Multivariate Behavioral Research, 49:119–129, 2014
Copyright © Taylor & Francis Group, LLC
ISSN: 0027-3171 print / 1532-7906 online
DOI: 10.1080/00273171.2013.866537

A Computationally Efficient State Space Approach to Estimating Multilevel Regression Models and Multilevel Confirmatory Factor Models

Fei Gu
Department of Psychology, McGill University

Kristopher J. Preacher
Department of Psychology and Human Development, Vanderbilt University

Wei Wu
Department of Psychology, University of Kansas

Yiu-Fai Yung
SAS Institute Inc.

Although the state space approach for estimating multilevel regression models has been well established for decades in the time series literature, it does not receive much attention from educational and psychological researchers. In this article, we (a) introduce the state space approach for estimating multilevel regression models and (b) extend the state space approach for estimating multilevel factor models. A brief outline of the state space formulation is provided and then state space forms for univariate and multivariate multilevel regression models, and a multilevel confirmatory factor model, are illustrated. The utility of the state space approach is demonstrated with either a simulated or real example for each multilevel model. It is concluded that the results from the state space approach are essentially identical to those from specialized multilevel regression modeling and structural equation modeling software. More importantly, the state space approach offers researchers a computationally more efficient alternative to fit multilevel regression models with a large number of Level 1 units within each Level 2 unit or a large number of observations on each subject in a longitudinal study.

The estimation of multilevel regression models as single-level structural equation models was addressed comprehensively by Bauer (2003) and Curran (2003). Based on the equivalence between single-level structural equation models and state space models (SSMs; e.g., Chow, Ho, Hamaker, & Dolan, 2010), state space forms for multilevel regression models can be derived by direct translation of the corresponding single-level structural equation models. The single-level structural equation modeling (SEM) approach requires that the original long-format data set, in which responses from all Level 1 units are stored in a single vector, be transformed to wide format such that the Level 1 units nested within Level 2 units are listed as separate variables stored in the new data set (Bauer, 2003; Curran, 2003; Mehta & Neale, 2005). Conceptually, the single-level SEM treatment requires a possibly nonintuitive approach to model specification and a subtle change in the research design. That is, the observations in the new data set are considered Level 2 units as if the data are collected from a nonstratified (nonnested) design instead of a stratified (nested) design. Although useful in many contexts, we argue that the wide format renders the single-level SEM approach computationally inefficient. Moreover, the steps involved in fitting a multilevel model as a single-level structural equation model have been

Correspondence concerning this article should be addressed to Fei Gu, Department of Psychology, McGill University, 1205 Dr. Penfield Avenue, Montreal, Quebec H3A 1B1, Canada. E-mail: fei.gu@mcgill.ca

characterized as a "data management nightmare" (Curran, 2003, p. 565).

Herein, we first introduce the state space approach for estimating multilevel regression models. In fact, the discrete- and continuous-time state space approach for estimating univariate multilevel regression models already has been investigated by statisticians in the time series literature (e.g., Icaza & Jones, 1999; Jones, 1993). Moreover, Jones (1993) also discussed the state space formulation for multivariate multilevel regression models. Although the state space approach for estimating multilevel regression models has been well established for decades, it does not receive much attention from educational and psychological researchers. To the best of our knowledge, the state space approach is barely known and (almost) never implemented by multilevel regression modelers in education and psychology, except that Ho, Shumway, and Ombao (2006, pp. 153–154) briefly described the specification of a discrete-time state space model for a univariate multilevel regression model.

Second, we extend the state space approach to handle the task of estimating multilevel factor models. In this treatment, we focus only on the discrete-time state space model and provide a brief outline of the state space formulation. Then, state space forms are illustrated for univariate and multivariate multilevel regression models and a random intercepts confirmatory factor model. The utility of the state space approach is demonstrated with a simulated or real example for each multilevel model. It is concluded that the results from the state space approach are essentially identical to those from specialized multilevel regression modeling (e.g., MLwiN, SAS PROC MIXED) and SEM (e.g., Mplus) software packages.[1] Furthermore, the algorithm used in the state space approach is computationally more efficient than those specifically designed for multilevel regression models and the single-level SEM treatment discussed by Bauer (2003) and Curran (2003).

STATE SPACE FORMULATION

Typical applications of the state space model can be found in optimal control and signal analysis in engineering, where the method originated. Starting at about the same time, the state space model also made, and is still making, important contributions to time series analysis (e.g., Durbin & Koopman, 2001; Harvey, 1989). In both areas, the unit of analysis is almost always a single case from which the (multivariate) time series data are gathered and analyzed. Let y_i be a p-variate data vector representing p manifest variables, η_i a q-variate vector representing q latent state variables, x_i a g-variate vector containing g exogenous variables representing external inputs to the system, and the subscript[2] i ($i = 1, 2, \ldots, \mathrm{T}$) denotes the time point. The general linear Gaussian single-subject state space model includes the following two equations:

$$\begin{aligned} y_i &= \tau + \Lambda\eta_i + \Gamma x_i + \varepsilon_i,\ \varepsilon_i \sim MVN(0, \Theta), \\ \eta_i &= \alpha + \mathrm{B}\eta_{i-1} + \Upsilon x_i + \zeta_i,\ \zeta_i \sim MVN(0, \Psi), \end{aligned} \quad (1)$$

where τ is a $p \times 1$ vector containing intercepts, Λ is a $p \times q$ loading matrix, Γ is a $p \times g$ matrix containing regressive parameters, ε_i is a $p \times 1$ vector containing measurement errors, Θ is a $p \times p$ (usually diagonal) covariance matrix, α is a $q \times 1$ vector containing latent means, β is a $q \times q$ transition matrix, Υ is a $q \times g$ matrix also containing regressive parameters, ζ_i is a $q \times 1$ vector containing latent state noises, and Ψ is a $q \times q$ (unstructured) covariance matrix. The first equation, called the measurement equation, describes the relationship between the observed variables and latent state variables (also known as the latent factors in SEM parlance), and the second equation is called the transition equation, depicting the temporal lead-lag relationship between latent state variables at adjacent measurement occasions.

Parameters in the state space model are estimated using a maximum likelihood (ML) estimator through a recursive algorithm called the Kalman filter (KF; Kalman, 1960). Throughout the observations of the time series, the Kalman recursion implements the equations sequentially in Block 2:

$$\begin{aligned} \eta_{i|i-1} &= \alpha + \mathrm{B}\eta_{i-1|i-1} + \Upsilon x_i \\ \mathrm{P}_{i|i-1} &= \mathrm{B}\mathrm{P}_{i-1|i-1}\mathrm{B}' + \Psi \\ e_i &= y_i - y_{i|i-1} = y_i - (\tau + \Lambda\eta_{i|i-1} + \Gamma x_i) \\ \mathrm{D}_i &= \Lambda\mathrm{P}_{i|i-1}\Lambda' + \Theta \\ \mathrm{K}_i &= \mathrm{P}_{i|i-1}\Lambda'\mathrm{D}_i^{-1} \\ \eta_{i|i} &= \eta_{i|i-1} + \mathrm{K}_i e_i = \eta_{i|i-1} + \mathrm{P}_{i|i-1}\Lambda'\mathrm{D}_i^{-1}e_i \\ \mathrm{P}_{i|i} &= \mathrm{P}_{i|i-1} - \mathrm{K}_i\mathrm{D}_i\mathrm{K}_i' \\ &= \mathrm{P}_{i|i-1} - \mathrm{P}_{i|i-1}\Lambda'\mathrm{D}_i^{-1}\Lambda\mathrm{P}_{i|i-1}. \end{aligned} \quad (2)$$

