Behav Res (2014) 46:385–395
DOI 10.3758/s13428-013-0384-4

Efficient Hessian computation using sparse matrix derivatives in RAM notation

Timo von Oertzen · Timothy R. Brick

Published online: 3 October 2013

Abstract This article proposes a new, more efficient method to compute the minus two log likelihood, its gradient, and the Hessian for structural equation models (SEMs) in reticular action model (RAM) notation. The method exploits the beneficial aspect of RAM notation that the matrix derivatives used in RAM are sparse. For an SEM with K variables, P parameters, and P' entries in the symmetrical or asymmetrical matrix of the RAM notation filled with parameters, the asymptotical run time of the algorithm is $O(P'K^2+P^2K^2+K^3)$. The naive implementation and numerical implementations are both $O(P^2K^3)$, so that for typical applications of SEM, the proposed algorithm is asymptotically K times faster than the best previously known algorithm. A simulation comparison with a numerical algorithm shows that the asymptotical efficiency is transferred to an applied computational advantage that is crucial for the application of maximum likelihood estimation, even in small, but especially in moderate or large, SEMs.

Keywords RAM notation · Minus two log likelihood · Hessian · Sparse density algorithm

T. von Oertzen (✉)
Department of Psychology, University of Virginia, 1023 Millmont Street, Charlottesville, VA 22903, USA
e-mail: timo@virginia.edu

T. R. Brick
Center for Lifespan Psychology, Max Planck Institute for Human Development, Berlin, Germany

Introduction

In the last decade, models in psychology have become increasingly larger and more complex, in terms of both the number of parameters and the number of observations for each data row (e.g., Voelkle, Oud, von Oertzen, & Lindenberger, 2012). At the same time, the body of researchers who use these complex models is also broadening.

Structural equation models (SEMs; cf. Baltes, Reese, & Nesselroade, 1988) have been found to be a useful compromise between model readability and model richness for a broad range of models. At the beginning of SEMs, these models were usually expressed in mathematically dense matrix representations. These representations are convenient for the design of efficient optimization routines for these models, as, for example, those used in LISREL (Jöreskog, and Sörbom, 1993), MPlus (Muthén & Muthén, 2004), or others. As the SEM framework has come to be used more broadly, path diagrams (McArdle & Boker, 1990; Stelzl, 1986; Wright, 1934) have emerged as a means of graphically representing models. For example, Fig. 1 shows a path diagram representing a latent growth curve. Path diagrams are intuitive to understand and convenient for both the researcher using the model and the reader of publications. However, path diagrams are less useful as a computer representation. When fitting a model to a data set with a suitable computer program, the path diagram has to be converted to a matrix representation, either by the computer or manually by the researcher (cf. Grimm & McArdle, 2005).

Reticular action model (RAM) notation (McArdle, 2005; McArdle & Boker, 1990; McArdle & McDonald, 1984) for SEMs overcomes the conflict between simple accessibility

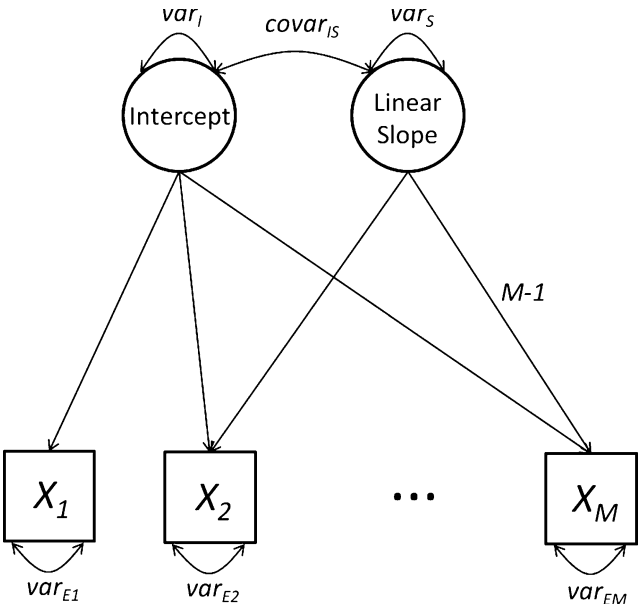

Fig. 1 Latent growth curve model with M measurement occasions (for mean centered data)

and computer efficiency. In RAM, two matrices (A for asymmetric, one-headed arrows and S for symmetric, two-headed arrows) directly represent the paths from a path diagram. The covariance matrix of all variables is given by the simple equation

$$(I-A)^{-1}S(I-A)^{-T}. \tag{1}$$

Both latent and manifest variables are described by this matrix. A filter matrix F that selects the manifest variables from the set of all variables is applied to the equation to obtain the total model-predicted covariance matrix

$$F(I-A)^{-1}S(I-A)^{-T}F^T. \tag{2}$$

Modern statistical programs like OpenMx (Boker et al., 2011) or Onyx (von Oertzen, Brandmaier, & Tsang, 2012) use RAM to represent the data covariance matrix and to compute the likelihood of a given data set. Since the matrix entries directly represent path weights in a path diagram, OpenMx allows both easy access to the SEM for researchers less familiar with matrix notation and a direct implementation in the software.

One main advantage of RAM is that since path diagrams directly represent all polynomial dependencies between variables, all entries in the A and S matrices of the RAM notation are either constants or single parameters. This is most critical if one aims to compute the gradient and the Hessian of likelihood analytically. In this case, we find that the first derivatives of A and S,

$$\frac{\partial A}{\partial \theta} \quad and \quad \frac{\partial S}{\partial \theta}, \tag{3}$$

are both sparse, since only the entries of A and S that are the parameter θ are one, while all other entries (all constants and any other parameters in A and S) are zero. In addition, the second derivatives

$$\frac{\partial^2 A}{\partial^2 \theta} \quad and \quad \frac{\partial^2 S}{\partial^2 \theta} \tag{4}$$

are both zero.

However, current implementations of statistical optimizers do not take advantage of this property. Usually, the gradient and/or Hessian are not computed explicitly but numerically approximated by the optimizer—for example, the NPSOL routine (Gill et al., 2001) in OpenMx. Other packages use a similar strategy. The sem package (Fox, 2006) does not use the Hessian in the optimization but computes it when requested by the user or required for some summary calculations. It computes the Hessian for all possible paths simultaneously and then selects those elements that are free parameters from the complete matrices. The lavaan (Rosseel, 2012) package also relies on a Hessian-free optimizer for maximum likelihood. When a Hessian is required for summary statistics, it uses a more precise numerical approximation. Both methods are very effective, but neither method takes into account the specific sparse structure of the Hessian in RAM notation, which can further improve the computation speed and precision.

