (s-\sigma) && \text{(B3)} \\ &= (s - (\mathbf{G}(\theta_{\mathbf{\Lambda}})\theta_{\mathbf{\Omega}} + \mathbf{G}_c c))^T \, \mathbf{V} \, (s - (\mathbf{G}(\theta_{\mathbf{\Lambda}})\theta_{\mathbf{\Omega}} + \mathbf{G}_c c)) && \text{(B4)} \\ &= ((s-\mathbf{G}_c c) - \mathbf{G}(\theta_{\mathbf{\Lambda}})\theta_{\mathbf{\Omega}})^T \, \mathbf{V} \, ((s-\mathbf{G}_c c) - \mathbf{G}(\theta_{\mathbf{\Lambda}})\theta_{\mathbf{\Omega}}) && \text{(B5)} \\ &= \left\| s' - \mathbf{G}(\theta_{\mathbf{\Lambda}})\theta_{\mathbf{\Omega}} \right\|^2_{\mathbf{V}} && \text{(B6)} \end{aligned}$$

with $s' = (s - \mathbf{G}_c c)$. For a fixed value of $\theta_{\mathbf{\Lambda}}$, we can now again solve for $\theta_{\mathbf{\Omega}}$.

Appendix C: Gradient of the SNLLS Objective

Let $a^T := s^T \mathbf{V}\mathbf{G} \left(\mathbf{G}^T\mathbf{V}\mathbf{G}\right)^{-1}$. We derive the differential as

$$\begin{aligned} \mathrm{d}\, F_{\mathrm{SNLLS}} &= \mathrm{d}\left(s^T\mathbf{V}s - s^T\mathbf{V}\mathbf{G}\left(\mathbf{G}^T\mathbf{V}\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{V}s \right) \\ &= -\,\mathrm{d}\left(s^T\mathbf{V}\mathbf{G}\left(\mathbf{G}^T\mathbf{V}\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{V}s \right) \\ &= -s^T\mathbf{V}\,\mathrm{d}\mathbf{G}\left(\mathbf{G}^T\mathbf{V}\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{V}s \\ &\quad - s^T\mathbf{V}\mathbf{G}\,\mathrm{d}\left(\mathbf{G}^T\mathbf{V}\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{V}s \\ &\quad - s^T\mathbf{V}\mathbf{G}\left(\mathbf{G}^T\mathbf{V}\mathbf{G}\right)^{-1}\mathrm{d}\,\mathbf{G}^T\mathbf{V}s \\ &= -2\, s^T\mathbf{V}\,\mathrm{d}\mathbf{G}\left(\mathbf{G}^T\mathbf{V}\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{V}s \\ &\quad + s^T\mathbf{V}\mathbf{G}\left(\mathbf{G}^T\mathbf{V}\mathbf{G}\right)^{-1}\mathrm{d}\left(\mathbf{G}^T\mathbf{V}\mathbf{G}\right)\left(\mathbf{G}^T\mathbf{V}\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{V}s \\ &= -2\, s^T\mathbf{V}\,\mathrm{d}\mathbf{G}\,a \\ &\quad + s^T\mathbf{V}\mathbf{G}\left(\mathbf{G}^T\mathbf{V}\mathbf{G}\right)^{-1}\left[\left(\mathrm{d}\,\mathbf{G}^T\mathbf{V}\mathbf{G}\right) + \left(\mathbf{G}^T\mathbf{V}\,\mathrm{d}\mathbf{G}\right)\right]\left(\mathbf{G}^T\mathbf{V}\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{V}s \\ &= -2\, s^T\mathbf{V}\,\mathrm{d}\mathbf{G}\,a \end{aligned}$$

https://doi.org/10.1007/s11336-022-09891-5 Published online by Cambridge University Press

