8-25-2016

Partially and fully time-unstructured residual variance-covariance matrices in growth curve modeling: Consequences of ignoring variability in times of assessment

Patrick Coulombe

Patrick Coulombe
Candidate

Psychology
Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Harold Delaney, Chairperson

James Selig

Katie Witkiewitz

Jacob Vigil

PARTIALLY AND FULLY TIME-UNSTRUCTURED RESIDUAL VARIANCE-COVARIANCE MATRICES IN GROWTH CURVE MODELING: CONSEQUENCES OF IGNORING VARIABILITY IN TIMES OF ASSESSMENT

by

PATRICK COULOMBE

B.S., PSYCHOLOGY, UNIVERSITÉ DU QUÉBEC À MONTRÉAL, 2010
M.S., PSYCHOLOGY, UNIVERSITY OF NEW MEXICO, 2014

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Psychology

The University of New Mexico
Albuquerque, New Mexico

July 2016

DEDICATION

To my brother, Philippe Coulombe, and my advisor, Harold Delaney, both of whom I admire.

ACKNOWLEDGMENTS

I will be forever grateful to my advisor, Harold Delaney, who orchestrated my admission into the Quantitative program, and who at times believed in my potential more than I did.

I owe an incalculable debt of thanks to my committee members, Katie Witkiewitz, James Selig, Jacob Vigil, and Harold Delaney, all four of whom played an instrumental role in helping me start off my professional career. They have shared their resources and have provided me with opportunities I would not have gotten without them. I also thank them for many stimulating conversations over the years.

Finally, I want to thank my father, Daniel Coulombe, my stepmother, Christianne Doré, my mother, Lucille Brière, and my brother, Philippe Coulombe, for their unconditional support despite being separated by a border.

PARTIALLY AND FULLY TIME-UNSTRUCTURED RESIDUAL VARIANCE-COVARIANCE MATRICES IN GROWTH CURVE MODELING: CONSEQUENCES OF IGNORING VARIABILITY IN TIMES OF ASSESSMENT

by

Patrick Coulombe

B.S., Psychology, Université du Québec à Montréal, 2010

M.S., Psychology, University of New Mexico, 2014

Ph.D., Psychology, University of New Mexico, 2016

ABSTRACT

In a longitudinal study, the data can be either time-structured, where the times of assessment are the same across participants, or time-unstructured, where the times of assessment vary across participants. Currently, growth curve modeling can accommodate time-unstructured data when modeling average change over time (fixed effects) and variability in change over time (random effects), but not residual variances and covariances. Through Monte Carlo simulation, a first goal of this study was to determine the effects of ignoring variability in times of assessment when modeling the residual variances and covariances in linear growth curve models in terms of convergence, parameter bias, power to detect change over time, and model fit. A second goal was to evaluate a novel method to construct "partially" time-unstructured matrices for residual variances and covariances. The simulation factors manipulated in this study are type of analysis (time-structured, partially time-unstructured, fully time-unstructured), residual

matrix (heterogeneous diagonal, first-order autoregressive), sample size (50, 200, 500), and number of time points (3, 6, 9). Results showed that convergence was generally high when the matrix was autoregressive, whereas when the matrix was heterogeneous diagonal, the time-structured and partially time-unstructured analyses converged most often; in many conditions, the fully time-unstructured analysis never converged. Fixed effects were generally spared from bias across conditions, as were the random effects when the matrix was heterogeneous diagonal. With the time-structured autoregressive matrix and only 3 time points, the intercept and slope variances were overestimated, and the intercept-slope correlation and residual variance were underestimated. There was a large effect of type of analysis on autocorrelation bias, with only the fully time-unstructured analysis yielding unbiased estimates, and the partially time-unstructured analysis yielding less bias than the time-structured analysis. Power to detect change over time was high across conditions. In terms of model fit, all fit indexes examined favored time-unstructured analyses when the matrix was autoregressive, whereas only deviance favored time-unstructured analyses when the matrix was heterogeneous diagonal. Overall, this study shows that accommodating time-unstructured data when modeling residual variances and covariances can be important, perhaps especially when residuals are autocorrelated. Moreover, when the fully time-unstructured matrix cannot be used, the partially time-unstructured matrix provides an improvement over the standard time-structured matrix under certain conditions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

Psychologists often design longitudinal studies in order to examine change over time in variables of interest. The resulting sample data can take one of two forms (Coulombe, Selig, & Delaney, 2016; Singer & Willett, 2003). First, the data can be *time-structured*, meaning that all participants are assessed at exactly the same times throughout the course of the study. Such a situation arises when assessments are done in groups (for example, all students in a given classroom are assessed during the same class period). Conversely, the data can be *time-unstructured*, meaning that not all participants are assessed at the same time; instead there are some differences in times of assessment across participants. Given this distinction, unless participants are assessed in groups, most longitudinal studies will have data that are time-unstructured to some degree.