In addition to the fact that numerical approximation is, by design, imprecise, for P parameters it needs P^2 calls to the likelihood computation. Since the derivatives of A and S are not used for the likelihood computation, the sparseness of these matrices is ignored. For K variables, K^3 multiplications need to be done to compute the matrix products in the likelihood computation. In total, numerical approximation of the Hessian requires $O(P^2K^3)$ operations.[1]

The RAM representation is uniquely suited to this optimization. The smaller, denser matrices of the LISREL specification require the same asymptotic runtime to compute, since the computation time still scales up to larger numbers of parameters and variables in roughly the same manner. At the same time, the increased density of these smaller matrices and

[1] Not all optimizers necessarily compute a full Hessian at each step of optimization. Some trade precision in the Hessian against computation time, even though they still do not benefit from the sparseness in RAM notation.

the increased complexity of the covariance equation make the LISREL representation less suitable for the sparse derivative method. While it might be possible to derive such a method for the LISREL specification, the authors are aware of no such algorithm at the time of printing.

In the present article, we suggest a method called the *sparse derivative algorithm* to compute the likelihood, the gradient, and the Hessian analytically and efficiently. In this method, the derivatives of the likelihood are expressed in terms of the derivatives of A and S. This permits efficient use of their sparseness. Furthermore, the sum of all nonzero entries in all derivatives of A and S is bounded by K^2. This provides good upper bounds to compute the running time.

The sparse derivative algorithm exploits these bounds to precompute a number of matrices that depend on either $\frac{\partial A}{\partial \theta}$ or $\frac{\partial S}{\partial \theta}$ for all parameters θ in $P'K^2$ steps, where P' is the number of nonconstant entries in A and S. Typically, this value is of the same order of magnitude as K. Then the likelihood, gradient, and Hessian are computed using these prepared matrices in P^2K^2 steps.

The derivations required to express the likelihood, its gradient, and the Hessian in terms of the derivatives of A and S are mathematically involved, but they are worth the effort. While the speed gain of the Hessian computation is less important for small models, we show that for models with a high number of variables and parameters, the proposed method outperforms classical methods by far. This allows a noticeable speedup in fitting data to complex SEMs, which widens the horizon of models psychologists can reasonably use in their research.

Mathematical methods

In RAM notation, SEMs are described by a symmetrical matrix $S \in \mathbb{R}^{K\times K}$, an asymmetrical matrix $A \in \mathbb{R}^{K\times K}$, a filter matrix $F \in \mathbb{R}^{M\times K}$, and a mean vector $m \in \mathbb{R}^K$, where M is the number of manifest variables and K the total number of variables including the latent variables from the model. The data in SEMs are normally distributed with covariance matrix and mean

$$\Sigma = F(I-A)^{-1}S(I-A)^{-T}F^T \qquad \mu = F(I-A)^{-1}m. \tag{5}$$

In general form, the minus two log likelihood divided by the number of samples N for a data covariance matrix D and mean d is

$$\mathcal{L} = \ln\Big((2\pi)^M det(\Sigma)\Big) + Tr\big(\Sigma^{-1}D\big) + (d-\mu)^T\Sigma^{-1}(d-\mu). \tag{6}$$

In this section, we describe the mathematical processes behind the algorithm (see Algorithm 1 below). In essence, we define some helper matrices that allow us to reexpress the second derivative of Equation 6 in different terms. In this new formulation, the computationally expensive part of the second derivative consists of products $[XYZ]$ in which Y is sparse—that is, has only a few nonzero elements. That permits efficient computation of the terms $[XYZ]$. We will first introduce the helper matrices and compute the derivatives of these matrices and of Σ and μ, using the helper matrices. We then reexpress the second derivative of Equation 6 in terms of these matrix derivatives. Other than matrix derivatives, no mathematical theorems are used. An expansion of all derivatives can be found in the Appendix. In the next section, we continue by describing the computational effects of this algorithm.

Algorithm 1: Outline of the Sparse Derivatives Algorithm

```
begin
    Precompute B, C, E, and b
    for each parameter p :
        Compute 14 [XYZ]_p terms
    end
    for each pair of parameters (i, j) :
        Compute Equation 22 and Line 23 from the [XYZ]_i and [XYZ]_j terms
        Add mean contributions (Lines 24 – 26)
        Populate Hessian elements (i, j) and (j, i)
    end
end
```

In the following, we will assume that all entries in A, S, and m are either constant or a single parameter, such that the entries of the first derivatives of these are either zero or one and the second derivative is zero. We abbreviate

$$B = (I - A)^{-1} \tag{7}$$

$$C = \left(I - \Sigma^{-1} D\right) \tag{8}$$

$$E = BSB^T \tag{9}$$

$$b = (d - \mu). \tag{10}$$

Observe that the derivatives of B, C, and E are given by

$$\frac{\partial B}{\partial \theta} = B \frac{\partial A}{\partial \theta} B \tag{11}$$

$$\frac{\partial C}{\partial \theta} = \Sigma^{-1} \frac{\partial \Sigma}{\partial \theta} \Sigma^{-1} D \tag{12}$$

$$\frac{\partial E}{\partial \theta} = \left[B \frac{\partial A}{\partial \theta} E \right]^{sym} + B \frac{\partial S}{\partial \theta} B^T, \tag{13}$$

where $A^{sym} = A + A^T$. In terms of these matrices, Σ can be expressed as

$$\Sigma = FEF^T \tag{14}$$

and

$$\frac{\partial \Sigma}{\partial \theta} = \left[FB \frac{\partial A}{\partial \theta} EF^T \right]^{sym} + FB \frac{\partial S}{\partial \theta} B^T F^T \tag{15}$$

and

$$\frac{\partial^2 \Sigma}{\partial \theta_1 \partial \theta_2} = \left[FB \frac{\partial A}{\partial \theta_1} B \frac{\partial A}{\partial \theta_2} EF^T + FB \frac{\partial A}{\partial \theta_2} B \frac{\partial A}{\partial \theta_1} EF^T \right. \tag{16}$$

$$+ FB \frac{\partial A}{\partial \theta_1} B \frac{\partial S}{\partial \theta_2} B^T F^T + FB \frac{\partial A}{\partial \theta_1} E \left(\frac{\partial A}{\partial \theta_2} \right)^T B^T F^T \tag{17}$$

$$\left. + FB \frac{\partial A}{\partial \theta_2} B \frac{\partial S}{\partial \theta_1} B^T F^T \right]^{sym}. \tag{18}$$

The means can be expressed in the same way as

$$\mu = FBm \tag{19}$$

with derivatives

$$\frac{\partial \mu}{\partial \theta} = FB \frac{\partial A}{\partial \theta} Bm + FB \frac{\partial m}{\partial \theta} \tag{20}$$

and

$$\frac{\partial^2 \mu}{\partial \theta_1 \partial \theta_2} = FB \frac{\partial A}{\partial \theta_2} B \frac{\partial A}{\partial \theta_1} Bm + FB \frac{\partial A}{\partial \theta_1} B \frac{\partial A}{\partial \theta_2} Bm + FB \frac{\partial A}{\partial \theta_2} B \frac{\partial m}{\partial \theta_1} + FB \frac{\partial A}{\partial \theta_1} B \frac{\partial m}{\partial \theta_2} \tag{21}$$

The derivatives of A, S, and μ are sparse, since the entries are zero everywhere, with the exception of those entries where the parameter occurs. Even better, the nonzero entries of the first derivatives of A, S, and μ for all parameters together are bounded by the size of the matrices and the vector, respectively.