$$
\begin{aligned}
&+ a^T \left[\left(\mathsf{d}\,\mathbf{G}^T \mathbf{V} \mathbf{G} \right) + \left(\mathbf{G}^T \mathbf{V}\, \mathsf{d}\,\mathbf{G} \right) \right] a \\
&= -2\, s^T \mathbf{V}\, \mathsf{d}\,\mathbf{G} a \\
&\quad + 2\, a^T \mathbf{G}^T \mathbf{V}\, \mathsf{d}\,\mathbf{G} a \\
&= -2 \left(a^T \otimes s^T \mathbf{V} \right) \mathsf{d}\,\mathrm{vec}\,\mathbf{G} \\
&\quad + 2 \left(a^T \otimes a^T \mathbf{G}^T \mathbf{V} \right) \mathsf{d}\,\mathrm{vec}\,\mathbf{G} \\
&= -2\ \mathrm{vec} \left(\mathbf{V} s a^T \right)^T \mathsf{d}\,\mathrm{vec}\,\mathbf{G} \\
&\quad + 2\ \mathrm{vec} \left(\mathbf{V}\mathbf{G} a a^T \right)^T \mathsf{d}\,\mathrm{vec}\,\mathbf{G} \\
&= 2\ \mathrm{vec} \left(\mathbf{V}\mathbf{G} a a^T - \mathbf{V} s a^T \right)^T \mathsf{d}\,\mathrm{vec}\,\mathbf{G} \\
&= 2\ \mathrm{vec} \left((\mathbf{V}\mathbf{G} a - \mathbf{V} s)\, a^T \right)^T \mathsf{d}\,\mathrm{vec}\,\mathbf{G} \\
&= \underbrace{2\ \mathrm{vec} \left((\mathbf{V}\mathbf{G} a - \mathbf{V} s)\, a^T \right)^T \mathsf{D}\,\mathbf{G}}_{=\mathsf{D}\, F_{\mathrm{SNLLS}}} \mathsf{d}\,\theta_{\mathbf{\Lambda}}
\end{aligned}
\tag{C1}
$$

Appendix D: Alternative Proof

This is the analogous formulation to the one given in Kreiberg et al. (2021) for CFA models. We see that the resulting expressions contain very large Kronecker products; for reasons of computational efficiency, we therefore favor the expressions given in the main text. Let $\mathbf{D}^+$ denote the Moore–Penrose inverse of the duplication matrix $\mathbf{D}_{m_{\mathrm{obs}}}$ from Magnus and Neudecker (2019b) such that

$$
\sigma = \mathbf{D}^+ \mathrm{vec}(\mathbf{\Sigma}) \tag{D1}
$$

and $\mathbf{L}$ be a matrix such that

$$
\mathrm{vec}(\mathbf{\Omega}) = \mathbf{L}\, \theta_{\mathbf{\Omega}} \tag{D2}
$$

We can obtain $\mathbf{L} \in \mathbb{R}^{m^2 \times \dim(\theta_{\mathbf{\Omega}})}$ as

$$
\mathbf{L}_{ij} = \begin{cases} 1, & \text{if } i = (k-1)m + l \vee i = (l-1)m + k \text{ with } (k,l) = \mathcal{C}_j \\ 0, & \text{otherwise} \end{cases} \tag{D3}
$$

and derive $\mathbf{G}(\theta_{\mathbf{\Lambda}})$ as

$$
\begin{aligned}
\sigma(\theta) &= \mathbf{D}^+ \mathrm{vec}(\mathbf{\Sigma}) && \text{(D4)} \\
&= \mathbf{D}^+ \mathrm{vec}(\mathbf{F}(\mathbf{I} - \mathbf{\Lambda})^{-1} \mathbf{\Omega} (\mathbf{I} - \mathbf{\Lambda})^{-T} \mathbf{F}^T) && \text{(D5)} \\
&= \mathbf{D}^+ \left(\mathbf{F}\,(\mathbf{I} - \mathbf{\Lambda})^{-1} \otimes \mathbf{F}\,(\mathbf{I} - \mathbf{\Lambda})^{-1} \right) \mathrm{vec}(\mathbf{\Omega}) && \text{(D6)} \\
&= \underbrace{\mathbf{D}^+ \left(\mathbf{F}\,(\mathbf{I} - \mathbf{\Lambda})^{-1} \otimes \mathbf{F}\,(\mathbf{I} - \mathbf{\Lambda})^{-1} \right) \mathbf{L}}_{=G(\theta_{\mathbf{\Lambda}})}\ \theta_{\mathbf{\Omega}} && \text{(D7)}
\end{aligned}
$$

https://doi.org/10.1007/s11336-022-09891-5 Published online by Cambridge University Press