The analysis of time-unstructured data poses more difficulties than the analysis of time-structured data. Fortunately, statistical methods do exist that accommodate to some extent time-unstructured data when investigating change over time (Coulombe et al., 2016). However, while such methods allow for time-unstructured data in modeling fixed effects (e.g., average linear change over time) and random effects (e.g., variability in linear change over time), no methods currently exist to accommodate time-unstructured data in modeling residual variances and covariances over time. For this reason, researchers generally ignore—by necessity—variability in times of assessment when modeling residual (co)variances. The consequences of this practice have not been investigated as yet in the methodological literature. Therefore, the goal of the current study is to examine the effects of ignoring variability in times of assessment when

modeling residual (co)variances in a linear growth curve model, in the absence of any other misspecification. I also investigate the performance of a partial solution to this problem, whereby the variability in times of assessment is reduced but not completely ignored. If the consequences of treating time-unstructured data as time-structured when modeling residual (co)variances are found to be important and difficult to ameliorate using the partial solution investigated here, then the current study would point to the need to develop new methods to correctly accommodate time-unstructured data when modeling residual (co)variances.

The rest of this introduction is organized as follows. First, I briefly provide some context for the current study by describing the linear growth curve model. Next, I review current methods available to accommodate time-unstructured data when investigating change over time. Following that, I describe recent research that has examined the consequences of treating time-unstructured data as time-structured when modeling fixed and random effects in growth curve modeling. I then describe research that has investigated the consequences of misspecifying the residual (co)variances specifically. Next, I review the sensitivity of some fit indexes to detect misspecified residual (co)variances. Finally, I conclude the introduction by describing the current study.

Background

To provide background for the current study, I begin by describing growth curve modeling in this section. This background is included for potential readers who are not already familiar with growth curve modeling in general, or with the multilevel modeling approach to growth curve modeling in particular.

Growth curve modeling (GCM; e.g., Curran, Obeidat, & Losardo, 2010) is a statistical approach used to model change over time. In a growth curve model, time is included as a predictor of the dependent variable. One distinguishing feature of the growth curve model is that two types of effects can be examined: fixed effects and random effects. Fixed effects represent *average effects* of predictors (e.g., time) on the dependent variable, averaging across all individuals. Random effects, on the other hand, represent *variability* in the effect of predictors across individuals. Specifically, when the slope (or coefficient) of a predictor in a growth curve model is random, the value of the slope of that predictor is allowed to vary across individuals.

There are two main approaches to growth curve modeling: multilevel modeling and structural equation modeling. I mention both approaches in the next section when I contrast them in terms of time-unstructured data. In this section, however, I focus on the multilevel modeling approach to growth curve modeling, which is the approach that I adopt in this study.

In multilevel modeling (MLM; e.g., Hox, 2010; Raudenbush & Bryk, 2002; Singer & Willett, 2003) of longitudinal data, a linear growth curve model is specified using equations at two different levels, the observational level (level 1) and the individual level (level 2), which allow for both random slopes and intercepts:

$$\text{Level 1: } y_{ti} = \beta_{0i} + \beta_{1i}T_{ti} + \epsilon_{ti} \quad (1)$$

$$\text{Level 2: } \beta_{0i} = \gamma_{00} + u_{0i} \quad (2)$$

$$\beta_{1i} = \gamma_{10} + u_{1i}$$

These equations can be combined into a reduced-form expression:

$$y_{ti} = \gamma_{00} + \gamma_{10}T_{ti} + u_{0i} + u_{1i}T_{ti} + \epsilon_{ti} \quad (3)$$

In these equations, y_{ti} is the value of the dependent variable for individual *i* at measurement occasion *t*, γ_{00} is the mean intercept, and γ_{10} is the mean linear slope. These two parameters do not vary from person to person, and represent fixed effects. Variable T_{ti} is the time value at measurement occasion *t*, and the subscript *i* on variable *T* indicates that times of assessment are allowed to vary across persons. The other terms in the model represent error terms, or deviation from mean effects: u_{0i} is the difference between each person's intercept and the mean intercept; u_{1i} is the difference between each person's time slope and the mean time slope; and ϵ_{ti} is the difference between each person's expected score at wave *t* and the person's actual score at that time. The two u error terms reside at level 2 and vary only between persons, and together these error terms represent random effects. The ϵ error term resides at level 1 and varies across time within each person. The error terms are assumed to be normally distributed with a mean of 0. Further, level-1 residuals are assumed to be independent from level-2 residuals:

$$\epsilon_{ti} \sim N(0, \sigma^2)\ ;\ (u_{0i}, u_{1i}) \sim MVN([0 \quad 0], \begin{bmatrix} \tau_{00} & \tau_{10} \\ \tau_{10} & \tau_{11} \end{bmatrix}) \qquad (4)$$