At the first observation, the Kalman recursion is initialized with a latent state vector, $\eta_{0|0}$, and its associated covariance matrix, $\mathrm{P}_{0|0}$. In practice, when nothing is known about the latent state variables, $\eta_{0|0}$ is often set as a $q \times 1$ null vector, and $\mathrm{P}_{0|0}$ is set as a $q \times q$ diagonal matrix with large values, essentially infinite (e.g., 1E6), assigned to the diagonal elements (i.e., $\mathrm{P}_{0|0}$ being a diffuse matrix). Final parameter estimates are not sensitive to the different choices of $\mathrm{P}_{0|0}$ when the time series is sufficiently long (Durbin & Koopman, 2001). However, for the state space approach to estimating the multilevel models in this article, initialization of the KF does need proper specification of $\mathrm{P}_{0|0}$, which is discussed throughout

[1] Specifically, MLwiN 2.26, SAS 9.3, and Mplus 7 are used in this article.

[2] In the time series literature, t is always used as the subscript to index the time point, but we choose to use i to match the subscript in the multilevel models presented later.

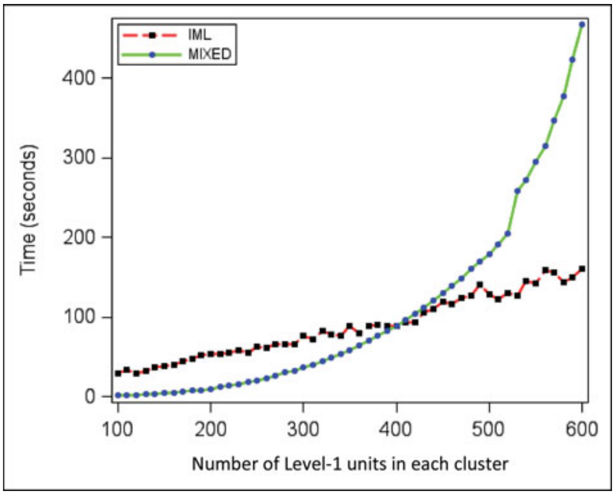

FIGURE 1 Computing time from PROC MIXED and PROC IML (Interactive Matrix Language) fitting random intercept and random slope models (color figure available online).

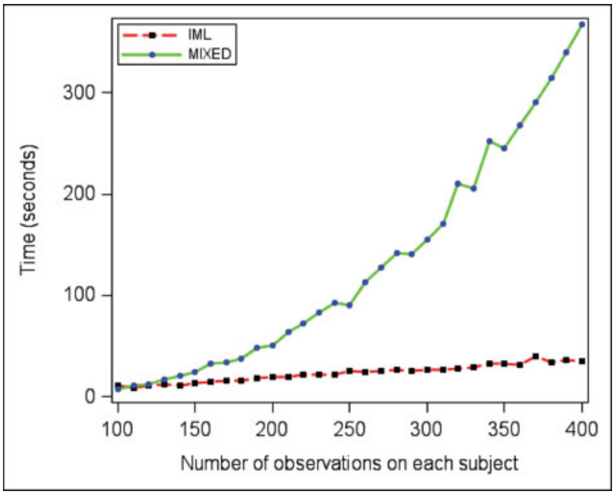

FIGURE 2 Computing time from PROC MIXED and PROC IML fitting random intercept longitudinal models (color figure available online).

the subsequent examples. Given a set of starting values for the parameters, the value of the log-density function at time i of the multivariate normal distribution can be calculated by inserting e_i and D_i. Finally, the log-likelihood function for the state space model, called *prediction error decomposition* (PED; Schweppe, 1965), is obtained by summing the log-density functions over i:

$$\mathrm{PED} = \frac{1}{2}\sum_{i=1}^{\mathrm{T}}\left[-p\log(2\pi) - \log|\mathrm{D}_i| - e_i'\mathrm{D}_i^{-1}e_i\right]. \qquad (3)$$

The solution for the parameter estimates (i.e., the ML estimates) is found where the PED function is maximized via any direct optimization technique (e.g., Newton-Raphson, quasi-Newton) or the Expectation-Maximization method. Throughout the equations of the Kalman recursion, only D_i needs to be inverted in the computation, and its dimension is $p \times p$. It is shown both analytically and empirically (Figure 1 and Figure 2) that the KF algorithm is more efficient than the algorithms specifically designed for multilevel regression models and those used in the single-level SEM approach (Bauer, 2003; Curran, 2003) for estimating multilevel regression models.

The state space approach introduced here utilizes the multiple-subject state space model, which is a direct extension of the single-subject state space model just described. In the multiple-subject state space model, a second subscript j ($j = 1, 2, \ldots,$ J) indexing different units (or clusters in the multilevel modeling context) is added to both the measurement and transition equations, that is,

$$\begin{aligned} y_{ij} &= \tau_{\cdot j} + \Lambda_{\cdot j}\eta_{ij} + \Gamma_{\cdot j}x_{ij} + \varepsilon_{ij},\ \ \varepsilon_{ij} \sim MVN(0, \Theta_{\cdot j}), \\ \eta_{ij} &= \alpha_{\cdot j} + \mathrm{B}_{\cdot j}\eta_{i-1,j} + \Upsilon_{\cdot j}x_{ij} \\ &\quad + \zeta_{ij}, \zeta_{ij} \sim MVN(0, \Psi_{\cdot j}). \end{aligned} \qquad (4)$$

Parameter estimation is implemented by the same Kalman recursion for each unit, but it is the total PED function that is maximized to provide the ML estimates in the multiple-subject state space model. More specifically, the Kalman recursion is initialized with $\eta_{0|0,j}$ and $\mathrm{P}_{0|0,j}$ for unit j, and PED_j is calculated. In other words, each individual can have a unique initial latent state and corresponding covariance matrix. The total PED function is obtained by summing PED_j over j. That is,

$$\begin{aligned} \text{total PED} = \frac{1}{2}\sum_{j=1}^{\mathrm{J}}\sum_{i=1}^{\mathrm{T}}\big[&- p_j\log(2\pi) \\ &- \log\left|\mathrm{D}_{ij}\right| - e_{ij}'\mathrm{D}_{ij}^{-1}e_{ij}\big]. \end{aligned} \qquad (5)$$

Finally, the solution for the parameter estimates is found where the total PED function is maximized with respect to the parameters from all units.

The idea of the multiple-subject extension for state space models is not different from that of the multiple-group framework for structural equation models, but it provides far more flexibility in that it subsumes many multilevel regression models. We illustrate the state space form for each multilevel model, followed by an example, in subsequent sections. We start with the state space approach to the simplest univariate multilevel regression model because it provides an easy stepping-stone for understanding the more complicated multivariate multilevel regression model.

