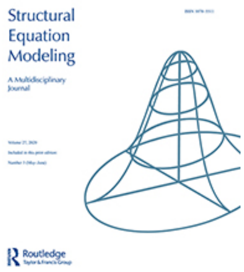

Structural Equation Modeling: A Multidisciplinary Journal

ISSN: 1070-5511 (Print) 1532-8007 (Online) Journal homepage: www.tandfonline.com/journals/hsem20

Integrating Out Nuisance Parameters for Computationally More Efficient Bayesian Estimation – An Illustration and Tutorial

Martin Hecht, Christian Gische, Daniel Vogel & Steffen Zitzmann

To cite this article: Martin Hecht, Christian Gische, Daniel Vogel & Steffen Zitzmann (2020) Integrating Out Nuisance Parameters for Computationally More Efficient Bayesian Estimation – An Illustration and Tutorial, Structural Equation Modeling: A Multidisciplinary Journal, 27:3, 483-493, DOI: 10.1080/10705511.2019.1647432

To link to this article: https://doi.org/10.1080/10705511.2019.1647432

View supplementary material

Published online: 24 Sep 2019.

Submit your article to this journal

Article views: 3086

View related articles

View Crossmark data

Citing articles: 7 View citing articles

Structural Equation Modeling: A Multidisciplinary Journal, 27: 483–493, 2020
ISSN: 1070-5511 print / 1532-8007 online
DOI: https://doi.org/10.1080/10705511.2019.1647432

Routledge
Taylor & Francis Group

OPEN ACCESS

Integrating Out Nuisance Parameters for Computationally More Efficient Bayesian Estimation – An Illustration and Tutorial

Martin Hecht,[1] Christian Gische,[1] Daniel Vogel,[2] and Steffen Zitzmann[3]

[1]*Humboldt-Universität zu Berlin*
[2]*University of Aberdeen*
[3]*University of Kiel*

Bayesian estimation has become very popular. However, run time of Bayesian models is often unsatisfactorily high. In this illustration, we show how to reduce run time by (a) integrating out nuisance model parameters and by (b) reformulating the model based on covariances and means. The core concept is to use the sample scatter matrix which is in our case Wishart distributed with the model-implied covariance matrix as the scale matrix. To illustrate this approach, we choose the popular multi-level null (intercept-only) model, provide a step-by-step instruction on how to implement this model in a multi-purpose Bayesian software, and show how structural equation modeling techniques can be employed to bypass mathematically challenging derivations. A simulation study showed that run time is considerably reduced and an empirical example illustrates our approach. Further, we show how the JAGS sampling progress can be monitored and stopped automatically when convergence and precision criteria are reached.

Keywords: Bayesian analysis, run time optimization, nuisance parameters, multi-level modeling, structural equation modeling, sampler monitoring

Bayesian statistics is gaining in popularity in many disciplines and are used for many different purposes, for instance, to include previous knowledge, to estimate otherwise intractable models, to model uncertainty (Van de Schoot, Winter, Ryan, Zondervan-Zwijnenburg, & Depaoli, 2017), and to stabilize parameter estimates (e.g., Zitzmann, 2018). Although the advantages of Bayesian approaches are certainly well appreciated, a frequently encountered obstacle is the high run time that might prevent users from using Bayesian estimation. For instance, Hecht, Hardt, Driver, and Voelkle (2019) report run times of hours to days for rather small Bayesian longitudinal models. Similar problems were encountered by Lüdtke, Robitzsch, and Wagner (2018) who note that "very long chains (more than 2×10^6 iterations) […] provided only poor approximations of the posterior distributions (e.g., small effective sample size)" (p. 577). Those implementations had in common that individual random effects ("person parameters") were sampled. These person-specific parameters are usually an integral part of the model formulation. However, research questions oftentimes focus on group-level statistics such as means and (between-person) variances. Thus, for these scenarios, the person parameters are not of substantive interest and can, therefore, be considered as *nuisance parameters* (e.g., Bernardo & Smith, 2000). The lack of interest in

Correspondence should be addressed to Martin Hecht, Department of Psychology, Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany. E-mail: martin.hecht@hu-berlin.de

Supplemental data for this article can be accessed here.

nuisance parameters opens the door for eliminating them from the model formulation what might reduce run time when estimating the model. As "eliminating nuisance parameters from a model is universally recognized as a major problem of statistics" (Basu, 1977, p. 355), a large number of elimination methods have been proposed. One approach is to integrate out the nuisance parameters from the joint posterior and then to reformulate the model based on a model-implied covariance matrix (Lüdtke et al., 2018).

PURPOSE AND SCOPE

In the present work, we illustrate the method of integrating out nuisance parameters and covariance-based model reformulation for computationally more efficient Bayesian estimation. In a step-by-step tutorial fashion, we present detailed instructions and guidance on how to proceed with this method. For this illustrative purpose, we choose the Bayesian software JAGS for its flexibility and widespread use and the multi-level null model because it is easy to understand and it provides the building block for many more complex models. This illustration is targeted at all users of Bayesian software who want to optimize run time of models in which some of the parameters are not of interest. To show the run time reduction of the proposed method, we present results from a simulation study in which the implementation that includes nuisance parameters is compared to two nuisance-free implementations.

The article is organized into the following sections. First, we present all steps necessary to derive a nuisance-free model formulation of the multi-level null model including JAGS code snippets and structural equation modeling (SEM) formulations. Second, we describe our simulation study including an approach to monitor convergence criteria when running JAGS analyses. Third, we present an empirical example using data from the MIDUS 2 study (Ryff & Almeida, 2017). Fourth, we conclude with a discussion of the proposed method. Annotated R code to run the implementations is provided in the Online Supplemental Material.

STEP-BY-STEP ILLUSTRATION

Step 1: model with nuisance parameters

For this illustration, we choose a simple multi-level model that is often called "null model" or "intercept-only model" (e.g., Hox, 2010; Raudenbush & Bryk, 2002) or "hierarchical normal model" (Hoff, 2009). This model can be used to disentangle within-group variation from between-group variation. In our presentation, the groups are persons and the nested observations are repeated measurements of persons, but of course, the proposed method can also be identically applied in standard cross-sectional scenarios (e.g., persons nested in groups). Let y_{jp} (stacked into person-specific column vectors $\mathbf{y}_j$ of length P) be the value of person $j = 1, \ldots, J$ at measurement occasions $p = 1, \ldots, P$ (with J being the total number of persons and P being the total number of measurement occasions). The level 1 (within-person) model equation decomposes the observations into a person parameter, θ_j, representing the person-specific mean level over time, and normally distributed error terms, ε_{jp}: $y_{jp} = \theta_j + \varepsilon_{jp}$, $\varepsilon_{jp} \sim \mathcal{N}(0, \sigma_\varepsilon^2)$, with σ_ε^2 being the within-person variance. On level 2 (between-person), the person parameters are decomposed into a total mean, μ, and normally distributed error terms, u_j: $\theta_j = \mu + u_j$, $u_j \sim \mathcal{N}(0, \sigma_\theta^2)$, where σ_θ^2 denotes the between-person variance. The model can equivalently be represented by the following equations:

$$\text{Level 1 (within-person)}: y_{jp} \sim \mathcal{N}(\theta_j, \sigma_\varepsilon^2), \tag{1}$$

$$\text{Level 2 (between-person)}: \theta_j \sim \mathcal{N}(\mu, \sigma_\theta^2). \tag{2}$$