The first derivative of $\mathcal{L}$ is given as (cf., e.g., Pinheiro & Bates, 2000)

$$\frac{\partial \mathcal{L}}{\partial \theta_1} = Tr\left(\Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_1} C \right) - \left(b^T \Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_1} + 2 \left(\frac{\partial \mu}{\partial \theta_1} \right)^T \right) \Sigma^{-1} b, \tag{22}$$

and the second derivative as

$$\frac{\partial^2 \mathcal{L}}{\partial \theta_1 \partial \theta_2} = Tr\left(\Sigma^{-1} \frac{\partial^2 \Sigma}{\partial \theta_1 \partial \theta_2} C - \Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_1} \Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_2} C + \Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_2} \Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_1} \Sigma^{-1} D \right) \tag{23}$$

$$+ b^T \left(2\Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_1} \Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_2} \Sigma^{-1} - \Sigma^{-1} \frac{\partial^2 \Sigma}{\partial \theta_1 \partial \theta_2} \Sigma^{-1} \right) b \tag{24}$$

$$+ 2 \left(\frac{\partial \mu}{\partial \theta_1} \right)^T \Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_2} \Sigma^{-1} b + 2 \left(\frac{\partial \mu}{\partial \theta_2} \right)^T \Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_1} \Sigma^{-1} b \tag{25}$$

$$+ 2 \left(\frac{\partial \mu}{\partial \theta_1} \right)^T \Sigma^{-1} \frac{\partial \mu}{\partial \theta_2} - 2 \left(\frac{\partial^2 \mu}{\partial \theta_1 \partial \theta_2} \right)^T \Sigma^{-1} b. \tag{26}$$

Considering computational efficiency, all terms outside the traces in the first and second derivatives are a number of matrix-vector multiplications, so they can be done efficiently in P^2K^2 steps at most. We concentrate on computing the trace in line 23. A close inspection reveals that the argument of the trace is a sum of multiple matrices of the form

$$[XYZ] \qquad or \qquad [X_1 Y_1 Z_1] \cdot [X_2 Y_2 Z_2], \tag{27}$$

where the Xs and Zs are potentially full matrices (sometimes the identity) and the Ys are the first derivative of either A or S and, thus, sparse matrices, where the number of nonzero entries for all parameters sum up to P'. As a result, the $[XYZ]$ for all parameters can be computed by looping through the parameters and (1) enumerating every position in Y that a given parameter appears in and (2) performing an outer vector multiplication on the corresponding row of X and column of

Z. Each outer vector multiplication requires K^2 steps. This is done for at most P' position of Y. As a result for all parameters this process can be done in $P'K^2$ steps for each term $[XYZ]$.

Once we have prepared all XYZ matrices, we compute the trace either directly for the $[XYZ]$ pairs or using

$$Tr([X_1Y_1Z_1]\cdot[X_2Y_2Z_2]) = \sum_{i,j=1}^{K} (X_1Y_1Z_1)_{ij}(X_2Y_2Z_2)_{ij} \quad (28)$$

in K^2 steps for each parameter combination. As a result, these multiplications can be done efficiently in P^2K^2 steps in total.

What remains is to identify the $[XYZ]$ triplets in the trace. Factorizing the trace term in Line 23 using the derivatives of Σ yields four terms to compute:

$$\overbrace{\Sigma^{-1}\frac{\partial^2\Sigma}{\partial\theta_1\partial\theta_2}C}^{4} - \overbrace{\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_1}}^{1}\overbrace{\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_2}C}^{3} + \overbrace{\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_2}}^{1}\overbrace{\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_1}\Sigma^{-1}D}^{2}. \quad (29)$$

These, in turn, factorize in terms of the derivatives of A and S to

$$\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta}Z = \overbrace{\left[\Sigma^{-1}FB\frac{\partial A}{\partial\theta}EF^TZ\right]}^{1,2,3} + \overbrace{\left[\Sigma^{-1}FE\left(\frac{\partial A}{\partial\theta}\right)^TB^TF^TZ\right]}^{4,5,6} \quad (30)$$

$$+ \overbrace{\left[\Sigma^{-1}FB\frac{\partial S}{\partial\theta}B^TF^TZ\right]}^{7,8,9} \quad (31)$$

$$\Sigma^{-1}\frac{\partial^2\Sigma}{\partial\theta_1\partial\theta_2}C = \overbrace{\left[\Sigma^{-1}FB\frac{\partial A}{\partial\theta_1}\right]}^{10}\overbrace{\left[B\frac{\partial A}{\partial\theta_2}EF^TC\right]}^{11} \quad (32)$$

$$+ \overbrace{\left[\Sigma^{-1}FB\frac{\partial A}{\partial\theta_2}\right]}^{10}\overbrace{\left[B\frac{\partial A}{\partial\theta_1}EF^TC\right]}^{11}$$

$$+ \overbrace{\left[\Sigma^{-1}FB\frac{\partial A}{\partial\theta_1}\right]}^{10}\overbrace{\left[B\frac{\partial S}{\partial\theta_2}B^TF^TC\right]}^{12} \quad (33)$$

$$+ \overbrace{\left[\Sigma^{-1}FB\frac{\partial A}{\partial\theta_1}\right]}^{10}\overbrace{\left[E\left(\frac{\partial A}{\partial\theta_2}\right)^TB^TF^TC\right]}^{13} \quad (34)$$

$$+ \overbrace{\left[\Sigma^{-1}FB\frac{\partial A}{\partial\theta_2}\right]}^{10}\overbrace{\left[B\frac{\partial S}{\partial\theta_1}B^TF^TC\right]}^{12} \quad (35)$$

$$+ \overbrace{\left[\Sigma^{-1}FE\left(\frac{\partial A}{\partial\theta_2}\right)^TB^T\right]}^{14}\overbrace{\left[\left(\frac{\partial A}{\partial\theta_1}\right)^TB^TF^TC\right]}^{15} \quad (36)$$

$$+ \overbrace{\left[\Sigma^{-1}FE\left(\frac{\partial A}{\partial\theta_1}\right)^TB^T\right]}^{14}\overbrace{\left[\left(\frac{\partial A}{\partial\theta_2}\right)^TB^TF^TC\right]}^{15} \quad (37)$$

$$+ \overbrace{\left[\Sigma^{-1}FB\frac{\partial S}{\partial\theta_2}B^T\right]}^{16}\overbrace{\left[\left(\frac{\partial A}{\partial\theta_1}\right)^TB^TF^TC\right]}^{15} \quad (38)$$

$$+ \overbrace{\left[\Sigma^{-1}FB\frac{\partial A}{\partial\theta_2}\mathrm{E}\right]}^{17}\overbrace{\left[\left(\frac{\partial A}{\partial\theta_1}\right)^TB^TF^TC\right]}^{15} \quad (39)$$

$$+ \overbrace{\left[\Sigma^{-1}FB\frac{\partial S}{\partial\theta_1}B^T\right]}^{16}\overbrace{\left[\left(\frac{\partial A}{\partial\theta_2}\right)^TB^TF^TC\right]}^{15} \quad (40)$$

The Z in Lines 30 and 31 is either identity, $\Sigma^{-1}D$, or C. Only two of those three—that is, 6 $[XYZ]$ terms—need to be computed, since $C=I-\Sigma^{-1}D$. The second derivative term has 8 unique $[XYZ]$ terms. In total, 14 such terms need to be prepared before starting the actual computation.