We further define $\mathbf{P} := \left(\mathbf{L}^T \otimes \mathbf{D}^+ (\mathbf{F} \otimes \mathbf{F})\right)$ and $\mathbf{Q} := (\mathbf{I}_m \otimes \mathbf{K}_m \otimes \mathbf{I}_m)$, where $\mathbf{K}_m$ is the commutation matrix from Magnus and Neudecker (2019b), and derive $\mathsf{D}\,\mathbf{G}$ as

$$
\begin{aligned}
\mathsf{d}\,\mathrm{vec}\,\mathbf{G} &= \mathsf{d}\,\mathrm{vec}\left[\mathbf{D}^+ \left(\mathbf{F}\,(\mathbf{I}-\mathbf{\Lambda})^{-1} \otimes \mathbf{F}\,(\mathbf{I}-\mathbf{\Lambda})^{-1}\right)\mathbf{L}\right] \\
&= \mathsf{d}\,\mathrm{vec}\left[\mathbf{D}^+ (\mathbf{F}\otimes\mathbf{F})\left((\mathbf{I}-\mathbf{\Lambda})^{-1} \otimes (\mathbf{I}-\mathbf{\Lambda})^{-1}\right)\mathbf{L}\right] \\
&= \left(\mathbf{L}^T \otimes \mathbf{D}^+ (\mathbf{F}\otimes\mathbf{F})\right)\mathsf{d}\,\mathrm{vec}\left((\mathbf{I}-\mathbf{\Lambda})^{-1} \otimes (\mathbf{I}-\mathbf{\Lambda})^{-1}\right) \\
&= \mathbf{P}\,\mathrm{vec}\left(\mathsf{d}\,(\mathbf{I}-\mathbf{\Lambda})^{-1} \otimes (\mathbf{I}-\mathbf{\Lambda})^{-1} + (\mathbf{I}-\mathbf{\Lambda})^{-1} \otimes \mathsf{d}\,(\mathbf{I}-\mathbf{\Lambda})^{-1}\right) \\
&= \mathbf{P}[\\
&\quad \mathrm{vec}\left((\mathbf{I}-\mathbf{\Lambda})^{-1}\,\mathsf{d}\,\mathbf{\Lambda}\,(\mathbf{I}-\mathbf{\Lambda})^{-1} \otimes (\mathbf{I}-\mathbf{\Lambda})^{-1}\right) \\
&\quad + \mathrm{vec}\left((\mathbf{I}-\mathbf{\Lambda})^{-1} \otimes (\mathbf{I}-\mathbf{\Lambda})^{-1}\,\mathsf{d}\,\mathbf{\Lambda}\,(\mathbf{I}-\mathbf{\Lambda})^{-1}\right) \\
&\quad] \\
&= \mathbf{PQ}[\\
&\quad \mathrm{vec}\left((\mathbf{I}-\mathbf{\Lambda})^{-1}\,\mathsf{d}\,\mathbf{\Lambda}\,(\mathbf{I}-\mathbf{\Lambda})^{-1}\right) \otimes \mathrm{vec}\,(\mathbf{I}-\mathbf{\Lambda})^{-1} \\
&\quad + \mathrm{vec}\,(\mathbf{I}-\mathbf{\Lambda})^{-1} \otimes \mathrm{vec}\left((\mathbf{I}-\mathbf{\Lambda})^{-1}\,\mathsf{d}\,\mathbf{\Lambda}\,(\mathbf{I}-\mathbf{\Lambda})^{-1}\right) \\
&\quad] \\
&= \mathbf{PQ}[\\
&\quad \left(\mathbf{I}_{m^2} \otimes \mathrm{vec}\,(\mathbf{I}-\mathbf{\Lambda})^{-1}\right)\mathrm{vec}\left((\mathbf{I}-\mathbf{\Lambda})^{-1}\,\mathsf{d}\,\mathbf{\Lambda}\,(\mathbf{I}-\mathbf{\Lambda})^{-1}\right) \\
&\quad + \left(\mathrm{vec}\,(\mathbf{I}-\mathbf{\Lambda})^{-1} \otimes \mathbf{I}_{m^2}\right)\mathrm{vec}\left((\mathbf{I}-\mathbf{\Lambda})^{-1}\,\mathsf{d}\,\mathbf{\Lambda}\,(\mathbf{I}-\mathbf{\Lambda})^{-1}\right) \\
&\quad] \\
&= \underbrace{\mathbf{PQ}\left[\left(\mathbf{I}_{m^2} \otimes \mathrm{vec}\,(\mathbf{I}-\mathbf{\Lambda})^{-1}\right) + \left(\mathrm{vec}\,(\mathbf{I}-\mathbf{\Lambda})^{-1} \otimes \mathbf{I}_{m^2}\right)\right]\left((\mathbf{I}-\mathbf{\Lambda})^{-T} \otimes (\mathbf{I}-\mathbf{\Lambda})^{-1}\right)\mathsf{D}\mathbf{\Lambda}}_{=\mathsf{D}\,\mathbf{G}}\,\mathsf{d}\,\theta_\Lambda
\end{aligned}
\tag{D8}
$$