The likelihood function for this model is (adapted from Hoff, 2009, Chapter 8):

$$p(\mathbf{y}_1, \ldots, \mathbf{y}_J | \mu, \sigma_\theta^2, \sigma_\varepsilon^2, \theta_1, \ldots, \theta_J) = \left\{ \prod_{j=1}^{J} \prod_{p=1}^{P} p(y_{jp} | \theta_j, \sigma_\varepsilon^2) \right\} \left\{ \prod_{j=1}^{J} p(\theta_j | \mu, \sigma_\theta^2) \right\}, \tag{3}$$

with p in the bracketed terms denoting the probability density function of the normal distribution. The assumed prior distributions expressed by $p(\mu)$, $p(\sigma_\theta^2)$, and $p(\sigma_\varepsilon^2)$ are a normal distribution for μ and a gamma distribution for $\frac{1}{\sigma_\theta^2}$ and $\frac{1}{\sigma_\varepsilon^2}$, respectively (see Hoff, 2009, Chapter 8). In Bayesian models, the posterior distribution is proportional to the product of the likelihood function and the prior distributions. Thus, for the joint posterior distribution (i.e., the distribution of the model parameters given the data) in the Bayesian multi-level null model, the following proportionality statement holds:

$$p(\mu, \sigma_\theta^2, \sigma_\varepsilon^2, \theta_1, \ldots, \theta_J | \mathbf{y}_1, \ldots, \mathbf{y}_J) \propto p(\mathbf{y}_1, \ldots, \mathbf{y}_J | \mu, \sigma_\theta^2, \sigma_\varepsilon^2, \theta_1, \ldots, \theta_J) p(\mu) p(\sigma_\theta^2) p(\sigma_\varepsilon^2). \tag{4}$$

In order to translate this model into JAGS syntax, we need two components: a "likelihood specification" in accordance with the likelihood function and specifications of the prior distributions. To specify the terms from the likelihood function, we loop over persons and measurement occasions. Within the loop over persons and occasions the distributional assumption of observations y_{jp} given the model parameters θ_j and σ_ε^2 (contained in the first bracketed term in the likelihood function, see Equation 3) is specified, and within the person loop the distributional assumption of the person parameters θ_j given μ and σ^2_θ (contained in the second bracketed term) is specified:

```
# loop over persons
for (j in 1:J) {
    # loop over measurement occasions
    for (p in 1:P) {
        # distributional assumption of
        # observations (Equation 1)
        D[p,j] ~ dnorm(theta[j] , prec.eps)
    }
    # distributional assumption of the
    # person parameters (Equation 2)
    theta[j] ~ dnorm(mu, prec.theta)
}
```

where $\mathbf{D} = [\mathbf{y}_1 \ldots \mathbf{y}_j \ldots \mathbf{y}_J]$ is a $P \times J$ matrix, or, in other words, $\mathbf{D}$ is the sample data matrix in "wide format" with rows containing the measurement occasions, columns the persons, and cells the values of persons at measurement occasions. As an example, we choose rather uninformative prior distributions for μ, σ^2_θ, and σ^2_ε:

```
# prior distribution for mu
mu ~ dnorm(0, 1/10000)
# prior distribution for 1/sigma2.theta
prec.theta ~ dgamma(0.001, 0.001)
sigma2.theta <- 1/prec.theta
# prior distribution for 1/sigma2.eps
prec.eps ~ dgamma(0.001, 0.001)
sigma2.eps <- 1/prec.eps
```

We call this version of the Bayesian multi-level null model the *classic implementation*.

Step 2: integrating out nuisance parameters

The model formulation in Step 1 contains the person parameters θ_j. For data sets with a large number of persons, sampling of those parameters may slow down the model estimation considerably. Therefore, we eliminate those nuisance parameters from the original likelihood function (Equation 3) via integration to arrive at the likelihood function without the nuisance parameters θ_j:

$$p(\mathbf{y}_1, \ldots, \mathbf{y}_J | \mu, \sigma^2_\theta, \sigma^2_\varepsilon) =$$

$$\underset{\{(\theta_1,\ldots,\theta_J) \in \mathbf{R}^J\}}{\int \cdots \int} p(\mathbf{y}_1, \ldots, \mathbf{y}_J | \mu, \sigma^2_\theta, \sigma^2_\varepsilon, \theta_1, \ldots, \theta_J) d\theta_1 \ldots d\theta_J . (5)$$

Solving this integral (see Appendix A for a step-by-step walk-through and Appendix B for another justification of the derived results) yields the conditional distribution of $(\mathbf{y}_1, ..., \mathbf{y}_J)$ given the parameters μ, σ^2_θ, and σ^2_ε as

$$\mathbf{y}_j \sim \mathcal{N}_P(\boldsymbol{\mu}_P, \boldsymbol{\Sigma}), \quad j = 1, \ldots, J, \quad \text{independent}, \tag{6}$$

where $\boldsymbol{\mu}_P$ is a column vector of length P with all elements being μ and $\boldsymbol{\Sigma}$ is a symmetric $P \times P$ covariance matrix with all off-diagonal elements being σ^2_θ and all diagonal elements being $\sigma^2_\theta + \sigma^2_\varepsilon$, that is,

$$\begin{aligned}\boldsymbol{\Sigma} &= \sigma^2_\theta \mathbf{1}_P \mathbf{1}'_P + \sigma^2_\varepsilon \mathbf{I}_P \\ &= \begin{pmatrix} \sigma^2_\theta & \cdots & \sigma^2_\theta \\ \vdots & & \vdots \\ \sigma^2_\theta & \cdots & \sigma^2_\theta \end{pmatrix} + \begin{pmatrix} \sigma^2_\varepsilon & & 0 \\ & \ddots & \\ 0 & & \sigma^2_\varepsilon \end{pmatrix} \\ &= \begin{pmatrix} \sigma^2_\theta + \sigma^2_\varepsilon & & \sigma^2_\theta \\ & \ddots & \\ \sigma^2_\theta & & \sigma^2_\theta + \sigma^2_\varepsilon \end{pmatrix},\end{aligned} \tag{7}$$

where $\mathbf{1}_P$ is a column vector of ones with length P and $\mathbf{I}_P$ is the identity matrix of size P. The corresponding precision matrix, $\mathbf{C}$, is then a symmetric $P \times P$ matrix with all off-diagonal elements being $-\frac{\sigma^2_\theta}{\xi}$ and all diagonal elements being $\frac{(P-1)\sigma^2_\theta + \sigma^2_\varepsilon}{\xi}$, that is,