UNIVARIATE MULTILEVEL REGRESSION MODELS

The Random Intercept and Random Slope Model

The univariate multilevel regression model with random intercept and random slopes is often expressed using separate

Level 1 and Level 2 scalar equations. The Level 1 equation is

$$y_{ij} = \beta_{0j} + \beta_{1j} x_{ij} + \varepsilon_{ij}, \tag{6}$$

where $\varepsilon_{ij} \sim N\left(0, \sigma_{\varepsilon}^{2}\right)$; the Level 2 equations are

$$\begin{aligned} \beta_{0j} &= \gamma_{00} + \gamma_{01} w_{1j} + u_{0j} \\ \beta_{1j} &= \gamma_{10} + \gamma_{11} w_{1j} + u_{1j}, \end{aligned} \tag{7}$$

where

$$\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} \sim MVN\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00} & \\ \tau_{10} & \tau_{11} \end{bmatrix} \right).$$

In this example, y_{ij} is the response variable for unit i within group j, x_{ij} is the Level 1 covariate, and w_{1j} is the Level 2 covariate. Taking the Level 1 and Level 2 equations together, the combined (reduced-form) equation is

$$\begin{aligned} y_{ij} = {} & \left[\gamma_{00} + \gamma_{01} w_{1j} + \gamma_{10} x_{ij} + \gamma_{11} w_{1j} x_{ij}\right] \\ & + \left[u_{0j} + u_{1j} x_{ij}\right] + \varepsilon_{ij}. \end{aligned} \tag{8}$$

Writing the model in the combined equation separates the fixed effects (the first bracket) from the random effects (the second bracket). In total, there are eight parameters in this model.

In order to estimate multilevel regression models as state space models, it is crucial to find corresponding state space forms. To do so, we first write the combined equation in its matrix form (Laird & Ware, 1982):[3]

$$\begin{aligned} \left[y_{ij}\right] = {} & \left[\, 1 \;\; w_{1j} \;\; x_{ij} \;\; w_{1j} x_{ij} \,\right] \begin{bmatrix} \gamma_{00} \\ \gamma_{01} \\ \gamma_{10} \\ \gamma_{11} \end{bmatrix} \\ & + \left[\, 1 \;\; x_{ij} \,\right] \begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} + \left[\varepsilon_{ij}\right]. \end{aligned} \tag{9}$$

Then, we rearrange the terms in Equation 9 so that the modified matrix form matches the measurement equation of the state space model:

$$\begin{aligned} \left[y_{ij}\right] = {} & \left[\, 1 \;\; x_{ij} \,\right] \begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} \\ & + \left[\, \gamma_{00} \;\; \gamma_{01} \;\; \gamma_{10} \;\; \gamma_{11} \,\right] \begin{bmatrix} 1 \\ w_{1j} \\ x_{ij} \\ w_{1j} x_{ij} \end{bmatrix} + \left[\varepsilon_{ij}\right]. \end{aligned} \tag{10}$$

Moreover, u_{0j} and u_{1j}, the random effects, are invariant across different Level 1 units within a given cluster in the multilevel regression model so that the transition equation within cluster j is

$$\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix}. \tag{11}$$

To obtain the state space form[4] for this univariate multilevel regression model, we specify the data vector (y_{ij}), latent state vector (η_{ij}), external input vector (x_{ij}), measurement error vector (ε_{ij}), latent noise vector (ζ_{ij}), and parameter matrices ($\tau_{\cdot j}$, $\alpha_{\cdot j}$, $\Lambda_{\cdot j}$, $\mathrm{B}_{\cdot j}$, $\Gamma_{\cdot j}$, $\Upsilon_{\cdot j}$, $\Theta_{\cdot j}$, and $\Psi_{\cdot j}$) of the multiple-subject SSM as

$$\begin{aligned} & y_{ij} = \left[y_{ij}\right], \eta_{ij} = \begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix}, x_{ij} = \begin{bmatrix} 1 \\ w_{1j} \\ x_{ij} \\ w_{1j} x_{ij} \end{bmatrix}, \\ & \varepsilon_{ij} = \left[\varepsilon_{ij}\right], \zeta_{ij} = [0], \tau_{\cdot j} = [0], \Lambda_{\cdot j} = \left[1 \; x_{ij}\right], \\ & \Gamma_{\cdot j} = [\gamma_{00} \; \gamma_{01} \; \gamma_{10} \; \gamma_{11}], \Theta_{\cdot j} = \left[\sigma_{\varepsilon}^{2}\right], \alpha_{\cdot j} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ & \mathrm{B}_{\cdot j} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \Upsilon_{\cdot j} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \text{ and } \Psi_{\cdot j} = \begin{bmatrix} 0 & \\ 0 & 0 \end{bmatrix}. \end{aligned} \tag{12}$$

As a general rule, the random effects are included in the latent state vector. Note that the fixed effect parameters are included in $\Gamma_{\cdot j}$ and the Level 1 variance parameter is included in $\Theta_{\cdot j}$, and other vectors and matrices are either fixed ($\tau_{\cdot j}$, $\alpha_{\cdot j}$, $\mathrm{B}_{\cdot j}$, $\Upsilon_{\cdot j}$, $\Psi_{\cdot j}$) or known ($\Lambda_{\cdot j}$) from the data. In addition, $\Gamma_{\cdot j}$ and $\Theta_{\cdot j}$, are invariant across j. Thus far, the Level 2 variance parameters have not been included in the state space form; the solution is that the Kalman recursion is initialized with

$$\eta_{0|0,j} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{ and } \mathrm{P}_{0|0,j} = \begin{bmatrix} \tau_{00} & \\ \tau_{10} & \tau_{11} \end{bmatrix} \tag{13}$$

for all j. The initialization condition is set up in this way so that the matrix $\mathrm{P}_{0|0,j}$ contains the variances and covariances among the Level 2 random effects. Now, the eight parameters of the univariate multilevel regression model can be estimated in the multiple-subject SSM by the KF algorithm.

There are two advantages of the state space approach for estimating multilevel regression models. The first advantage

[3] The Laird-Ware form states that for each Level 2 unit, i, $\mathbf{y}_i = \mathbf{X}_i \alpha + \mathbf{Z}_i \mathbf{b}_i + \mathbf{e}_i$, where $\mathbf{X}_i$ and $\mathbf{Z}_i$ are the design matrices linking α and $\mathbf{b}_i$ to $\mathbf{y}_i$. The $\mathbf{b}_i$ are independent of each other and of the $\mathbf{e}_i$. The normal distribution assumptions are $\mathbf{b}_i \sim N(\mathbf{0}, \mathbf{D})$ and $\mathbf{e}_i \sim N(\mathbf{0}, \mathbf{R}_i)$. Now the total covariance matrix for unit i is $\mathrm{var}(\mathbf{y}_i) = \mathbf{V}_i = \mathbf{Z}_i \mathbf{D} \mathbf{Z}_i{}' + \mathbf{R}_i$. Assuming that there are n_i Level 1 units in unit i, the log-likelihood for the entire sample is

$$\ell\ell = -\frac{1}{2} \sum_{i} \left(n_i \log(2\pi) + \log |\mathrm{V}_i| + (\mathrm{y}_i - \mathrm{X}_i \alpha)' \, \mathrm{V}_i^{-1} \, (\mathrm{y}_i - \mathrm{X}_i \alpha) \right),$$

where

$$\alpha = \left(\sum_{i} \mathrm{X}_i' \mathrm{V}_i^{-1} \mathrm{X}_i \right)^{-1} \left(\sum_{i} \mathrm{X}_i' \mathrm{V}_i^{-1} \mathrm{y}_i \right).$$

[4] Alternative state space forms exist for the same multilevel regression model. Given space limitations, we do not show other possible state space forms.