To summarize the algorithm, we first have to compute the matrices B, C, E, and Σ. This takes $O(K^3)$ steps. We then prepare the 14 $[XYZ]$ terms in Lines 30–40 for all P parameters in parallel, which needs $O(P'K^2)$ operations, where P' is the number of nonconstant entries in A and S. Using these matrices, we compute the trace in Equation 22 in $O(PK)$ steps and in Equation 23 in $O(P^2K^2)$ steps. Finally, adding the mean contributions to the gradient and Hessian requires a number of matrix-vector multiplications. These computations take $O(PK^2)$ and $O(P^2K^2)$ steps, respectively. In total, the running time of the algorithm is $O(P^2K^2+K^3+P'K^2)$.

Speed comparison in simulation

To validate the method and to test its efficiency, we simulated data from a linear latent growth curve model (LGCM; Laird &

Ware, 1982; McArdle & Epstein, 1987). The model is shown in Fig. 1. We systematically varied the number of observations M measured per participant in the LGCM from 10 to 100 in steps of 10. Note that 100 observations per individual is realistic even for behavioral data (Schmiedek & Lindenberger, 2010) and can be considered rather small for multivariate analysis of neurological data (Schmitt et al., 2007). Note that while the power to find an effect in the fitted model depends on the number of participants observed (Prindle & McArdle, 2012), the Hessian computation speed depends only on the number of variables in the model.

The total number of variables is $K=M+2$, the M manifest and the two latent variables. The total number of parameters is $P=M+3$, the variances and covariance of intercept and slope, and one variance parameter for the measurement error at each measurement occasion. In this situation, the A matrix is constant, and the S matrix contains the two by two covariance matrix of the latent variables plus all measurement error variances as parameterized entries. As a result, we have $P'=M+4=K+2$ in total.

We computed the Hessian of the minus two log likelihood 100 times using the free SEM software Onyx (von Oertzen et al., 2012), either by numerical approximation via P^2 calls of the minus two log likelihood function or by the sparse derivative analytical method. Figure 2 (top panel) shows the resulting average time to compute these for the numerical (dotted line) and the sparse derivative (solid line) method. The lower panel shows the same graph with the lower part of the y-axis enlarged to make the curvature of the sparse derivative algorithm more visible. Table 1 gives the numerical values of the computation times in milliseconds. Even for only 10 observations, the sparse computation method is already 10 times faster. It outperforms numerical approximation by far for larger models, in addition to providing a more precise result.

For increasing K, we expect the naive algorithm to run in $O(K^5)$ steps and the sparse density algorithm to run in $O(K^4)$ steps. To validate this, Fig. 3 gives a log-log plot of the same results as Fig. 2. Again, the dotted line is the log-log line of the numerical algorithm, the solid line the sparse derivative algorithm. As was expected, we see straight lines for both methods, since both are polynomial in K after K exceeds an initial range. The average linear slope of the log-log plot for the numerical algorithm is 4.70, indicating approximately the

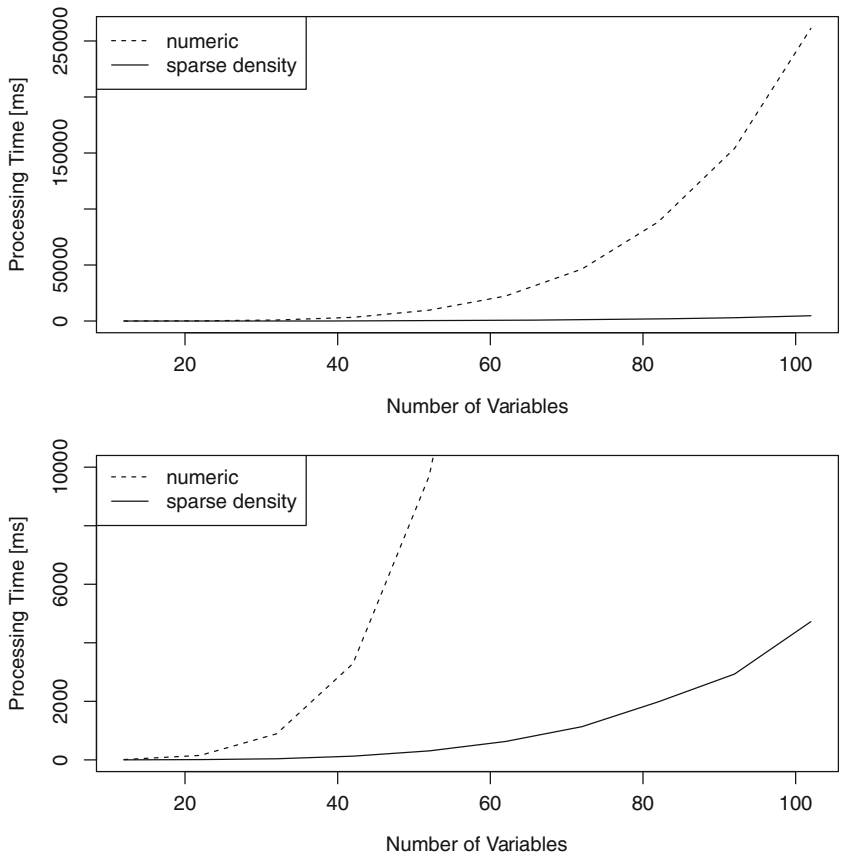

Fig. 2 Average time needed to compute the Hessian of the minus two log likelihood for a Latent growth curve model in reticular action model specification in Onyx. The dotted line is the result for the numerical computation, while the solid line gives the result with the sparse derivative computation. The horizontal axis is the number of variables, including two latent variables. The top panel gives the full range of values for the vertical axis; the lower panel enlarges the lower range of the vertical axis

Table 1 Average time in milliseconds needed to compute the Hessian of a minus two log likelihood for a latent growth curve model in reticular action model specification in Onyx. The numbers are graphically shown in Fig. 2

Observations	Numerical	Sparse Derivative
10	10.7	1.1
20	153.5	9.4
30	891.2	37.6
40	3,289.0	126.6
50	9,705.0	307.4
60	22,320.0	628.6
70	46,410.0	1133.9
80	88,470.0	1982.0
90	154,100.0	2933.0
100	261,300.0	4723.2

expected result of 5. The log-log plot for the sparse derivative algorithm is 3.84, also close to the expected result of 4.