$$\begin{aligned}\mathbf{C} = \boldsymbol{\Sigma}^{-1} &= -\frac{\sigma^2_\theta}{\xi} \mathbf{1}_P \mathbf{1}'_P + \frac{P\sigma^2_\theta + \sigma^2_\varepsilon}{\xi} \mathbf{I}_P \\ &= \begin{pmatrix} -\frac{\sigma^2_\theta}{\xi} & \cdots & -\frac{\sigma^2_\theta}{\xi} \\ \vdots & & \vdots \\ -\frac{\sigma^2_\theta}{\xi} & \cdots & -\frac{\sigma^2_\theta}{\xi} \end{pmatrix} + \begin{pmatrix} \frac{P\sigma^2_\theta + \sigma^2_\varepsilon}{\xi} & & 0 \\ & \ddots & \\ 0 & & \frac{P\sigma^2_\theta + \sigma^2_\varepsilon}{\xi} \end{pmatrix}, \\ &= \begin{pmatrix} \frac{(P-1)\sigma^2_\theta + \sigma^2_\varepsilon}{\xi} & & -\frac{\sigma^2_\theta}{\xi} \\ & \ddots & \\ -\frac{\sigma^2_\theta}{\xi} & & \frac{(P-1)\sigma^2_\theta + \sigma^2_\varepsilon}{\xi} \end{pmatrix},\end{aligned} \tag{8}$$

$$\text{with } \xi = P\sigma^2_\theta \sigma^2_\varepsilon + \sigma^4_\varepsilon . \tag{9}$$

This formulation could easily be run with JAGS, but sampling J times from a multivariate normal distribution (Equation 6) with non-zero off-diagonals in the precision matrix is computationally less efficient than sampling $(JP + J)$ times from a univariate normal distribution as in the classic formulation (Equations 1 and 2). Thus, in the next step, the model is reformulated in terms of covariances.

Shortcut via structural equation modeling

Integrating out parameters from the posterior distribution (resp. the "likelihood" part of it) might become challenging and may require advanced mathematical skills.

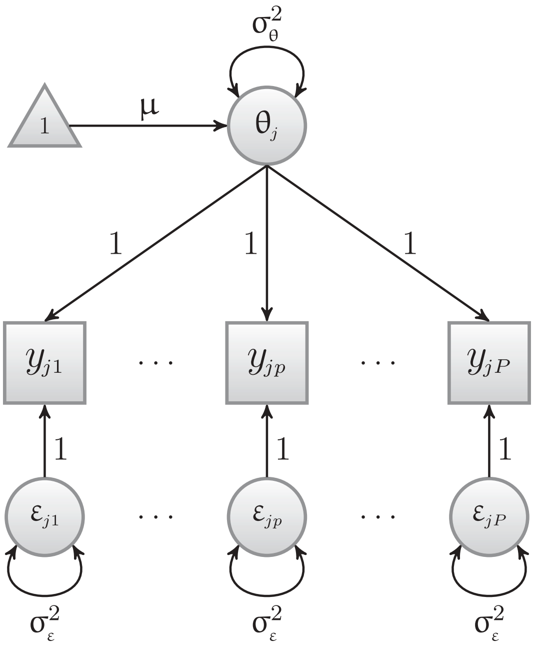

FIGURE 1 Graphical SEM representation of the multi-level null model.

Fortunately, the SEM framework offers easy to obtain solutions to derive the model-implied covariance matrix for a wide variety of models without the nuisance person parameters. A good starting point is usually to represent the model of interest as a path diagram using SEM conventions. The multi-level null model can be depicted as a path diagram as shown in Figure 1 (for an introduction to multi-level structural equation modeling, see, e.g., Mehta & Neale, 2005). Based on such a model representation, we can use tracing rules (Wright, 1934; see also, e.g., Chapter 7 in Kline, 2016) and calculation rules for variances and covariances (e.g., Steyer & Nagel, 2017, Chapter 6) to obtain the model-implied covariance structure. For the presented model, we yield:

$$\begin{aligned} \mathrm{Var}(y_{jp}) &= \mathrm{Var}(1\cdot\theta_j + 1\cdot\varepsilon_{jp}) \\ &= \mathrm{Var}(\theta_j) + \mathrm{Var}(\varepsilon_{jp}) + \mathrm{Cov}(\theta_j,\varepsilon_{jp}) \\ &= \sigma^2_\theta + \sigma^2_e \\ \mathrm{Cov}(y_{jk},y_{jl}) &= \mathrm{Cov}(1\cdot\theta_j + 1\cdot\varepsilon_{jk}, 1\cdot\theta_j + 1\cdot\varepsilon_{jl}) \\ &= \mathrm{Cov}(\theta_j,\theta_j) + \mathrm{Cov}(\theta_j,\varepsilon_{jl}) + \mathrm{Cov}(\varepsilon_{jk},\theta_j) \\ &\quad + \mathrm{Cov}(\varepsilon_{jk},\varepsilon_{jl}) \\ &= \sigma^2_\theta, \qquad \text{for k}\neq\text{l}. \end{aligned}$$

Alternatively, we could use an SEM framework to obtain the model-implied covariance matrix. In the Appendices C and D, this is shown within the LISREL (Jöreskog, Olsson, & Wallentin, 2016) and the RAM (McArdle & McDonald, 1984; see also Boker, 2019) framework.

Step 3: covariance-based model reformulation

The basic idea of this step is to find the distribution of the sample covariance matrix or of a related matrix for the model of interest. The sample covariance matrix, **Q**, is (e.g., Carroll & Green, 1997, Subsection 2.8.3):

$$\mathbf{Q} = \frac{1}{J-1}\mathbf{S}, \tag{10}$$

where **S** is the sample scatter matrix (also called mean-corrected sums of squares and cross product [SSCP] matrix, Carroll & Green, 1997) given by:

$$\mathbf{S} = (\mathbf{D} - \bar{\mathbf{y}}\mathbf{1}_J')(\mathbf{D} - \bar{\mathbf{y}}\mathbf{1}_J')', \tag{11}$$

with **D** as defined in Step 1, $\mathbf{1}_J$ being a column vector of ones with length J, and $\bar{\mathbf{y}}$ is the sample mean column vector of length P:

$$\bar{\mathbf{y}} = [\bar{y}_{\bullet 1} \ldots \bar{\mathrm{y}}_{\bullet \mathrm{p}} \ldots \bar{\mathrm{y}}_{\bullet \mathrm{P}}]', \tag{12}$$

with $\bar{y}_{\bullet p} = \frac{1}{J}\sum_{j=1}^{J} y_{jp}$. As $\mathbf{y}_j$ are independently sampled from a multivariate normal distribution (see Equation 6), the sample scatter matrix[1] has a Wishart distribution (e.g., Pham-Gia & Choulakian, 2014):

$$\mathbf{S} \sim \mathcal{W}_P(\mathbf{\Sigma}, J-1). \tag{13}$$

We can directly translate this to JAGS syntax (note that the precision matrix **C** instead of the covariance matrix **Σ** needs to be used for `dwish()`:

```
# distributional assumption of
# sample scatter matrix (Equation 13)
S ~ dwish(C, J-1)

## construction of
## precision matrix C (Equation 8)
# main diagonal
for (p in 1:P){
    C[p,p] <- ((P-1)*sigma2.theta +
               sigma2.eps) / ksi
}
# lower/upper triangle
for (k in 2:P) {
    for (l in 1:(k-1)) {
        C[k,l] <- -sigma2.theta / ksi
```