References

Boker, S. M., McArdle, J. J., & Neale, M. (2002). An algorithm for the hierarchical organization of path diagrams and calculation of components of expected covariance. *Structural Equation Modeling: A Multidisciplinary Journal, 9*(2), 174–194. https://doi.org/10.1207/S15328007SEM0902_2

Bollen, K. A. (1989). *Structural equations with latent variables*. John Wiley & Sons, Ltd. https://doi.org/10.1002/9781118619179

Browne, M. W. (1982). Covariance structures. In D. M. Hawkins (Ed.), *Topics in applied multivariate analysis* (pp. 72–141). Cambridge University Press. https://doi.org/10.1017/CBO9780511897375.003

Browne, M. W. (1984). Asymptotically distribution-free methods for the analysis of covariance structures. *British Journal of Mathematical and Statistical Psychology, 37*(1), 62–83. https://doi.org/10.1111/j.2044-8317.1984.tb00789.x

De Jonckere, J., & Rosseel, Y. (2022). Using bounded estimation to avoid nonconvergence in small sample structural equation modeling. *Structural Equation Modeling: A Multidisciplinary Journal, 29*(3), 412–427. https://doi.org/10.1080/10705511.2021.1982716

Drton, M. (2018). Algebraic problems in structural equation modeling. *Advanced Studies in Pure Mathematics, 77*, 35–86. https://doi.org/10.2969/aspm/07710035

Ernst, M. S. (2022). *Separable nonlinear least squares estimation of structural equation models*. Master's thesis. Humboldt-Universität zu Berlin.

Golub, G. H., & Pereyra, V. (1973). The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on Numerical Analysis, 10*(2), 413–432. https://doi.org/10.1137/0710036

Golub, G. H., & Pereyra, V. (2003). Separable nonlinear least squares: The variable projection method and its applications. *Inverse Problems, 19*(2), R1–R26. https://doi.org/10.1088/0266-5611/19/2/201

Hägglund, G. (1982). Factor analysis by instrumental variables methods. *Psychometrika, 47*(2), 209–222. https://doi.org/10.1007/BF02296276

Kaufman, L. (1975). A variable projection method for solving separable nonlinear least squares problems. *BIT Numerical Mathematics, 15*(1), 49–57. https://doi.org/10.1007/BF01932995