To demonstrate the advantage of the sparse Hessian computation for actual maximum likelihood optimization, we simulated 100 data sets for the above models, with between $M=5$ and $M=40$ observations, and performed a complete maximum likelihood estimation both with a numerical Hessian computation and with the sparse Hessian computation in Onyx. We expected a double gain for the sparse Hessian computation, (1) since each single Hessian computation runs faster and (2) because of the higher precision of the Hessian, fewer calls are needed for the estimates to converge.

Figure 4 shows the time needed for the maximum likelihood optimizations. As was expected, the time differences are even more pronounced in a complete estimation process than in the single Hessian computation shown in Fig. 2.

To show the robustness of the method, we repeated the Hessian computation from Fig. 2, using the free statistical software OpenMx (Boker et al., 2011). OpenMx's existing Hessian computation method can perform the computation in parallel and spread work across a number of computer processors. We wanted to demonstrate that the advantage of the sparse Hessian computation adds to the parallelization advantage without negative interference. Figure 5 shows the result of the Hessian computation for the same model as above in OpenMx, without parallelization (dotted/gray solid line) and with parallelization (dashed/dark solid line) on an 8-core machine. While the asymptotic behavior of both algorithms is not influenced by parallelization, the individual compute times are 6 to 8 times faster when using parallelization both with and without sparse Hessian computation. It can be seen that the sparse Hessian computation outperforms numerical

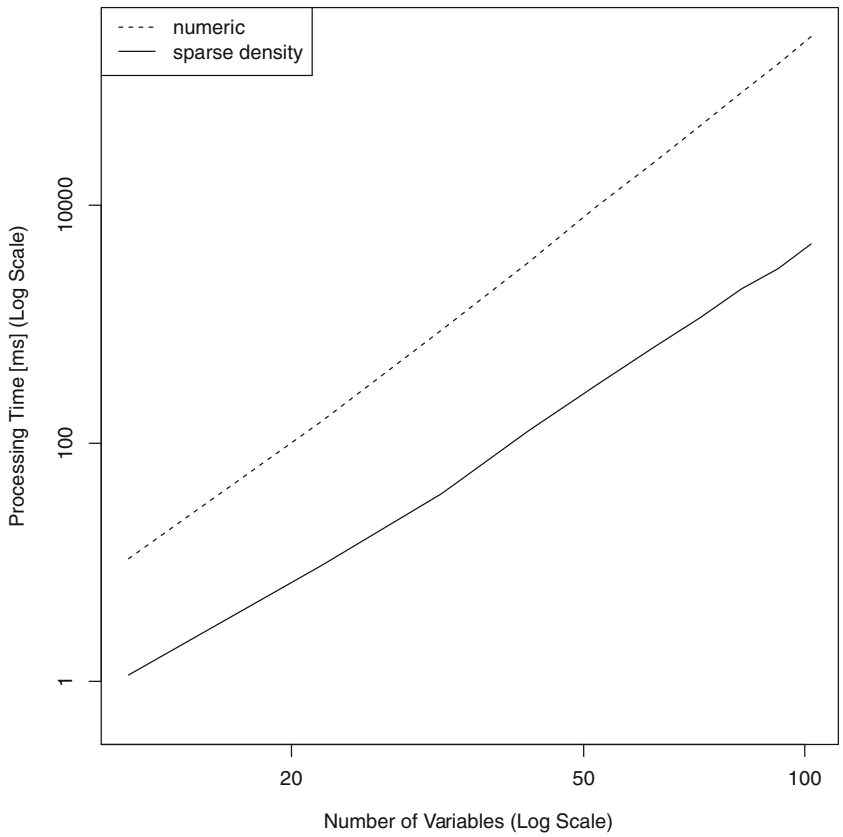

Fig. 3 Log-log plot of the number of variables and the computation time for the numerical and the sparse derivative computation

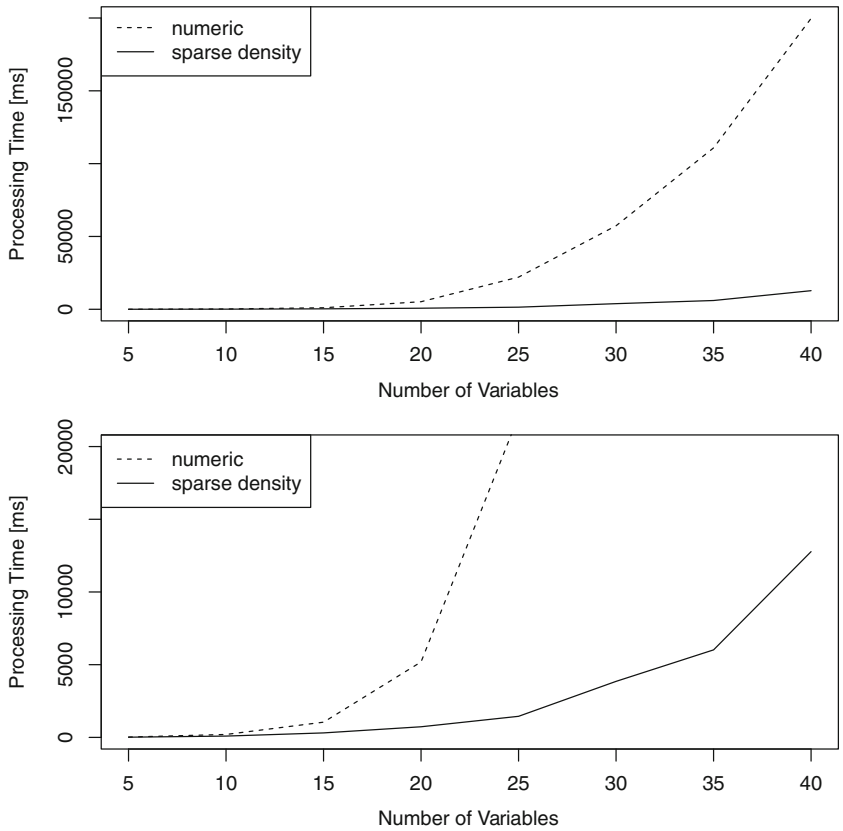

Fig. 4 Average time for a complete maximum likelihood estimation with numerical Hessian computations (*dotted line*) and sparse Hessian computation (*solid line*). The top panel gives the full range of values for the vertical axis; the lower panel enlarges the lower range of the vertical axis

approximation even if the sparse computation is done on a single processor and the numerical approximation is done in parallel.