[1] Instead of the sample scatter matrix, the sample covariance matrix can be used alternatively which is Wishart distributed with scale matrix $\mathbf{\Sigma}/(J-1)$ and $(J-1)$ degrees of freedom (Pham-Gia & Choulakian, 2014).

```
        C[l,k] <- C[k,l]
    }
}
# ksi (Equation 9)
ksi <- P * sigma2.theta * sigma2.eps +
       sigma2.eps^2
```

The prior distributions for σ^2_θ and σ^2_ε can be set as in the classic implementation in Step 1 (see above). The sample scatter matrix **S** needs to be computed from the sample data according to Equation 11 preferably before inputting it into JAGS. For instance, this can be easily done in R:

```
# sample mean column vector
# y.bar (Equation 12)
y.bar <- matrix(rowMeans(D), P, 1)
# column vector of ones
ones.vec <- matrix(1, J, 1)
# sample scatter matrix (Equation 11)
S <- (D - y.bar %*% t(ones.vec)) %*%
     t((D - y.bar %*% t(ones.vec)))
```

We call this version of the Bayesian multi-level null model the *nuisance-free covariance-based implementation*. This formulation contains only covariance-based terms (i.e., both error variances, σ^2_θ and σ^2_ε), but no mean-structure-based terms (i.e., the mean μ dropped out). If this mean is of interest, we can re-add it to the model.

Step 4 (optional): re-adding the mean structure

The conditional distribution of the sample means given the parameters μ, σ^2_θ, and σ^2_ε is (adapted from Flury, 1997, Chapter 4):

$$\bar{\mathbf{y}} \sim \mathcal{N}_P(\boldsymbol{\mu}_P, \frac{1}{J}\boldsymbol{\Sigma}). \tag{14}$$

This translates to JAGS code (note that `dmnorm()` needs a precision matrix instead of a covariance matrix as input):

```
  # precision matrix of means
  Cm <- J * C
  # distributional assumption of
  # sample means (Equation 14)
  y.bar ~ dmnorm(mu.vec, Cm)
  # construction of mu vector
  for (p in 1:P){
    mu.vec[p] <- mu
  }
```

The prior distribution for μ can be set as in the classic implementation in Step 1 (see above). We call this version of the Bayesian multi-level null model the *nuisance-free covariance- and mean-based implementation*.

Shortcut via structural equation modeling

Similar to the covariance structure, we can derive the model-implied means for the response variables y_{jp} via SEM. Based on Figure 1, we apply tracing rules and calculation rules for expectations (e.g., Steyer & Nagel, 2017, Chapter 6) and obtain:

$$\mathrm{E}(y_{jp}) = E(1 \cdot \theta_j + 1 \cdot \varepsilon_{jp}) = \mathrm{E}(\mu \cdot 1) + \mathrm{E}(\varepsilon_{jp}) = \mu.$$

Alternatively, we could use an SEM framework like LISREL or RAM (see Appendices C and D).

SIMULATION STUDY

One objective of this work is to show that the proposed approach leads to computational more efficient Bayesian model estimation. To this end, we conducted a simulation study in which we compared the run time between the described implementations. As run time gains are less expedient if they are at the expense of the parameter estimation quality, we include parameter recovery and precision statistics in our comparison.

Data generation

The data generating model was the multi-level null model described in Equations 1 and 2 with $\mu = 0$, $\sigma^2_\theta = 1$, $\sigma^2_\varepsilon = 1$, $J = 5{,}000$ persons (level 2 units), and $P = 20$ measurement occasions (level 1 units):

$$y_{jp} \sim \mathcal{N}(\theta_j, 1), \tag{15}$$

$$\theta_j \sim \mathcal{N}(0, 1). \tag{16}$$

The number of generated data sets (replications) was $N_{\text{repl}} = 1{,}000$.

Analysis

We ran the three Bayesian implementations of the multi-level null model described above: the classic implementation, the nuisance-free covariance-based implementation, and the nuisance-free covariance- and mean-based implementation. For all three implementations, the same $N_{\text{repl}} = 1{,}000$ generated data sets (replications) were used. All models were estimated with JAGS 4.3.0 (Plummer, 2017) interfaced by the R package `rjags` (Plummer, 2016) running on R version 3.5.0 (R Core Team, 2018). Starting values for the model parameters were random draws from the priors, that is, a normal distribution with a variance of 10,000 for μ, and an inverse gamma distribution for σ^2_θ and σ^2_ε with shape and rate parameter being 0.001. We used the following procedure to determine

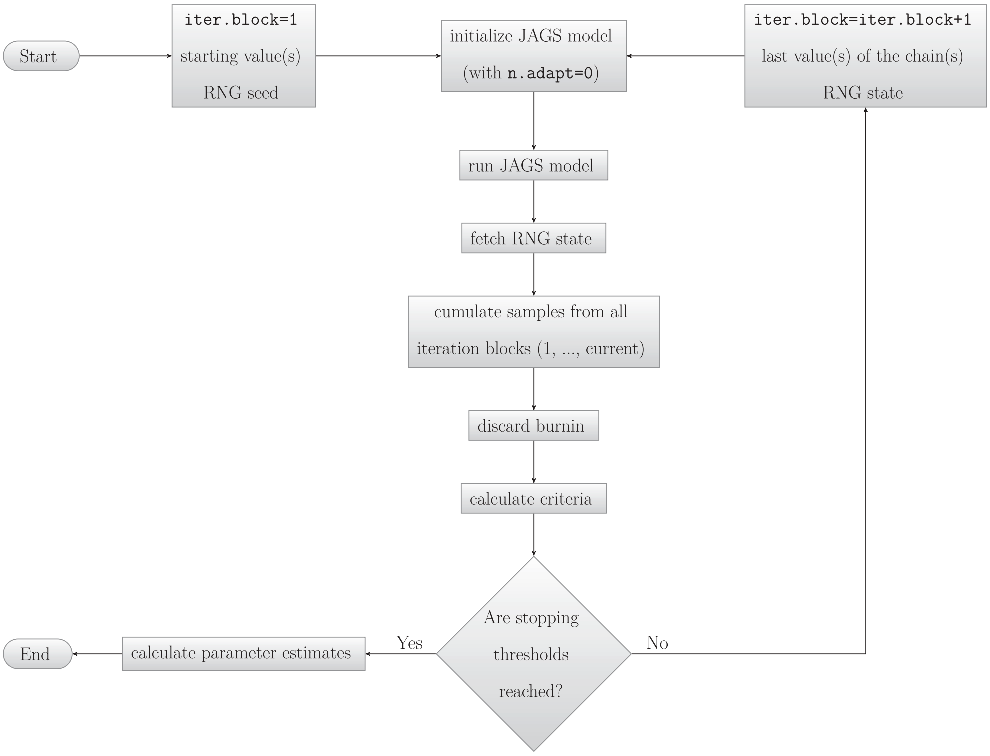

FIGURE 2 Procedure for monitoring criteria (such as convergence and parameter precision) in JAGS until their stopping thresholds are met. RNG = Random number generator.