is that the state space approach is more computationally efficient. As we noted earlier, D_i (or D_{ij}) is the only matrix that needs to be inverted in the (multiple-subject) Kalman recursion. The dimension of D_i (or D_{ij}) corresponds to that of the data vector. In other words, the Kalman filter algorithm requires an amount of memory proportional to p^2 to store the related matrices, denoted as $O(p^2)$, and the related matrix inversion requires the number of operations to be of order p^3, denoted as $O(p^3)$. In this example, D_i (or D_{ij}) reduces to a scalar, rendering the computations very efficient. In contrast, the commonly used ML estimators (both residual/restricted ML and the full information ML [FIML, or equivalently the iterative generalized least squares; Goldstein, 1986] estimators), specifically designed for multilevel regression models, involve matrix operations at the cluster level, which is less computationally efficient. Particularly, the dimension of the covariance matrix to be inverted at the cluster level in these estimation methods is determined by the number of Level 1 units within each Level 2 unit, or the number of observations on each subject if a longitudinal model is considered. In other words, letting n_j represent the number of Level 1 units in cluster j, the usual estimators for multilevel regression models require $O(n_j^2)$ for matrix storage and $O(n_j^3)$ for the operations in cluster j in the related matrix inversion. When data are gathered from nested structures, it is almost always the case that p is much smaller than n_j such that when the cluster size reaches a few hundred or thousand, the usual estimators for multilevel regression models may require too much memory, CPU time, or both; or the required calculations may be infeasible using traditional multilevel regression software (Wang & Tobias, 2009).[5] The state space approach, on the other hand, demands less memory. The memory reduction may make an infeasible computational task become feasible.

The large matrix problem also exists in the single-level SEM approach for estimating multilevel regression models because the Level 1 units within each Level 2 unit are listed as different columns in the new (wide-formatted) data set prepared for the single-level SEM approach (Bauer, 2003; Curran, 2003). Consequently, the dimensions of the sample covariance matrix and the corresponding matrix for covariance structures are also determined by the number of Level 1 units within each Level 2 unit. As the ML or FIML discrepancy function for SEM (see, e.g., Bollen, 1989) involves the inversion of the covariance structure matrix, the large matrix problem is clear.

TABLE 1
Results of the Random Intercept and Random Slope Model

		MLwiN		PROC MIXED		SSM-IML	
Parameter	True Value	Point	*SE*	Point	*SE*	Point	*SE*
γ_{00}	10	10.144	.704	10.146	.704	10.146	.704
γ_{01}	8	7.679	.866	7.679	.866	7.679	.866
γ_{10}	5	5.224	.611	5.226	.611	5.226	.611
γ_{11}	4	3.675	.752	3.674	.752	3.674	.752
σ_ε^2	1	1.063	.049	1.063	.049	1.063	.049
τ_{00}	10	8.182	1.689	8.187	1.678	8.188	1.678
τ_{10}	4	2.796	1.106	2.800	1.104	2.800	1.104
τ_{11}	8	6.341	1.266	6.339	1.269	6.339	1.270
–2*loglik		3636.527		3636.530		3636.530	

The second advantage (related to the first advantage) of the state space approach is that, compared with the single-level SEM approach for estimating multilevel regression models, the state space form is invariant for both balanced and unbalanced designs. No additional data management or modeling effort is required.

Based on the state space form for this random intercept and random slope model, it is straightforward to obtain state space forms for more complex models. In general, when there are two or more Level 1 predictor variables carrying additional fixed (random) effects in the multilevel regression model, the external input (latent state) vector will be extended, as will other related parameter matrices in the multiple-subject SSM.

Example 1. A data set is simulated from the random intercept and random slope model in Equation 8. There are 50 Level 2 units, and the number of Level 1 units within Level 2 units is unbalanced, ranging from 12 to 30 (these numbers are simulated from a Poisson distribution with the parameter equal to 20, i.e., Poisson(20)). We fit the random intercept and random slope model to the simulated data using the state space approach, PROC MIXED, and MLwiN. The state space approach is implemented by code written in PROC IML (IML stands for Interactive Matrix Language, a matrix-oriented programming language), and point estimates of all parameters are obtained by minimizing the negative total PED function via the quasi-Newton method using the NLPQN subroutine in PROC IML; the corresponding standard errors are then obtained by taking the square roots of the diagonal elements of the numerical Hessian matrix of the negative total PED function evaluated at the point estimates. Note that the numerical Hessian matrix is calculated by the finite-differences method using the NLPFDD subroutine in PROC IML. The true values and the estimated values are summarized in Table 1. It is clear that the results from PROC IML are essentially identical to those from PROC MIXED and MLwiN.

[5] Particularly, PROC MIXED may reach a memory limitation, which is set by SAS, rendering the computational task infeasible. Currently, PROC MIXED is limited to 16 GB memory in the latest release of SAS (9.4, October of 2013) in the 64-bit Windows operation system regardless of how much physical memory the machine has. If the physical memory is larger than 16 GB, the user is obliged to use the UNIX operating system in order to take advantage of all available memory.

In addition to the numeric results, we also compare the computing time between PROC MIXED and PROC IML. As we indicated earlier, the algorithms specifically designed for multilevel regression models involve matrix computations at the cluster level. Particularly, if there are n_j Level 1 units in cluster j, the dimension of the covariance matrix in this cluster is $n_j \times n_j$. It is expected that, as the number of Level 1 units in each cluster increases, the computing time from PROC MIXED will increase faster than that from PROC IML. Herein, we simulated more data sets with 50 clusters from balanced designs, and the number of Level 1 units in each cluster varies from 100 to 600. Figure 1 shows the computing time from PROC MIXED and PROC IML fitting the same model to the same data set. The figure shows that PROC MIXED and PROC IML used about the same amount of time when there are 400 Level 1 units in each cluster, but the key point here is not the comparison of the real computing time because the SSM method implemented with PROC IML was not compiled at the same machine code level as that of PROC MIXED. Instead, attention should be paid to the pattern of the computing time in relation to the number of Level 1 units in each cluster. As we expected, the computing time from PROC MIXED increased in a quadratic manner, whereas the computing time from PROC IML increased in a linear manner. Furthermore, because PROC IML is a general-purpose high-level interactive matrix language, its underlying code for computations is not optimized for either memory storage or matrix operations. Presumably, if the state space approach is implemented by languages (e.g., C, Fortran) with optimized storage and matrix operations, in other words, at the same machine code level as PROC MIXED, the computing time will be much reduced.