The increase in speed provided by the sparse derivative computation method is particularly important in processes with repeated optimizations, as, for example, in a bootstrap method (Efron, 1979) or a Monte Carlo simulation. Assume, for example, a simulation with 1,000 repetitions on an LGCM as in the above example with 20 observations. With numerical Hessian computation, this process would take 1.5 h. With the sparse derivative method, the time would be reduced to 12 min. For a larger LGCM with 40 observations, the process would take 3.5 h, instead of 2.5 days.

Discussion

In this article, we presented the sparse derivative algorithm to compute the minus two log likelihood for SEM specifications in RAM notation and its first two derivatives more efficiently than with previously known algorithms. In typical cases, the method is one order of magnitude faster; in our example for an LGCMwith 100 observations, it is about 50 times faster. This speed increase is possible because of the sparse structure of the derivatives of the A and S matrices in RAM notation, which constitutes a major advantage of RAM that, so far, has not been exploited computationally.

The speed gain for the sparse density algorithm is most pronounced in large SEMs. Some of those are already used today (Schmiedek & Lindenberger, 2010), and new uses for SEMs that include large models are proposed that may exceed 100 observations by far (Voelkle et al., 2012). In addition, there are multiple data sets—for example, large genome studies (Laird & Ware, 1982)—in which SEMs might provide increased power and theoretical insights but which are, to date, beyond the range of statistical programs (Evans, 2002; Kaabi et al., 2006). For such applications, it is crucial to apply all possible means of increasing efficiency of fitting algorithms, including taking advantage of the sparsity of matrix derivatives in RAM notation.

There are special cases in which the sparse derivative method as described here is as bad as or even worse than the naive method. The method is designed for cases in which the numbers of parameters, P, is on the order of magnitude of the number of variables, K. For small numbers of parameters, the term $O(P^2K^3)$ of standard algorithms approaches the $O(K^3)$ term in the present method. Even worse, if P is very small and P' is in the order of magnitude of K^2 (i.e., most entries of S are parameters that are used repeatedly), then $O(P'K^2)$ approaches $O(K^4)$, which is worse than the naive algorithm

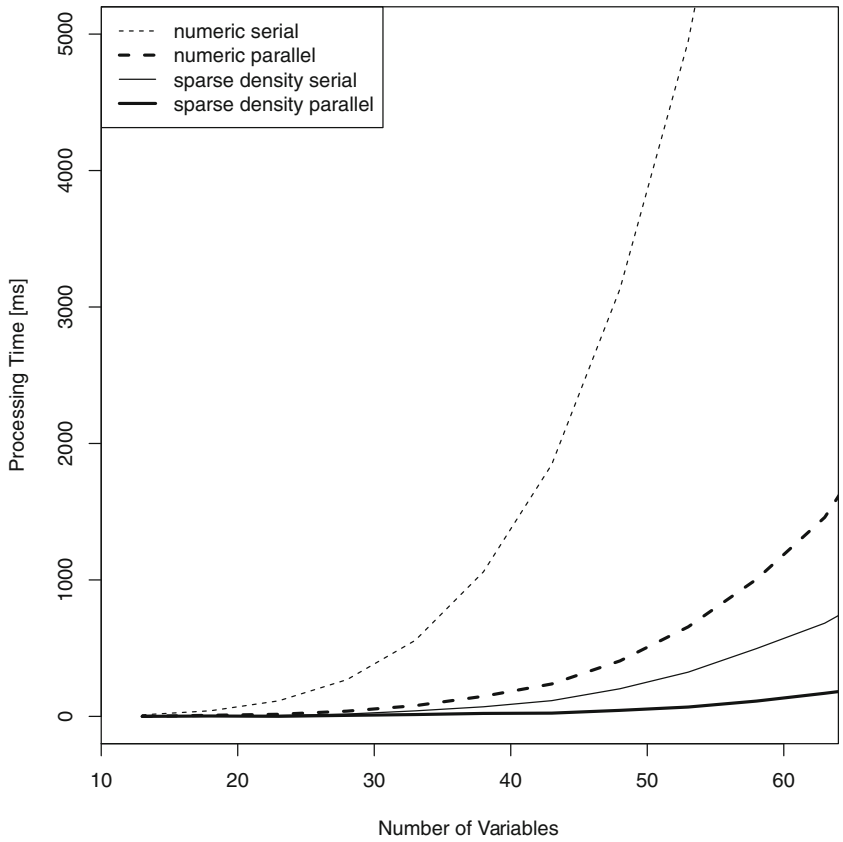

Fig. 5 Comparison of Hessian computation time using a numerical and sparse Hessian computation with (*dashed/dark solid line*) and without (*dotted/gray solid line*) parallelization. The parallel versions are consistently 6 to 8 times faster. This advantage does not interact with the speed gain using the analytical sparse Hessian algorithm

for low P. However, these cases are easy to recognize automatically. The sparse derivative methods should be used only if either P is large or P' is low. In models typically used in psychology, S and A are sparse themselves, which implies small P' to begin with.

The present article concentrates on the theoretical derivation of the sparse derivative method. The algorithm has been implemented in the SEM software Onyx based on a prototype in LOGH (von Oertzen, Ghisletta, & Lindenberger, 2010) and in the open-source OpenMx software. At time of writing, it is expected that these advances will be available in the next version of OpenMx.

Appendix

The purpose of this appendix is to expand computations where the full expansion would impede reading in the main article. The following derivative rules for matrices X is used repeatedly:

$$\frac{\partial X^{-1}}{\partial \theta} = -X^{-1}\frac{\partial X}{\partial \theta}X^{-1}. \tag{41}$$

The following is the expansion of the derivative of B in Equation 11,

$$B = (I-A)^{-1} \tag{42}$$

$$\frac{\partial B}{\partial \theta} = -(I-A)^{-1}\frac{\partial (I-A)}{\partial \theta}(I-A)^{-1} \tag{43}$$

$$= -(I-A)^{-1}(-1)\frac{\partial A}{\partial \theta}(I-A)^{-1} \tag{44}$$

$$= B\frac{\partial A}{\partial \theta}B, \tag{45}$$

C in Equation 12,

$$C = \left(I-\Sigma^{-1}D\right) \tag{46}$$

$$\frac{\partial C}{\partial \theta} = -\frac{\partial \Sigma^{-1}D}{\partial \theta} \tag{47}$$

$$= -\frac{\partial \Sigma^{-1}}{\partial \theta}D \tag{48}$$

$$= \Sigma^{-1}\frac{\partial \Sigma}{\partial \theta}\Sigma^{-1}D, \tag{49}$$

and E in Equation 13,

$$E = BSB^T \tag{50}$$

$$\frac{\partial E}{\partial \theta} = \frac{\partial B}{\partial \theta} S B^T + B \frac{\partial S}{\partial \theta} B^T + BS \left(\frac{\partial B}{\partial \theta} \right)^T \tag{51}$$