convergence and ensure sufficient precision of Bayesian estimates (see Figure 2 for a flow chart of this procedure): 50 iterations on one chain were run; from those 50 iterations, the first 25% were discarded as burn-in; on those remaining 38 iterations, the potential scale reduction factor (PSR) and the effective sample size (ESS) were computed for all model parameters using the R package shinystan (Gabry, 2018). If the stopping criteria were not met, another 50 iterations were added and the resulting 100 iterations were processed again with the same procedure. This approach of adding iterations was repeated until the stopping criteria were reached. As stopping criteria we used PSR ≤ 1.001 and ESS ≥ 400 for all parameters in order to ensure that the Bayesian estimates are approximated well by summary statistics for the MCMC chain (see Zitzmann & Hecht, 2019). After that, the mode of the converged chain served as the parameter estimate. We used the mode because it can be considered a natural extension of the ML estimator (see DeCarlo, Kim, & Johnson, 2011) and it might outperform the mean and the median (e.g., Zitzmann, Lüdtke, & Robitzsch, 2015). As parameter recovery and precision statistics, bias, root mean squared error (RMSE), and coverage rate were computed. The coverage rate was calculated as the fraction of the number of replications in which the 95% credible interval covered the true parameter value and the number of replications (N_{repl}). All analyses were run on one Intel i7-5820K (3.30 GHz) CPU of a Windows 10 64-bit desktop computer.

Results

Table 1 shows the results and the run time for all three implementations. All recovery and precision statistics (bias, RMSE, coverage rate) are quite similar in all implementations. Bias is very low (<.001) and coverage rates are very good (between .934 and .951) for all parameters. Thus, all implementations are equally well suited for the estimation of the multi-level null model. Mean PSR is comparable across implementations. Mean ESS is roughly equal for both implementations that contain the same number of parameters (i.e., the classic and the nuisance-free covariance- and mean-based implementation). In the nuisance-free covariance-based implementation this statistic is somewhat lower. The number of iterations until the stopping criteria were met is lowest for this implementation as well. This is likely because the parameter μ is not included and therefore the conjunct stopping criteria are more easily met.

With respect to run time, the implementations differ considerably. Whereas the estimation with the classic implementation needs on average 86.5 s, the nuisance-free covariance-based implementation is roughly 47 times faster ($M = 1.8$ s) and the

TABLE 1.
Parameter Recovery Parameter, Precision and Run Time for Three Bayesian Implementations of the Multi-level Null Model

			nuisance-free	
		classic	cov.-based	cov./mean-based
Statistic	Parameter	Value	Value	Value
Bias	μ	−0.000003	—	−0.000287
	σ_θ^2	−0.000317	0.000076	−0.000033
	σ_ε^2	−0.000250	−0.000260	−0.000210
RMSE	μ	0.0155	—	0.0146
	σ_θ^2	0.0218	0.0209	0.0212
	σ_ε^2	0.0051	0.0050	0.0049
Coverage rate 95%	μ	.951	—	.945
	σ_θ^2	.951	.951	.948
	σ_ε^2	.934	.934	.934
Run time (seconds)	*M*	86.5	1.8	2.4
	min	49.3	1.3	1.5
	max	1962.3	4.1	10.5
PSR	*M*	0.9992	0.9989	0.9990
	min	0.9976	0.9976	0.9976
	max	1.0010	1.0010	1.0010
ESS	*M*	561	483	526
	min	400	400	400
	max	6794	1060	1746
Number of iterations	*M*	897	683	738
	min	550	550	550
	max	20250	1550	2600

Note. cov. = covariance, $N_{\mathrm{repl}} = 1{,}000$, PSR ≤ 1.001 and ESS ≥ 400 for all parameters, $J = 5{,}000$ level 2 units (persons), $P = 20$ level 1 units (measurement occasions), one chain ran on one Intel i7-5820K (3.30 GHz) CPU of a Windows 10 64-bit desktop computer.

nuisance-free covariance- and mean-based implementation is around 35 times faster ($M = 2.4$ s). Thus, integrating out nuisance parameters and reformulating the model leads to a massive reduction in run time.

EMPIRICAL EXAMPLE

We use publicly available data from the ‘Midlife in the United States (MIDUS 2): Daily Stress Project, 2004-2009’ (Ryff & Almeida, 2017), a longitudinal study of health and well-being, to illustrate our Bayesian nuisance-free covariance/mean-based implementation of the multi-level null model. For reference purposes, we as well report results obtained from frequentist estimation with the R package `lme4` (Bates, Mächler, Bolker, & Walker, 2019). The MIDUS 2 data set contains data from 2,022 persons assessed at 8 measurement occasions. For the purpose of providing an example, we pick the variable “symptom severity” (B2DSYMAV, 1 = very mild, 10 = very severe). There are no missing values; thus, all persons have data for all measurement occasions (giving a total of $8 \cdot 2{,}022 = 16{,}176$ observations).

The analyses were run on the same machine and with the same specifications and run parameters as described in the simulation study. Results are shown in Table 2. The Bayesian results only marginally differ from the frequentist results. However, the `lme4` estimation was somewhat faster than our Bayesian JAGS estimation (0.16 s vs. 0.69 s). The estimated mean symptom severity is $\mu = 2.61$ and persons differ in symptom severity with a variance of $\sigma_\theta^2 = 1.65$, whereas the within-person variance is $\sigma_\varepsilon^2 = 1.94$. Inference for these parameters can be drawn by investigating the Bayesian credibility intervals or the frequentist confidence intervals. For all model parameters, the respective intervals do not include zero, indicating significance.

DISCUSSION

Run time of Bayesian models might be unsatisfactorily high. In this illustration, we showed how to integrate out model parameters that are not of substantive interest and therefore considered as a nuisance. We reformulated the nuisance-free model based on covariances and means and

TABLE 2.
Results of the Multi-level Null Model for the Variable “Symptom Severity” from the MIDUS 2 Study

	Bayesian nuisance-free cov./mean-based					`lme4`		
Parameter	*Est.*	*LL95*	*UL95*	*PSR*	*ESS*	*Est.*	*LL95*	*UL95*
μ	2.605	2.541	2.656	0.998	525	2.599	2.540	2.659
σ_θ^2	1.649	1.525	1.766	0.999	525	1.646	1.533	1.766
σ_ε^2	1.938	1.892	1.982	0.998	525	1.936	1.891	1.981
Run time (sec)	0.687					0.156		

Note. cov. = covariance, $J = 2{,}022$ persons, $P = 8$ measurement occasions. *LL95* and *UL95* are the lower and upper limit of the credibility interval in the Bayesian model and profile likelihood confidence interval limits in the `lme4` model.

utilized the property of the sample scatter matrix being Wishart distributed with the model-implied covariance matrix as the scale matrix. This led to a considerable reduction in run time compared to a classic implementation that included the nuisance parameters.