Kepner, J., & Gilbert, J. (Eds.). (2011). *Graph algorithms in the language of linear algebra*. Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9780898719918

https://doi.org/10.1007/s11336-022-09891-5 Published online by Cambridge University Press

Kreiberg, D., Marcoulides, K., & Olsson, U. H. (2021). A faster procedure for estimating cfa models applying minimum distance estimators with a fixed weight matrix. *Structural Equation Modeling: A Multidisciplinary Journal, 28*(5), 725–739. https://doi.org/10.1080/10705511.2020.1835484

Kreiberg, D., Söderström, T., & Yang-Wallentin, F. (2016). Errors-in-variables system identification using structural equation modeling. *Automatica, 66*, 218–230. https://doi.org/10.1016/j.automatica.2015.12.007

Lee, S. Y., & Jennrich, R. I. (1979). A study of algorithms for covariance structure analysis with specific comparisons using factor analysis. *Psychometrika, 44*(1), 99–113. https://doi.org/10.1007/BF02293789

Magnus, J. R., & Neudecker, H. (2019a). Differentials and differentiability. In *Matrix differential calculus with applications in statistics and econometrics* (pp. 87–110). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781119541219.ch5

Magnus, J. R., & Neudecker, H. (2019b). Miscellaneous matrix results. In *Matrix differential calculus with applications in statistics and econometrics* (pp. 47–70). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781119541219.ch3

McArdle, J. J., & McDonald, R. P. (1984). Some algebraic properties of the reticular action model for moment structures. *British Journal of Mathematical and Statistical Psychology, 37*(2), 234–251. https://doi.org/10.1111/j.2044-8317.1984.tb00802.x

Mulaik, S. A. (2009). *Linear causal modeling with structural equations*. Chapman and Hall/CRC. https://doi.org/10.1201/9781439800393

O'Leary, D. P., & Rust, B. W. (2013). Variable projection for nonlinear least squares problems. *Computational Optimization and Applications, 54*(3), 579–593. https://doi.org/10.1007/s10589-012-9492-9

Pritikin, J. N., Brick, T. R., & Neale, M. C. (2018). Multivariate normal maximum likelihood with both ordinal and continuous variables, and data missing at random. *Behavior Research Methods, 50*(2), 490–500. https://doi.org/10.3758/s13428-017-1011-6

Pritikin, J. N., Hunter, M. D., von Oertzen, T., Brick, T. R., & Boker, S. M. (2017). Many-level multilevel structural equation modeling: An efficient evaluation strategy. *Structural Equation Modeling: A Multidisciplinary Journal, 24*(5), 684–698. https://doi.org/10.1080/10705511.2017.1293542

R Core Team. (2021). *R: A language and environment for statistical computing*. Vienna: R Foundation for Statistical Computing. https://www.R-project.org/

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software, 48*(2), 1–36. https://doi.org/10.18637/jss.v048.i02

Ruhe, A., & Wedin, P. Å. (1980). Algorithms for separable nonlinear least squares problems. *SIAM Review, 22*(3), 318–337. https://doi.org/10.1137/1022057

von Oertzen, T., Brandmaier, A. M., & Tsang, S. (2015). Structural equation modeling with Ωnyx. *Structural Equation Modeling: A Multidisciplinary Journal, 22*(1), 148–161. https://doi.org/10.1080/10705511.2014.935842

Wickham, H. (2016). *Ggplot2: Elegant graphics for data analysis*. Springer. https://www.ggplot2.tidyverse.org

Wickham, H., François, R., Henry, L., & Müller, K. (2021). *Dplyr: A grammar of data manipulation*. https://www.CRAN.R-project.org/package=dplyr

Manuscript Received: 5 FEB 2022
Final Version Received: 21 AUG 2022
Accepted: 19 OCT 2022
Published Online Date: 25 DEC 2022

https://doi.org/10.1007/s11336-022-09891-5 Published online by Cambridge University Press