The Random Intercept First-Order Autoregressive Model

When data are collected from longitudinal designs in which multiple units are measured at multiple occasions, it is often of interest to consider change over time in the repeated measures. Whereas different error structures are available to account for the effect of repeated measures, we consider a random intercept model with an error component following a first-order autoregressive process, denoted AR(1). The following model is also considered in Icaza and Jones (1999, p. 547):

$$\begin{aligned} y_{ij} &= \mu + \gamma_j + \varepsilon_{ij} + \nu_{ij}, \\ \text{where} \quad & \gamma_j \sim N(0, \tau), \nu_{ij} \sim N\left(0, \sigma_\nu^2\right), \\ \varepsilon_{ij} &= \phi\varepsilon_{i-1,j} + u_{ij}, \\ \text{where} \quad & u_{ij} \sim N\left(0, \sigma_u^2\right), \end{aligned} \tag{14}$$

TABLE 2a
Results of the Random Intercept Longitudinal Model

		PROC MIXED		SSM-IML	
Parameter	True Value	Point	*SE*	Point	*SE*
ϕ	0.5	.451	.066	.459	.066
μ	1	.969	.112	.971	.112
σ_u^2	0.75	.695	.137	.684	.135
σ_ν^2	0.25	.257	.116	.270	.113
τ	1	1.139	.178	1.143	.179
–2*loglik		5904.340		5904.718	

where μ is the grand mean, γ_j is the difference between the mean of group j and the grand mean, ε_{ij} is the error component, ν_{ij} is the residual, ϕ is the autoregressive parameter, and u_{ij} is the innovation. Note that the observations in the longitudinal model are not exchangeable because of the temporal lead-lag relationship. The state space form of this longitudinal model is obtained by defining

$$\begin{aligned} & y_{ij} = \left[y_{ij}\right], \eta_{ij} = \begin{bmatrix} \varepsilon_{ij} \\ \gamma_j \end{bmatrix}, x_{ij} = [1], \varepsilon_{ij} = \left[\nu_{ij}\right], \\ & \zeta_{ij} = \begin{bmatrix} u_{ij} \\ 0 \end{bmatrix}, \tau_{\cdot j} = [0], \Lambda_{\cdot j} = \begin{bmatrix} 1 & 1 \end{bmatrix}, \Gamma_{\cdot j} = [\mu], \\ & \Theta_{\cdot j} = \left[\sigma_\nu^2\right], \alpha_{\cdot j} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \mathrm{B}_{\cdot j} = \begin{bmatrix} \phi & 0 \\ 0 & 1 \end{bmatrix}, \\ & \Upsilon_{\cdot j} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \text{ and } \Psi_{\cdot j} = \begin{bmatrix} \sigma_u^2 & \\ 0 & 0 \end{bmatrix}, \end{aligned} \tag{15}$$

and the Kalman recursion is initialized with

$$\eta_{0|0,j} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{ and } \mathrm{P}_{0|0,j} = \begin{bmatrix} \frac{\sigma_u^2}{1-\phi^2} & \\ 0 & \tau \end{bmatrix} \tag{16}$$

for all j. The first diagonal element in $\mathrm{P}_{0|0,j}$ is obtained by the following derivation:

$$\begin{aligned} \mathrm{Var}(\varepsilon_{ij}) &= \phi^2 \mathrm{Var}(\varepsilon_{i-1,j}) + \mathrm{Var}(u_{ij}) \\ \Rightarrow \mathrm{Var}(\varepsilon_{ij}) &= \mathrm{Var}(\varepsilon_{i-1,j}) = \frac{\sigma_u^2}{1-\phi^2}. \end{aligned}$$

State space forms for more complicated longitudinal models can be easily derived; we refer readers to Jones (1993).

Example 2. A data set is simulated from the random intercept longitudinal model in Block 14. It contains 100 subjects with 20 observations on each subject. The true values and the estimated values are presented in Table 2a. The results from PROC MIXED[6] are almost identical to those from PROC IML. But the –2*log-likelihood value produced by PROC MIXED is slightly different from that produced by PROC IML. Thus, we further conduct

[6] The TYPE = AR(1) option can be used in the REPEATED statement to specify the AR(1) structure for the error component.

TABLE 2b
Results of the Monte Carlo Simulation

		PROC MIXED[a]			SSM-IML[a]			SSM-IML[b]		
Parameter	True Value	*M*	*SD*	Avg. *SE*	*M*	*SD*	Avg. *SE*	*M*	*SD*	Avg. *SE*
ϕ	0.5	.493	.054	.060	.495	.052	.059	.491	.056	.059
μ	1	1.001	.107	.106	1.001	.107	.106	1.001	.107	.106
σ_u^2	0.75	.749	.118	.132	.761	.114	.130	.772	.125	.133
σ_v^2	0.25	.239	.097	.110	.235	.093	.106	.225	.102	.109
τ	1	.993	.165	.162	.992	.165	.162	.992	.165	.162

[a]Results are based on 957 replications (i.e., 43 nonconvergent replications encountered in PROC MIXED are removed). [b]Results are based on 1,000 replications.

a small Monte Carlo simulation, generating 1,000 random samples using the same true values and the same model. Table 2b shows the mean and the standard deviation of the point estimates and the averaged standard errors. We conclude that both PROCs produce the same results.

Because the large matrix problem also exists for longitudinal models, we also compare the computing time between PROC MIXED and PROC IML for this random intercept longitudinal model. Similarly, the dimension of the covariance matrix at the cluster level depends on the number of observations on each subject. For example, because each subject has 20 observations, the dimension of the covariance matrix at the cluster level is 20 × 20. It is expected that, as the number of observations on each subject is increased, the computing time from PROC MIXED will increase faster than that from PROC IML. Figure 2 shows the computing time from PROC MIXED and PROC IML fitting the same model to the same data sets (the number of subjects is reduced to 50 in these data sets so that the computing tasks are parallel to those in Figure 1), and the number of observations on each subject varies from 100 to 400. Similar to the pattern in Figure 1, the computing time from PROC MIXED increased in a quadratic manner, whereas the computing time from PROC IML increased in a linear manner. It is worth noting that the rate of change of the computing time from PROC MIXED in Figure 2 is larger than that in Figure 1. Particularly, at the cluster size of 400, PROC MIXED uses more than 300 s to estimate the random intercept longitudinal model (Figure 2), whereas it uses only 100 s to estimate the random intercept and random slope model (Figure 1). It is obvious that, for PROC MIXED, the random intercept longitudinal model in Example 2 is more computationally expensive than the random intercept and random slope model in Example 1 even though matrix storage and operations are parallel in both scenarios. In contrast, the story for PROC IML is reversed. That is, with 400 observations on each subject, PROC IML takes less than 50 s to estimate the random intercept longitudinal model (Figure 2), whereas it uses about 100 s to estimate the random intercept and random slope model (Figure 1).

MULTIVARIATE MULTILEVEL REGRESSION MODEL

The Random Intercepts Regression Model

The multivariate multilevel regression model is a straightforward extension of the univariate models discussed in previous sections. We consider a three-variate random intercepts multilevel regression model with an unrestricted covariance matrix for the measurement error vector. This model is also considered in Kamata, Bauer, and Miyazaki (2008, Equations 10.8a–10.8g). The model formulation we present here is somewhat different from that used in Kamata et al. Specifically, the model formulation in Kamata et al. requires users to create a new data set in the first step, in which a new response variable is created by stacking the three response variables within each Level 1 unit, and dummy variables are added to associate the observations of the three response variables with those of the new response variable. Then, the model was formulated as a univariate model by separate equations at three levels. In other words, this model formulation escalates a two-level multivariate model to a three-level univariate model (see Goldstein, 1995; Hox, 2002; Snijders & Bosker, 2012). One limitation of this formulation is that it imposes a highly restrictive structure on the between-group covariance matrix.[7] The model formulation we adopt here treats the three response variables as separate outcome variables in a two-level multivariate model. Instead of writing the Level 1 and Level 2 equations separately, we provide the matrix form of the combined equations for this model:

$$\begin{bmatrix} y1_{ij} \\ y2_{ij} \\ y3_{ij} \end{bmatrix} = \begin{bmatrix} \gamma 1_{00} \\ \gamma 2_{00} \\ \gamma 3_{00} \end{bmatrix} + \begin{bmatrix} u1_{0j} \\ u2_{0j} \\ u3_{0j} \end{bmatrix} + \begin{bmatrix} \varepsilon 1_{ij} \\ \varepsilon 2_{ij} \\ \varepsilon 3_{ij} \end{bmatrix},$$

where

$$\begin{bmatrix} \varepsilon 1_{ij} \\ \varepsilon 2_{ij} \\ \varepsilon 3_{ij} \end{bmatrix} \sim MVN\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \delta_{11} & & \\ \delta_{21} & \delta_{22} & \\ \delta_{31} & \delta_{32} & \delta_{33} \end{bmatrix} \right). \quad (17)$$

[7] The between-group covariance matrix (see Equation 18) has to be a singular matrix due to this limitation.