Several issues and limitations of our work need to be taken into consideration: (1) To keep it simple for the purpose of illustration, we used a basic model and a balanced design. Future research should extend this approach to more complex models and designs. (2) The described mathematical derivations might become challenging in more complex models and might require advanced mathematical skills. However, especially for popular, frequently used models we think that run time reduction is worthwhile because computational time is always a limited resource associated with costs. And with the demonstrated shortcut via structural equation modeling the mathematically challenging derivations can be bypassed for a large variety of models. (3) Frequentist estimation might be faster than Bayesian estimation (as shown in our empirical example). However, Bayesian estimation provides several advantages (e.g., inclusion of previous knowledge, estimation of otherwise intractable models, stabilization of parameter estimates) that users might want to profit from. (4) Our simulation study consisted of only one simulation design factor, namely, the implementations. All other factors (e.g., the number of persons and the number of measurement occasions) that might have an effect on the dependent variable (run time) were kept constant. Thus, our findings cannot be generalized beyond the investigated condition. Nevertheless, we believe that our main argument is not limited to the condition and the model investigated. Although run time reduction might vary in extent, we believe it should generalize to a large number of conditions and models. We speculate that with an increasing ratio of level 2 to level 1 units the run time reduction gains of the presented nuisance-free implementations increase, because each additional level 2 unit implies additionally sampling $(P + 1)$ times from a normal distribution in the classic implementation, whereas the number of sampling instances in the nuisance-free implementations does not depend on the number of level 2 units (J). Thus, the presented nuisance-free model is especially advantageous for a high number of level 2 units and a low number of level 1 units, a situation oftentimes encountered in, for instance, longitudinal large-scale assessments. (5) In our illustration, we deemed the person parameters θ_j, which are the person-specific means across time in the multi-level null model, as not relevant and thus dispensable. Only parameters that are not of interest from a substantive point of view are suitable for being integrated out. However, even if person parameters are of interest, the proposed nuisance-free implementations can still be used and person parameters might then be estimated in a further step using individual score methods (see, e.g., Hardt, Hecht, Oud, & Voelkle, 2019; Hardt, Hecht, & Voelkle, in press). (6) JAGS was used because it is a popular and stable multi-purpose Bayesian software. The proposed approach should easily work with other Bayesian software as well. However, the run time gain depends on the efficiency and speed of the implemented sampler and thus might differ.

In conclusion, we have shown that the proposed approach of integrating out nuisance parameters and model reformulation is a worthwhile strategy for run time reduction in Bayesian estimation and hope that our tutorial will help researchers to speed up their Bayesian models in the future.

FUNDING

We acknowledge support by the Open Access Publication Fund of Humboldt-Universität zu Berlin.

ORCID

Martin Hecht http://orcid.org/0000-0002-5168-4911
Daniel Vogel http://orcid.org/0000-0002-6360-4077

REFERENCES

Basu, D. (1977). On the elimination of nuisance parameters. *Journal of the American Statistical Association*, *72*, 355–366. doi:10.1080/01621459.1977.10481002

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2019). lme4: Linear mixed-effects models using 'Eigen' and S4 (Version 1.1-21) [Computer software]. Retrieved from http://cran.r-project.org/package=lme4

Bernardo, J. M., & Smith, A. F. M. (2000). *Bayesian theory*. Chichester, UK: John Wiley & Sons.

Boker, S. M. (2019). The Reticular Action Model: A remarkably lasting achievement. In E. Ferrer, S. M. Boker, & K. J. Grimm (Eds.), *Longitudinal Multivariate Psychology*(pp. 126–142). New York, NY: Routledge.

Carroll, J. D., & Green, P. E. (1997). Vector and matrix operations for multivariate analysis. In J. D. Carroll & P. E. Green (Eds.), *Mathematical tools for applied multivariate analysis* (pp. 26–76). San Diego, CA: Academic Press. doi:10.1016/B978-012160954-2/50003-2

DeCarlo, L. T., Kim, Y., & Johnson, M. S. (2011). A hierarchical rater model for constructed responses, with a signal detection rater model. *Journal of Educational Measurement*, *48*, 333–356. doi:10.1111/j.1745-3984.2011.00143.x

Flury, B. (1997). *A first course in multivariate statistics*. New York, NY: Springer.

Gabry, J. (2018). shinystan: Interactive visual and numerical diagnostics and posterior analysis for Bayesian models (Version 2.5.0) [Computer software]. Retrieved from http://cran.r-project.org/package=shinystan

Hardt, K., Hecht, M., Oud, J. H. L., & Voelkle, M. C. (2019). Where have the persons gone? – An illustration of individual score methods in autoregressive panel models. *Structural Equation Modeling*, *26*, 310–323. doi:10.1080/10705511.2018.1517355

Hardt, K., Hecht, M., & Voelkle, M. C. (in press). Robustness of individual score methods against model misspecification in autoregressive panel models. *Structural Equation Modeling*. doi:10.1080/10705511.2019.1642755

Hecht, M., Hardt, K., Driver, C. C., & Voelkle, M. C. (2019). Bayesian continuous-time Rasch models. *Psychological Methods*, *24*, 516–537. doi:10.1037/met0000205

Hoff, P. D. (2009). *A first course in Bayesian statistical methods*. New York, NY: Springer. doi:10.1007/978-0-387-92407-6

Hox, J. J. (2010). *Multilevel analysis: Techniques and applications* (2nd ed.). New York, NY: Routledge.

Jöreskog, K. G., Olsson, U. H., & Wallentin, F. Y. (2016). *Multivariate analysis with LISREL*. Basel, Switzerland: Springer International Publishing.

Kline, R. B. (2016). *Principles and practice of structural equation modeling* (4th ed.). New York, NY: The Guilford Press.

Lüdtke, O., Robitzsch, A., & Wagner, J. (2018). More stable estimation of the STARTS model: A Bayesian approach using Markov Chain Monte Carlo techniques. *Psychological Methods*, *23*, 570–593. doi:10.1037/met0000155

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*, 234–251. doi:10.1111/j.2044-8317.1984.tb00802.x

Mehta, P. D., & Neale, M. C. (2005). People are variables too: Multilevel structural equations modeling. *Psychological Methods*, *10*, 259–284. doi:10.1037/1082-989X.10.3.259

Pham-Gia, T., & Choulakian, V. (2014). Distribution of the sample correlation matrix and applications. *Open Journal of Statistics*, *4*, 330–344. doi:10.4236/ojs.2014.45033

Plummer, M. (2016). rjags: Bayesian graphical models using MCMC. (Version 4-6) [Computer Software]. Retrieved from http://cran.r-project.org/package=rjags

Plummer, M. (2017). JAGS. (Version 4.3.0) [Computer Software]. Retrieved from https://sourceforge.net/projects/mcmc-jags/files

Raudenbush, S. W., & Bryk, A. S. (2002). *Hierarchical linear models – Applications and data analysis methods* (2nd ed.). Thousand Oaks, CA: SAGE.

R Core Team. (2018). R: A language and environment for statistical computing. (Version 3.5.0) [Computer Software]. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from http://www.R-project.org

Ryff, C. D., & Almeida, D. M. (2017). *Midlife in the United States (MIDUS 2): Daily Stress Project, 2004–2009* [Data file, documentation, and code book]. Ann Arbor, MI: Inter-university Consortium for Political and Social Research. doi:10.3886/ICPSR26841.v2

Steyer, R., & Nagel, W. (2017). *Probability and conditional expectation: Fundamentals for the empirical sciences*. Oxford, UK: John Wiley & Sons.