$$= \left[B \frac{\partial A}{\partial \theta} BSB^T \right]^{sym} + B \frac{\partial S}{\partial \theta} B^T \tag{52}$$

$$= \left[B \frac{\partial A}{\partial \theta} E \right]^{sym} + B \frac{\partial S}{\partial \theta} B^T, \tag{53}$$

where, again, $X^{sym}=X+X^T$. Since Σ and E are identical up to the F on both sides, the derivative of Σ in Equation 15 is as above with additional Fs on both sides. The first term in Equation 15 in the $[.]^{sym}$ operator, differentiated for a second parameter θ_2 gives

$$\frac{\partial \left(FB \frac{\partial A}{\partial \theta_1} EF^T \right)}{\partial \theta_2} = F \frac{\partial B}{\partial \theta_2} \frac{\partial A}{\partial \theta_1} EF^T + FB \frac{\partial^2 A}{\partial \theta_1 \partial \theta_2} EF^T + FB \frac{\partial A}{\partial \theta_1} \frac{\partial E}{\partial \theta_2} F^T \tag{54}$$

$$= FB \frac{\partial A}{\partial \theta_2} B \frac{\partial A}{\partial \theta_1} EF^T + FB \frac{\partial A}{\partial \theta_1} \left[B \frac{\partial A}{\partial \theta_2} E \right]^{sym} F^T \tag{55}$$

$$+ FB \frac{\partial A}{\partial \theta_1} B \frac{\partial S}{\partial \theta_2} B^T F^T \tag{56}$$

$$= FB \frac{\partial A}{\partial \theta_2} B \frac{\partial A}{\partial \theta_1} EF^T + FB \frac{\partial A}{\partial \theta_1} B \frac{\partial A}{\partial \theta_2} EF^T \tag{57}$$

$$FB \frac{\partial A}{\partial \theta_1} E \left(\frac{\partial A}{\partial \theta_2} \right)^T B^T F^T + FB \frac{\partial A}{\partial \theta_1} B \frac{\partial S}{\partial \theta_2} B^T F^T. \tag{58}$$

The second derivative of the second term in Equation 15 is

$$\frac{\partial \left(FB \frac{\partial S}{\partial \theta_1} B^T F^T \right)}{\partial \theta_2} = F \frac{\partial B}{\partial \theta_2} \frac{\partial S}{\partial \theta_1} B^T F^T + FB \frac{\partial^2 S}{\partial \theta_1 \partial \theta_2} B^T F^T + FB \frac{\partial S}{\partial \theta_1} \frac{\partial B^T}{\partial \theta_2} F^T \tag{59}$$

$$= FB \frac{\partial A}{\partial \theta_2} B \frac{\partial S}{\partial \theta_1} B^T F^T + FB \frac{\partial S}{\partial \theta_1} B^T \left(\frac{\partial A}{\partial \theta_2} \right)^T B^T F^T \tag{60}$$

$$= \left[FB \frac{\partial A}{\partial \theta_2} B \frac{\partial S}{\partial \theta_1} B^T F^T \right]^{sym}. \tag{61}$$

Since the first term is in a $[.]^{sym}$ operator, the complete second derivative of Σ is given as in Equations 16–18 (note that θ_1 and θ_2 are exchanged for better readability):

$$\frac{\partial^2 \Sigma}{\partial \theta_1 \partial \theta_2} = \left[FB \frac{\partial A}{\partial \theta_1} B \frac{\partial A}{\partial \theta_2} EF^T + FB \frac{\partial A}{\partial \theta_2} B \frac{\partial A}{\partial \theta_1} EF^T \right. \tag{62}$$

$$+ FB \frac{\partial A}{\partial \theta_1} B \frac{\partial S}{\partial \theta_2} B^T F^T + FB \frac{\partial A}{\partial \theta_1} E \left(\frac{\partial A}{\partial \theta_2} \right)^T B^T F^T \tag{63}$$

$$\left. + FB \frac{\partial A}{\partial \theta_2} B \frac{\partial S}{\partial \theta_1} B^T F^T \right]^{sym}. \tag{64}$$

For the first derivative of Equation 6, we use the matrix derivative

$$\frac{\partial \log\det(\Sigma)}{\partial \theta} = Tr \left(\Sigma^{-1} \frac{\partial \Sigma}{\partial \theta} \right). \tag{65}$$

Using also $b=(d-\mu)$ as before, the first derivative of Equation 6 is expanded as

$$\mathcal{L} = \ln\left((2\pi)^M det(\Sigma) \right) + \mathrm{Tr}\left(\Sigma^{-1} D \right) + b^T \Sigma^{-1} b \tag{66}$$

$$\frac{\partial \mathcal{L}}{\partial \theta} = \mathrm{Tr}\left(\Sigma^{-1} \frac{\partial \Sigma}{\partial \theta} \right) + \mathrm{Tr}\left(-\Sigma^{-1} \frac{\partial \Sigma}{\partial \theta} \Sigma^{-1} D \right) \tag{67}$$

$$- \left(\frac{\partial \mu}{\partial \theta} \right)^T \Sigma^{-1} b - b^T \Sigma^{-1} \frac{\partial \Sigma}{\partial \theta} \Sigma^{-1} b - b^T \Sigma^{-1} \frac{\partial \mu}{\partial \theta} \tag{68}$$

$$= \mathrm{Tr}\left(\Sigma^{-1} \frac{\partial \Sigma}{\partial \theta} \left(I - \Sigma^{-1} D \right) \right) - b^T \Sigma^{-1} \frac{\partial \Sigma}{\partial \theta} \Sigma^{-1} b - 2 \left(\frac{\partial \mu}{\partial \theta} \right)^T \Sigma^{-1} b \tag{69}$$

$$= \mathrm{Tr}\left(\Sigma^{-1} \frac{\partial \Sigma}{\partial \theta} C \right) - \left(b^T \Sigma^{-1} \frac{\partial \Sigma}{\partial \theta} + 2 \left(\frac{\partial \mu}{\partial \theta} \right)^T \right)^T \Sigma^{-1} b, \tag{70}$$

which is Equation 22. The second derivative of the trace term in this equation is

$$\frac{\partial \left(\Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_1} C \right)}{\partial \theta_2} = -\Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_2} \Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_1} C + \Sigma^{-1} \frac{\partial^2 \Sigma}{\partial \theta_1 \partial \theta_2} C + \Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_1} \Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_2} \Sigma^{-1} D. \tag{71}$$

To separate the remaining computation in two parts, we concentrate on the two remaining terms in Line 69. The second derivative of the middle term is

$$\frac{\partial\left(b^T\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_1}\Sigma^{-1}b\right)}{\partial\theta_2} = -2\left(\frac{\partial\mu}{\partial\theta_2}\right)^T\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_1}\Sigma^{-1}b - b^T\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_2}\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_1}\Sigma^{-1}b \tag{72}$$

$$+b^T\Sigma^{-1}\frac{\partial^2\Sigma}{\partial\theta_1\partial\theta_2}\Sigma^{-1}b - b^T\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_1}\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_2}\Sigma^{-1}b. \tag{73}$$

The second and fourth terms are identical because Σ is symmetrical and $b^TXb = b^TX^Tb$. This allows us to exchange the derivative terms while leaving the quantities equivalent.