The random effect (multivariate random intercept) vector is

$$\begin{bmatrix} u1_{0j} \\ u2_{0j} \\ u3_{0j} \end{bmatrix} \sim MVN\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau & & \\ \tau & \tau & \\ \tau & \tau & \tau \end{bmatrix}\right). \tag{18}$$

Note that there is only 1 unique parameter in the covariance matrix, resulting in a singular matrix; this specification is chosen purposefully to correspond with the three-level univariate model in Kamata et al. (2008, Equations 10.8a–10.8g). In total, there are 10 parameters in this model.

In order to obtain the state space form for this model, the matrix form of the combined equations is rewritten as

$$\begin{bmatrix} y1_{ij} \\ y2_{ij} \\ y3_{ij} \end{bmatrix} = \begin{bmatrix} 1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \end{bmatrix} \begin{bmatrix} u1_{0j} \\ u2_{0j} \\ u3_{0j} \end{bmatrix} + \begin{bmatrix} \gamma 1_{00} \\ \gamma 2_{00} \\ \gamma 3_{00} \end{bmatrix} [1] + \begin{bmatrix} \varepsilon 1_{ij} \\ \varepsilon 2_{ij} \\ \varepsilon 3_{ij} \end{bmatrix}. \tag{19}$$

It becomes clear that the state space form for the three-variate multilevel regression model can be obtained by specifying

$$y_{ij} = \begin{bmatrix} y1_{ij} \\ y2_{ij} \\ y3_{ij} \end{bmatrix}, \eta_{ij} = \begin{bmatrix} u1_{0j} \\ u2_{0j} \\ u3_{0j} \end{bmatrix}, x_{ij} = [1], \varepsilon_{ij} = \begin{bmatrix} \varepsilon 1_{ij} \\ \varepsilon 2_{ij} \\ \varepsilon 3_{ij} \end{bmatrix},$$

$$\zeta_{ij} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tau_{\cdot j} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \Lambda_{\cdot j} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\Gamma_{\cdot j} = \begin{bmatrix} \gamma 1_{00} \\ \gamma 2_{00} \\ \gamma 3_{00} \end{bmatrix}, \Theta_{\cdot j} = \begin{bmatrix} \delta_{11} & & \\ \delta_{21} & \delta_{22} & \\ \delta_{31} & \delta_{32} & \delta_{33} \end{bmatrix}, \alpha_{\cdot j} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

$$\mathrm{B}_{\cdot j} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \Upsilon_{\cdot j} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \text{ and } \Psi_{\cdot j} = \begin{bmatrix} 0 & & \\ 0 & 0 & \\ 0 & 0 & 0 \end{bmatrix}. \tag{20}$$

Finally, the Kalman recursion is initialized with

$$\eta_{0|0,j} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{ and } \mathrm{P}_{0|0,j} = \begin{bmatrix} \tau & & \\ \tau & \tau & \\ \tau & \tau & \tau \end{bmatrix} \tag{21}$$

for all j. Note that the dimension of D_i (or D_{ij}) in this model is 3 × 3 because there are three response variables.

Example 3. The model in Blocks 17 and 18 is fit to the data set[8] provided by Kamata et al. (2008) using MLwiN, Mplus, PROC MIXED, and PROC IML. Table 3 shows the results. Again, the results are essentially identical across different programs.

In terms of computational efficiency, the state space approach adds up the PED function at each observation, and the matrices involved are of small dimension (i.e., 3 × 3). On the other hand, the multilevel regression algorithms based on the three-level univariate approach (Goldstein, 1995; Hox, 2002; Snijders & Bosker, 2012) require inverting large matrices of dimension $3n_j \times 3n_j$. Analytically, it is obvious that the state space approach is more efficient.

TABLE 3
Results of the Random Intercept Multivariate Model

	MLwiN		Mplus[a]		PROC MIXED		SSM-IML	
Parameter	Point	*SE*	Point	*SE*	Point	*SE*	Point	*SE*
$\gamma 1_{00}$	3.677	.072	3.677	.072	3.677	.072	3.677	.072
$\gamma 2_{00}$	3.717	.072	3.717	.072	3.717	.072	3.717	.072
$\gamma 3_{00}$	3.721	.072	3.721	.072	3.721	.072	3.721	.072
τ	.130	.038	.129	.037	.130	.037	.130	.037
δ_{11}	3.036	.070	3.036	.071	3.036	.071	3.036	.071
δ_{21}	1.206	.054	1.206	.054	1.206	.054	1.206	.054
δ_{22}	3.096	.072	3.095	.072	3.096	.072	3.096	.072
δ_{31}	1.328	.054	1.328	.054	1.328	.054	1.328	.054
δ_{32}	1.290	.054	1.290	.054	1.290	.054	1.290	.054
δ_{33}	2.966	.069	2.966	.069	2.966	.069	2.966	.069
−2*loglik	42658.679		42657.388		42658.678		42658.678	

[a] Mplus encountered difficulty estimating a between-group covariance matrix with all equal elements, so we instead fit a one-factor model in the between-group factor model with unit loadings and no residual variances. This implies a covariance structure with all equal elements.

MULTILEVEL CONFIRMATORY FACTOR MODEL

As a didactic illustration, we show two examples in this section. The first example contains an unstructured covariance matrix for the random effect vector (i.e., an unstructured between-group covariance matrix). The covariance matrices for the measurement error vector and the random effect vector, representing the within- and between-group covariance matrices in the multilevel confirmatory factor model, are directly estimated. This model is a direct extension of the multivariate random intercepts regression model in the previous section. The second example is a random intercepts confirmatory factor model of which both the within- and between-group covariance matrices are reparameterized as functions of some basic set of parameters from the within- and between-group factor models (e.g., the factor loading matrices and the covariance matrices for the factors and residuals).