Van de Schoot, R., Winter, S. D., Ryan, O., Zondervan-Zwijnenburg, M., & Depaoli, S. (2017). A systematic review of Bayesian articles in psychology: The last 25 years. *Psychological Methods*, *22*, 217–239. doi:10.1037/met0000100

Wright, S. (1934). The method of path coefficients. *The Annals of Mathematical Statistics*, *5*, 161–215. doi:10.1214/aoms/1177732676

Zitzmann, S. (2018). A computationally more efficient and more accurate stepwise approach for correcting for sampling error and measurement error. *Multivariate Behavioral Research*, *53*, 612–632. doi:10.1080/00273171.2018.1469086

Zitzmann, S., & Hecht, M. (2019). Going beyond convergence in Bayesian estimation: Why precision matters too and how to assess it. *Structural Equation Modeling*, *26*, 646–661. doi:10.1080/10705511.2018.1545232

Zitzmann, S., Lüdtke, O., & Robitzsch, A. (2015). A Bayesian approach to more stable estimates of group-level effects in contextual studies. *Multivariate Behavioral Research*, *50*, 688–705. doi:10.1080/00273171.2015.1090899

APPENDIX A. SOLVING THE INTEGRAL TO DERIVE THE NUISANCE-FREE SOLUTION

We treat the hyperparameters μ, σ_ε^2, and σ_θ^2 as fixed. We have Equation 5:

$$\begin{aligned} p(\mathbf{y}_1, \ldots, \mathbf{y}_J | \mu, \sigma_\theta^2, \sigma_\varepsilon^2) &= \underset{\{(\theta_1,\ldots,\theta_J) \in \mathbf{R}^J\}}{\int \cdots \int} p(\mathbf{y}_1, \ldots, \mathbf{y}_J | \mu, \sigma_\theta^2, \sigma_\varepsilon^2, \theta_1, \ldots, \theta_J) \mathrm{d}\theta_1 \ldots \mathrm{d}\theta_J \\ &= \prod_{j=1}^{J} \left[\int_{\theta_j} p(\theta_j | \mu, \sigma_\theta^2) \prod_{p=1}^{P} p(y_{jp} | \theta_j, \sigma_\varepsilon^2) \mathrm{d}\theta_j \right]. \end{aligned}$$

Each integral within the product on the right-hand side is of the same form, and we drop the dependence on the index j. Thus

$$\begin{aligned} I &= \int_\theta p(\theta | \mu, \sigma_\theta^2) \prod_{p=1}^{P} p(y_p | \theta, \sigma_\varepsilon^2) \mathrm{d}\theta \\ &= \int_\theta \left(\frac{1}{\sqrt{2\pi}} \right)^{P+1} \frac{1}{\sigma_\theta \sigma_\varepsilon^P} \exp\left\{ -\frac{1}{2} \left[\frac{(\theta - \mu)^2}{\sigma_\theta^2} + \sum_{p=1}^{P} \frac{(y_p - \theta)^2}{\sigma_\varepsilon} \right] \right\} \mathrm{d}\theta \\ &= \int_\theta \left(\frac{1}{\sqrt{2\pi}} \right)^{P+1} \frac{1}{\sigma_\theta \sigma_\varepsilon^P} \exp\left\{ -\frac{1}{2} \left[\frac{(\theta - \eta^2 (P\bar{y}/\sigma_\varepsilon^2 + \mu/\sigma_\theta^2))^2}{\eta^2} \right] \right. \\ &\qquad \left. - \frac{1}{2} \left[\sum_{p=1}^{P} \frac{y_p^2}{\sigma_\varepsilon^2} + \frac{\mu^2}{\sigma_\theta^2} - \eta^2 \left(\frac{P\bar{y}}{\sigma_\varepsilon^2} + \frac{\mu}{\sigma_\theta^2} \right)^2 \right] \right\} \mathrm{d}\theta, \end{aligned}$$

$$\text{with } \bar{y} = \frac{1}{P} \sum_{p=1}^{P} y_p \text{ and } \eta^2 = \frac{1}{P/\sigma_\varepsilon^2 + 1/\sigma_\theta^2} = \frac{\sigma_\varepsilon^2 \sigma_\theta^2}{P\sigma_\theta^2 + \sigma_\varepsilon^2}.$$

Since the normal density integrates to 1, we have that

$$\int_\theta \exp\left\{ -\frac{1}{2} \left[\frac{(\theta - \eta^2 (P\bar{y}/\sigma_\varepsilon^2 + \mu/\sigma_\theta^2))^2}{\eta^2} \right] \right\} \mathrm{d}\theta = \sqrt{2\pi}\eta,$$

and hence

$$I = \left(\frac{1}{\sqrt{2\pi}} \right)^{P} \frac{\eta}{\sigma_\theta \sigma_\varepsilon^P} \exp\left\{ -\frac{1}{2} \left[\sum_{p=1}^{P} \frac{y_p^2}{\sigma_\varepsilon^2} + \frac{\mu^2}{\sigma_\theta^2} - \eta^2 \left(\frac{P\bar{y}}{\sigma_\varepsilon^2} + \frac{\mu}{\sigma_\theta^2} \right)^2 \right] \right\}.$$

This can be written in the form of the density of a P-variate normal distribution:

$$I = \left(\frac{1}{\sqrt{2\pi}} \right)^{P} \sqrt{\det(\mathbf{C})} \exp\left\{ -\frac{1}{2} (\mathbf{y} - \boldsymbol{\mu})' \mathbf{C} (\mathbf{y} - \boldsymbol{\mu}) \right\},$$

$$\text{where } \mathbf{y} = (y_1, \ldots, y_P)', \boldsymbol{\mu} = (\mu, \ldots, \mu)', \text{and}$$

$$\mathbf{C} = c_1 \mathbf{I}_P + c_2 \mathbf{1}_P \mathbf{1}'_P,$$

$$\text{with } c_1 = 1/\sigma_\varepsilon^2 \text{ and}$$

$$c_2 = -\frac{\eta^2}{\sigma_\varepsilon^4} = -\frac{\sigma_\theta^2}{P\sigma_\theta^2 \sigma_\varepsilon^2 + \sigma_\varepsilon^4}.$$

The matrix $\mathbf{C}$ has two distinct eigenvalues: $\lambda_1 = c_1 + Pc_2$ with multiplicity 1, corresponding to the eigenvector $\mathbf{1}_P$, and $\lambda_2 = c_1$ with multiplicity $P - 1$, the corresponding eigenspace being orthogonal to $\mathbf{1}_P$. Hence,

$$\sqrt{\det(\mathbf{C})} = \sqrt{(c_1 + Pc_2) c_1^{P-1}} = \frac{\eta}{\sigma_\theta \sigma_\varepsilon^P}.$$