The derivative of the third term in line 69 expands to

$$\frac{\partial\left(2\left(\frac{\partial\mu}{\partial\theta_1}\right)^T\Sigma^{-1}b\right)}{\partial\theta_2} = 2\left(\frac{\partial^2\mu}{\partial\theta_1\partial\theta_2}\right)^T\Sigma^{-1}b - 2\left(\frac{\partial\mu}{\partial\theta_1}\right)^T\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_2}\Sigma^{-1}b \tag{74}$$

$$-2\left(\frac{\partial\mu}{\partial\theta_1}\right)^T\Sigma^{-1}\left(\frac{\partial\mu}{\partial\theta_2}\right)^T. \tag{75}$$

Note that the second and third terms in Line 69 are negative. Taking the second derivative of $\mathcal{L}$ together gives Lines 23–26:

$$\frac{\partial^2\mathcal{L}}{\partial\theta_1\partial\theta_2} = \mathrm{Tr}\left(-\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_2}\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_1}C + \Sigma^{-1}\frac{\partial^2\Sigma}{\partial\theta_1\partial\theta_2}C + \Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_1}\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_2}\Sigma^{-1}D\right) \tag{76}$$

$$+b^T\left(2\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_1}\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_2}\Sigma^{-1} - \Sigma^{-1}\frac{\partial^2\Sigma}{\partial\theta_1\partial\theta_2}\Sigma^{-1}\right)b \tag{77}$$

$$+2\left(\frac{\partial\mu}{\partial\theta_1}\right)^T\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_2}\Sigma^{-1}b + 2\left(\frac{\partial\mu}{\partial\theta_2}\right)^T\Sigma^{-1}\frac{\partial\Sigma}{\partial\theta_1}\Sigma^{-1}b \tag{78}$$

$$+2\left(\frac{\partial\mu}{\partial\theta_1}\right)^T\Sigma^{-1}\left(\frac{\partial\mu}{\partial\theta_2}\right)^T - 2\left(\frac{\partial^2\mu}{\partial\theta_1\partial\theta_2}\right)^T\Sigma^{-1}b. \tag{79}$$

References

Baltes, P. B., Reese, H. W., & Nesselroade, J. R. (1988). *Life-span developmental psychology: Introduction to research methods (reprint of* (1977th ed.). Hillsdale: Lawrence Erlbaum Associates.

Boker, S., Neale, M., Maes, H., Wilde, M., Spiegel, M., Brick, T.,… Fox, J. (2011). OpenMx: An open source extended structural equation modeling software. *Psychometrika*, *76*(2), 306–317.

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *Annals of Statistics, 7*(1), 1–26.

Evans, D. (2002). The power of multivariate quantitative-trait loci linkage analysis is influenced by the correlation between variables. *American Journal of Human Genetics, 70*(6), 1599–1602.

Fox, J. (2006). Structural equation modeling with the sem package in r. *Structural Equation Modeling, 13,* 465–486.

Gill, P. E., Murray, W., Saunders, M. A., Wright, M. H. (2001). User's guide for NPSOL 5.0: A Fortran package for nonlinear programming. San Diego: University of California.

Grimm, K. J., & McArdle, J. J. (2005). A note on the computer generation of mean and covariance expectations in latent growth curve analysis. In F. Danserau & F. Yammarino (Eds.), *Multi-level issues in strategy and methods* (pp. 335–364). New York: Elsiver.

Jöreskog, K. G., & Sörbom, D. (1993). *New features in LISREL 8*. Chicago: Scientific Software.

Kaabi, B., Gelernter, J., Woods, S., Goddard, A., Page, G., & Elston, R. (2006). Genome scan for loci predisposing to anxiety disorders using a novel multivariate approach: Strong evidence for a chromosome 4 risk locus. *American Journal of Human Genetics, 78*(4), 543–553.

Laird, N. M., & Ware, J. H. (1982). Random effect models for longitudinal data. *Biometrics, 38,* 963–974.

McArdle, J., & McDonald, R. (1984). Some algebraic properties of the reticular action model for moment structures. *British Journal of Mathematical and Statistical Psychology, 37,* 234–251.

McArdle, J. J. (2005). The development of RAM rules for latent variable structural equation modeling. In A. Madeau (Ed.), *Contemporary advances in psychometrics* (pp. 225–273). Mahwah: Lawrence Erlbaum Associates, Inc.

McArdle, J. J., & Boker, S. M. (1990). *Rampath: A computer program for automatic path diagrams*. Hillsdale: Lawrence Erlbaum Publishers.

McArdle, J. J., & Epstein, D. B. (1987). Latent growth curves within developmental structural equation models. *Child Development, 58,* 110–133.

Muthén, L. K., & Muthén, B. O. (2004). *MPlus user's guide*. Los Angeles: Muthén and Muthén.

von Oertzen, T., Brandmaier, A. M., & Tsang, S. (2012). *The onyx manual.* Onyx Developer Team. (to be found at onyx.brandmaier.de)

von Oertzen, T., Ghisletta, P., & Lindenberger, U. (2010). Simulating statistical power in latent growth curve modeling: A strategy for evaluating age-based changes in cognitive resources. In M. Crocker & J. Siekmann (Eds.), *Resource adaptive cognitive processes* (pp. 95–117). Heidelberg: Springer–Verlag.

Pinheiro, J. C., & Bates, D. M. (2000). *Mixed-effects models in S and S-Plus* (1st ed.). New York: Springer.

Prindle, J., & McArdle, J. (2012). An analysis of statistical power in dynamic structural equation models. *Structural Equation Modeling, 19*(3), 351–371.

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software, 48*(2), 1–36.

Schmiedek, F., & Lindenberger, U. (2010). of cognitive training enhance broad cognitive abilities in adulthood: Findings from the cogito study. *Frontiers in Aging Neuroscience, 2,* 1–10.

Schmitt, J. E., Wallce, G., Rosenthal, M., Molloy, E., Ordaz, S., Lenroot, R.,…, Giedd, J. (2007). multivariate analysis of neuroanatomic relationships in a genetically infromative pediatric sample. *Neuroimage*, *35*, 70–82.

Stelzl, I. (1986). Changing a causal hypothesis without changing the fit: Some rules for generating equivalent path models. *Multivariate Behavioral Research, 21,* 309–331.

Voelkle, M., Oud, J., von Oertzen, T., & Lindenberger, U. (2012). Maximum likelihood dynamic factor modeling for arbitrary N and T using SEM. *Structural Equation Modeling, 19*(3), 329–350.

Wright, S. (1934). The method of path coefficients. *Annals of Mathematical Statistics, 5,* 161–215.