Direct Estimation of the Within- and Between-Group Covariance Matrices

The multilevel confirmatory factor model considered in this section is extended from the multivariate multilevel regression model in the previous section. We repeat the matrix form (Equation 17) of the three-variate multilevel regression

[8] It can be retrieved from http://www.infoagepub.com/products/content/files/serlinfiles_multilevel/Chapter%2010/Mplus/

model,

$$\begin{bmatrix} y1_{ij} \\ y2_{ij} \\ y3_{ij} \end{bmatrix} = \begin{bmatrix} \gamma1_{00} \\ \gamma2_{00} \\ \gamma3_{00} \end{bmatrix} + \begin{bmatrix} u1_{0j} \\ u2_{0j} \\ u3_{0j} \end{bmatrix} + \begin{bmatrix} \varepsilon1_{ij} \\ \varepsilon2_{ij} \\ \varepsilon3_{ij} \end{bmatrix},$$

where

$$\begin{bmatrix} \varepsilon1_{ij} \\ \varepsilon2_{ij} \\ \varepsilon3_{ij} \end{bmatrix} \sim MVN\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \delta_{11} & & \\ \delta_{21} & \delta_{22} & \\ \delta_{31} & \delta_{32} & \delta_{33} \end{bmatrix} \right).$$

But the random effect vector is unrestricted:

$$\begin{bmatrix} u1_{0j} \\ u2_{0j} \\ u3_{0j} \end{bmatrix} \sim MVN\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{11} & & \\ \tau_{21} & \tau_{22} & \\ \tau_{31} & \tau_{32} & \tau_{33} \end{bmatrix} \right). \qquad (22)$$

Therefore, the state space form for this model is the same as that in Block 20, but the Kalman recursion is instead initialized with

$$\eta_{0|0,j} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{ and } \mathrm{P}_{0|0,j} = \begin{bmatrix} \tau_{11} & & \\ \tau_{21} & \tau_{22} & \\ \tau_{31} & \tau_{32} & \tau_{33} \end{bmatrix} \qquad (23)$$

for all j. In total, there are 15 parameters in this model.

Note that the covariance matrices for the measurement error vector and the random effect vector are, in fact, the within- and between-group covariance matrices in the multilevel confirmatory factor model that we discuss in the subsection immediately after Example 4. This formulation does not involve any basic parameter of a factor model (e.g., factor loading parameters, variances and covariances of the factors and residuals), but it is helpful for readers to build a conceptual bridge between the multilevel regression model and the multilevel confirmatory factor model.

Example 4. The same data set from Example 3 is used here and is used in Example 5. This model is still estimable in the multilevel regression framework, so we continue to provide the results from MLwiN and PROC MIXED in addition to those from Mplus and PROC IML. Table 4 shows the results of directly estimating the within- and between-group covariance matrices, and they are essentially identical across different programs.

TABLE 4
Results From Direct Estimation of the Within- and Between-Group Covariance Matrices

	MLwiN		Mplus		PROC MIXED		SSM-IML	
Parameter	Point	*SE*	Point	*SE*	Point	*SE*	Point	*SE*
$\gamma1_{00}$	3.677	.073	3.677	.073	3.677	.073	3.677	.073
$\gamma2_{00}$	3.731	.082	3.731	.082	3.731	.082	3.731	.082
$\gamma3_{00}$	3.706	.080	3.706	.080	3.706	.080	3.706	.080
τ_{11}	.132	.041	.132	.041	.132	.041	.132	.041
τ_{21}	.111	.039	.111	.039	.111	.039	.111	.039
τ_{22}	.173	.052	.173	.051	.173	.051	.173	.051
τ_{31}	.116	.040	.116	.039	.116	.039	.116	.039
τ_{32}	.121	.043	.121	.043	.121	.043	.121	.043
τ_{33}	.164	.049	.164	.049	.164	.049	.164	.049
δ_{11}	3.017	.070	3.017	.070	3.017	.070	3.017	.070
δ_{21}	1.220	.054	1.220	.054	1.220	.054	1.220	.054
δ_{22}	3.062	.071	3.062	.071	3.062	.071	3.062	.071
δ_{31}	1.336	.054	1.336	.054	1.336	.054	1.336	.054
δ_{32}	1.306	.054	1.306	.054	1.306	.054	1.306	.054
δ_{33}	2.940	.068	2.940	.068	2.940	.068	2.940	.068
–2*loglik	42566.156		42566.156		42566.156		42566.156	

The Random Intercepts Confirmatory Factor Model

We now turn to a random intercepts confirmatory factor model, which is derived by decomposing both the within- and between-group covariance matrices (see Equation 17 and Equation 22). Specifically, instead of estimating the within- and between-group covariance matrices directly, the two covariance matrices are decomposed by basic parameters from the within- and between-group factor models. That is,

$$\begin{aligned} \mathrm{cov}\left(\begin{bmatrix} \varepsilon1_{ij} \\ \varepsilon2_{ij} \\ \varepsilon3_{ij} \end{bmatrix} \right) &= \Sigma_W = \Theta_{\cdot j} = \begin{bmatrix} \delta_{11} & & \\ \delta_{21} & \delta_{22} & \\ \delta_{31} & \delta_{32} & \delta_{33} \end{bmatrix} \\ &= \Lambda_W \Psi_W \Lambda_W' + \Theta_W \\ &= \begin{bmatrix} 1 \\ \lambda_{W2} \\ \lambda_{W3} \end{bmatrix} [\psi_W] \begin{bmatrix} 1 & \lambda_{W2} & \lambda_{W3} \end{bmatrix} \\ &\quad + \begin{bmatrix} \theta_{W1} & & \\ 0 & \theta_{W2} & \\ 0 & 0 & \theta_{W3} \end{bmatrix}, \\ \mathrm{cov}\left(\begin{bmatrix} u1_{0j} \\ u2_{0j} \\ u3_{0j} \end{bmatrix} \right) &= \Sigma_B = \mathrm{P}_{0|0,j} = \begin{bmatrix} \tau_{11} & & \\ \tau_{21} & \tau_{22} & \\ \tau_{31} & \tau_{32} & \tau_{33} \end{bmatrix} \\ &= \Lambda_B \Psi_B \Lambda_B' + \Theta_B \\ &= \begin{bmatrix} 1 \\ \lambda_{B2} \\ \lambda_{B3} \end{bmatrix} [\psi_B] \begin{bmatrix} 1 & \lambda_{B2} & \lambda_{B3} \end{bmatrix} + \begin{bmatrix} \theta_{B1} & & \\ 0 & \theta_{B2} & \\ 0 & 0 & \theta_{B3} \end{bmatrix}. \end{aligned} \qquad (24)$$

Fundamentally, this represents a simple algebraic trick, transferring the information from the two covariance matrices to their decomposed elements. The rest of the specification for this model is the same as that in Block 20 (except for $\Theta_{\cdot j}$, which is already reparameterized in Block 24). Therefore, this model still has 15 parameters to be estimated.

Example 5. The results of fitting the random intercepts confirmatory factor model in Mplus and PROC IML are presented in Table 5, and they are essentially identical. Using the parameter estimates in Example 5, we can recover the model implied within- and between-group covariance matrices, which are parallel to the specification adopted in

TABLE 5
Results of the Random Intercepts Confirmatory Factor Model

	Mplus		SSM-IML	
Parameter	Point	*SE*	Point	*SE*
$\gamma 1_{00}$	3.677	.073	3.677	.073
$\gamma 2_{00}$	3.731	.082	3.731	.082
$\gamma 3_{00}$	3.706	.080	3.706	.080
λ_{W2}	.978	.040	.979	.040
λ_{W3}	1.070	.044	1.071	.044
λ_{B2}	1.039	.234	1.037	.233
λ_{B3}	1.091	.228	1.081	.225
ψ_W	1.248	.074	1.247	.074
ψ_B	.107	.042	.108	.042
θ_{W1}	1.769	.063	1.769	.063
θ_{W2}	1.868	.063	1.868	.063
θ_{W3}	1.510	.065	1.510	.065
θ_{B1}	.025	.019	.025	.019
θ_{B2}	.058	.026	.058	.026
θ_{B3}	.037	.023	.038	.023
−2*loglik	42566.158		42566.158	

Example 4. In fact, the recovered matrices are almost identical to the results in Table 4. The purpose of presenting both Example 4 and Example 5 is to provide an easy transition from the multilevel regression model to the multilevel confirmatory factor model.