This matrix $\mathbf{C}$ is the same that is given in Equation 8. Let $\boldsymbol{\Sigma}$ denote the inverse of $\mathbf{C}$. It remains to show that $\boldsymbol{\Sigma}$ has the form given in Equation 7. Towards this end, note that $\mathbf{C}^{-1}$ has eigenvalues $\delta_1 = \lambda_1^{-1} = \sigma_\varepsilon^2 + P\sigma_\theta^2$ (with multiplicity 1) and $\delta_2 = \lambda_2^{-1} = \sigma_\varepsilon^2$ (with multiplicity $P - 1$). Further, using the decomposition

$$\mathbf{A} = \sum_{p=1}^{P} a_p \mathbf{v}_p \mathbf{v}'_p$$

for any real symmetric square matrix $\mathbf{A}$ where $a_1, \ldots, a_P$ are the (generally non-distinct) eigenvalues and $\mathbf{v}_1, \ldots, \mathbf{v}_P$ form an orthonormal basis of corresponding eigenvectors, we obtain that $\boldsymbol{\Sigma}$ has the same structure as $\mathbf{C}$, that is,

$$\boldsymbol{\Sigma} = s_1 \mathbf{I}_P + s_2 \mathbf{1}_P \mathbf{1}'_P,$$

and by inverting the relationship between (c_1, c_2) and (λ_1, λ_2), we arrive at $s_1 = \sigma_\varepsilon^2$ and $s_2 = \sigma_\theta^2$. This completes the proof.

APPENDIX B. ANOTHER JUSTIFICATION OF THE DERIVED RESPONSE DISTRIBUTION

Here we present a way of deriving the results presented in Equations 5–7. All distributions in the following are conditional on μ, σ_θ^2, and σ_ε^2, that is, these parameters are treated as non-random. Then, we have from Equation 2

$$\theta_j \sim \mathcal{N}(\mu, \sigma_\theta^2), \qquad j = 1, \ldots, J.$$

Further, define

$$z_{jp} \sim \mathcal{N}(0, \sigma_\varepsilon^2), \qquad j = 1, \ldots, J,\ p = 1, \ldots, P,$$

with all variables being independent. Then, for each $j = 1, \ldots, J$,

$$\begin{pmatrix} z_{j1} \\ \vdots \\ z_{jP} \\ \theta_j \end{pmatrix} \sim \mathcal{N}_{P+1} \left(\begin{pmatrix} 0 \\ \vdots \\ 0 \\ \mu \end{pmatrix}, \begin{pmatrix} \sigma_\varepsilon^2 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \sigma_\varepsilon^2 & 0 \\ 0 & \cdots & 0 & \sigma_\theta^2 \end{pmatrix} \right),$$

and hence

$$\mathbf{y}_j = \begin{pmatrix} y_{j1} \\ \vdots \\ y_{jP} \end{pmatrix} \overset{\mathrm{d}}{=} \begin{pmatrix} 1 & 0 & \cdots & 0 & 1 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & 1 & 0 & 1 \\ 0 & \cdots & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} z_{j1} \\ \vdots \\ z_{jP} \\ \theta_j \end{pmatrix},$$

where $\overset{\mathrm{d}}{=}$ denotes equality in distribution. Hence, $\mathbf{y}_j$ has a P-variate normal distribution with mean vector $\boldsymbol{\mu}_P$ and covariance matrix

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0 & \cdots & 0 & 1 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & 1 & 0 & 1 \\ 0 & \cdots & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} \sigma_\varepsilon^2 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \sigma_\varepsilon^2 & 0 \\ 0 & \cdots & 0 & \sigma_\theta^2 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & 1 & 0 \\ 0 & \cdots & 0 & 1 \\ 1 & \cdots & 1 & 1 \end{pmatrix},$$

which can be seen to be identical to the form given in Equation 7.

APPENDIX C. LISREL MODEL FORMULATION

Here we present a way to obtain the model-implied covariance matrix and the model-implied means using the LISREL structural equation modeling framework (Jöreskog et al., 2016). The multi-level null model depicted in Figure 1 translates into the following LISREL matrices:

$$\boldsymbol{\Phi} = \left(\sigma_\theta^2\right), \boldsymbol{\Theta_\delta} = \begin{pmatrix} \sigma_\varepsilon^2 & & 0 \\ & \ddots & \\ 0 & & \sigma_\varepsilon^2 \end{pmatrix}, \boldsymbol{\Lambda_x} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}, \text{and}$$

$$\boldsymbol{\kappa} = (\mu).$$

The model-implied covariance matrix can then be calculated as:

$$\begin{aligned} \boldsymbol{\Sigma} &= \boldsymbol{\Lambda_x}\boldsymbol{\Phi}\boldsymbol{\Lambda_x'} + \boldsymbol{\Theta_\delta} \\ &= \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \left(\sigma_\theta^2\right)(1 \ldots 1) + \begin{pmatrix} \sigma_\varepsilon^2 & & 0 \\ & \ddots & \\ 0 & & \sigma_\varepsilon^2 \end{pmatrix} \\ &= \begin{pmatrix} \sigma_\theta^2 + \sigma_\varepsilon^2 & & \sigma_\theta^2 \\ & \ddots & \\ \sigma_\theta^2 & & \sigma_\theta^2 + \sigma_\varepsilon^2 \end{pmatrix}, \end{aligned}$$

which is identical to the matrix given in Equation 7.
The model-implied mean vector can be calculated as:

$$\boldsymbol{\mu}_P = \boldsymbol{\Lambda_x}\boldsymbol{\kappa} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} (\mu) = \begin{pmatrix} \mu \\ \vdots \\ \mu \end{pmatrix},$$

which is identical to the vector used in Equation 6.

APPENDIX D. RAM MODEL FORMULATION

Here we present a way to obtain the model-implied covariance matrix and the model-implied means using the RAM structural equation modeling framework (McArdle & McDonald, 1984; see also Boker, 2019). The multi-level null model depicted in Figure 1 translates into the following RAM matrices:

$$\mathbf{S} = \begin{pmatrix} \sigma_\theta^2 & & & 0 \\ & \sigma_\varepsilon^2 & & \\ & & \ddots & \\ 0 & & & \sigma_\varepsilon^2 \end{pmatrix}, \mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{F} = \begin{pmatrix} 0 & 1 & & 0 \\ \vdots & \ddots & \ddots & \\ 0 & \ldots & 0 & 1 \end{pmatrix}, \text{ and } \mathbf{M} = \begin{pmatrix} \mu \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

The model-implied covariance matrix can then be calculated as:

$$\begin{aligned} \boldsymbol{\Sigma} &= \mathbf{F}(\mathbf{I}_{P+1} - \mathbf{A})^{-1}\mathbf{S}(\mathbf{I}_{P+1} - \mathbf{A})^{-1'}\mathbf{F}' \\ &= \begin{pmatrix} \sigma_\theta^2 + \sigma_\varepsilon^2 & & \sigma_\theta^2 \\ & \ddots & \\ \sigma_\theta^2 & & \sigma_\theta^2 + \sigma_\varepsilon^2 \end{pmatrix}, \end{aligned}$$

which is identical to the matrix given in Equation 7.
The model-implied mean vector can be calculated as:

$$\boldsymbol{\mu}_P = \mathbf{F}(\mathbf{I}_{P+1} - \mathbf{A})^{-1}\mathbf{M} = \begin{pmatrix} \mu \\ \vdots \\ \mu \end{pmatrix},$$

which is identical to the vector used in Equation 6.