SUMMARY AND DISCUSSION

In this article, we provide five examples to illustrate the state space approach for estimating linear multilevel models. Starting from the state space approach for estimating univariate and multivariate multilevel regression models, the approach is further extended to a multilevel factor model. Throughout the examples, the results from the state space approach are essentially identical to those from multilevel regression modeling (e.g., PROC MIXED, MLwiN) and SEM (e.g., Mplus) software.

Because the state space approach is an extremely flexible modeling framework, the state space form we illustrate in each example is only one of the several possible state space forms. In general, the solution to representing the multilevel models using state space forms is to (a) include the random effects in the latent state vector and then (b) initialize the Kalman recursion by properly specifying the covariance matrix ($P_{0|0,j}$) associated with the latent state vector ($\eta_{0|0,j}$) to subsume the parameters (i.e., the random effects) at the cluster level. Once a proper state space form is obtained, the Kalman filter algorithm can be implemented to estimate the parameters.

We explicitly stressed one important advantage of the state space approach (and empirically illustrated it in Example 1 and Example 2)—the KF algorithm utilized in the state space model is more efficient than the algorithms used in the multilevel regression model. In particular, the large matrix problem is avoided in the state space approach regardless of the number of Level 1 units within each Level 2 unit or the number of observations on each subject. The specially designed high performance procedure provided by SAS (i.e., the HPMIXED procedure) can partially solve the large matrix problem, but PROC HPMIXED is efficient only for models containing large sparse matrices (Wang & Tobias, 2009). Therefore, the state space approach offers researchers an efficient alternative to fit multilevel regression models with a large number of Level 1 units within each Level 2 unit or a large number of observations on each subject in a longitudinal study.

Another important application of the state space approach is the estimation of the latent state vector. For a single-subject state space model, the parameters are first estimated by maximizing the PED function from the Kalman filter, and then the estimates are used in the Kalman smoother, a backward recursion initialized with $\eta_{T|T}$ and $P_{T|T}$, to compute the smoothed latent state vector, $\eta_{i|T}$, and its covariance matrix, $P_{i|T}$, for $i =$ T-1, T-2, . . ., 1 (for technical details, see Harvey, 1989; Song & Ferrer, 2009). Essentially, the Kalman smoother propagates back the stored information in the last observation to all previous observations. Thus, it produces conditional expectations for all observations based on the information of the entire sample. For the multiple-subject state space model, the Kalman smoother is initialized with $\eta_{T|T,j}$ and $P_{T|T,j}$ for subject *j* and proceeds in the same ways as in the single-subject model. Applying the state space approach to multilevel models, the smoothed values in latent state vector for subject *j* are actually the estimated random effects for cluster *j* such that the smoothed latent state vector varies only between subjects/clusters but remains invariant within a subject/cluster. Taking the estimates from Example 1, the smoothed values from PROC IML are identical to the estimated random effects from PROC MIXED. However, the standard errors for the random effects from the two approaches are not the same. The calculated correlations between the two types of standard errors are 0.63 and 0.19 for the random intercepts and random slopes, respectively. Future research is warranted to investigate the discrepancy between the standard errors from the two approaches.

One limitation of the state space approach is that no commercial software currently has provided convenient modeling tools for routine applications. Researchers are obliged to write their own programs to implement the state space approach described in this article.[9] The PROC IML code can be easily translated to any matrix-oriented programming language (e.g., R, MATLAB) if the researcher is more familiar with other software. Implementation of the state space approach is not currently feasible using the free Fortran program, MKFM6, provided by Dolan (2005; see also Zhang,

[9] The PROC IML code used in this article can be requested from the first author.

Hamaker, & Nesselroade, 2008, for the use of the code [by Dolan]) because the elements in $P_{0|0,j}$ are not allowed to be freely estimated. The Fortran code provided by Jones (1993) is a valuable resource, and it can be modified for specialized purposes.

REFERENCES

Bauer, D. J. (2003). Estimating multilevel linear models as structural equation models. *Journal of Educational and Behavioral Statistics*, *28*, 135–167.

Bollen, K. A. (1989). *Structural equations with latent variables*. New York, NY: Wiley.

Chow, S.-M., Ho, M.-H. R., Hamaker, E. L., & Dolan, C. V. (2010). Equivalence and differences between structural equation modeling and state-space modeling techniques. *Structural Equation Modeling*, *17*, 303–332.

Curran, P. J. (2003). Have multilevel models been structural equation models all along? *Multivariate Behavioral Research*, *38*, 529–569.

Dolan, C. V. (2005). *MKFM6: Multi-group, multi-subject stationary time series modeling based on the Kalman filter*. Retrieved from http://users/fmg.uva.nl/cdolan/

Durbin, J., & Koopman, S. J. (2001). *Time series analysis by state space methods*. New York, NY: Oxford University Press.

Goldstein, H. (1986). Multilevel mixed linear model analysis using iterative generalized least squares. *Biometrika*, *73*, 43–56.

Goldstein, H. (1995). *Multilevel statistical models* (2nd ed.). London, UK: Edward Arnold.

Harvey, A. C. (1989). *Forecasting, structural time series models and the Kalman filter*. Princeton, NJ: Princeton University Press.

Ho, M.-H. R., Shumway, R., & Ombao, H. (2006). State-space approach to modeling dynamic processes: Applications in biological and social sciences. In T. A. Walls & J. L. Schafer (Eds.), *Models for intensive longitudinal data* (pp. 148–170). New York, NY: Oxford University Press.

Hox, J. J. (2002). *Multilevel analysis: Techniques and applications*. Mahwah, NJ: Erlbaum.

Icaza, G., & Jones, R. H. (1999). A state-space EM algorithm for longitudinal data. *Journal of Time Series Analysis*, *20*, 537–550.

Jones, R. H. (1993). *Longitudinal data with serial correlation: A state-space approach*. London, UK: Chapman & Hall.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering (Series D)*, *82*, 35–45.

Kamata, A., Bauer, D. J., & Miyazaki, Y. (2008). Multilevel measurement modeling. In A. A. O'Connell & D. B. McCoach (Eds.), *Multilevel modeling of educational data* (pp. 345–388). Charlotte, NC: Information Age.

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

Mehta, P. D., & Neale, M. C. (2005). People are variables too: Multilevel structural equations modeling. *Psychological Methods*, *10*, 259–284.

Schweppe, F. (1965). Evaluation of likelihood functions for Gaussian signals. *IEEE Transactions on Information Theory*, *11*, 61–70.

Snijders, T. A. B., & Bosker, R. J. (2012). *Multilevel analysis: An introduction to basic and advanced multilevel modeling*. Thousand Oaks, CA: Sage.

Song, H., & Ferrer, E. (2009). State-space modeling of dynamic psychological processes via the Kalman smoother algorithm: Rationale, finite sample properties, and applications. *Structural Equation Modeling*, *16*, 338–363.

Wang, T., & Tobias, R. (2009, March). *All the cows in Canada: Massive mixed modeling with the HPMIXED procedure in SAS 9.2*. Paper presented at SAS Global Forum 2009, Washington, DC.

Zhang, Z., Hamaker, E. L., & Nesselroade, J. R. (2008). Comparisons of four methods for estimating a dynamic factor model. *Structural Equation Modeling*, *15*, 377–402.