The focus of this study is on the left-hand side of Equation 4, namely the level-1 residual variances and covariances. Equation 4 defines the residuals for one assessment and one person at a time. Using matrix notation, it is possible to specify the residuals for all measurement occasions for a given individual simultaneously. Equation 4 then becomes:

$$\boldsymbol{\epsilon_i} \sim N(\mathbf{0}, \boldsymbol{I}\sigma^2) \qquad (5)$$

In Equation 5, $\boldsymbol{\epsilon_i}$ is a $(t \times 1)$ vector of occasion-specific residuals stemming from a normal distribution with a $(t \times 1)$ mean vector of zeros, and a $(t \times t)$ homogeneous-diagonal variance-covariance matrix. Assuming the simple situation of *t* = 3 assessments

per participant, expanding Equation 5 to show the corresponding $\boldsymbol{I}\sigma^2$ homogeneous-diagonal variance-covariance matrix yields:

$$\begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix} \quad (6)$$

The default in most multilevel modeling software packages is to specify that the variances and covariances of the level-1 residuals may be characterized by a homogeneous-diagonal variance-covariance matrix, as was done in Equation 6. In a homogeneous-diagonal matrix, the residual variance is constant over time (i.e., all diagonal entries in Equation 6 are identical), and the residual covariances are zero (i.e., all off-diagonal entries in Equation 6 are 0). However, researchers can decide to estimate separate residual variances for each measurement occasion (when the residual variance changes over time), and/or to estimate residual correlations or covariances. Two other matrix structures are of interest in this study: the heterogeneous-diagonal matrix, where the diagonal entries of Equation 6 are freely estimated rather than constrained to equality; and the first-order autoregressive matrix, denoted AR(1), where the diagonal entries are the same as in Equation 6, but the off-diagonal entries are estimated using an autocorrelation parameter. I provide more detail regarding the structure of these two matrices and modified versions of them below.

One crucial thing to note about the matrix presented in Equation 6 is that the dimensions (and therefore the parameters) of the matrix are generally determined by the number of assessments rather than the number of distinct time values in the sample. If residual variances are constant over time and residual correlations are 0 at all times, then Equation 6 is an appropriate yet parsimonious matrix to model the residual (co)variances. However, if the entries of the level-1 matrix are a function of time (as is the case when

the matrix is heterogeneous diagonal or first-order autoregressive), then the correctly specified matrix is determined by the individual time values rather than the assessment numbers. More generally, when the data are time-unstructured, special care must be given to the choice of the time variable used to define the level-1 matrix. In this study, I examine the consequences of misspecifying the level-1 matrix by using assessment number rather than time values when the data are time-unstructured. Since time values cannot be used to define the level-1 matrix when every single person is assessed at different times, I also evaluate the performance of a novel method to construct "partially" time-unstructured matrices, where some but not all of the variation in times of assessment is used in constructing the level-1 matrix.

Analysis of Time-Unstructured Data

Traditionally, researchers interested in change over time in a given variable could examine change in means over time through repeated-measures (within-subjects) analysis of variance (ANOVA; e.g., Keppel & Wickens, 2004; Maxwell & Delaney, 2004). However, whether or not the specific times of assessment were collected for each participant, the times of assessment cannot vary from person to person in repeated-measures ANOVA. Therefore, repeated-measures ANOVA cannot accommodate time-unstructured data when investigating change over time.

With recent advances, researchers can employ growth curve modeling instead of repeated-measures ANOVA to examine change over time in variables of interest. Two types of effects are potentially of interest in growth curve modeling (see "Background" above). First, as noted previously, researchers can look at the fixed effects, which refer to average effects across all participants (e.g., average linear change over time). Second,

researchers can look at the random effects, which refer to variability in effects across participants (e.g., variability in extent of linear change over time from one person to the next). As I describe below, both of these types of effects can be estimated while accommodating time-unstructured data.

Growth curve modeling can be implemented through two main analytical frameworks: multilevel modeling and structural equation modeling. In multilevel modeling, the time variable is included as a predictor of the dependent variable. Just like any other predictor variable, the specific values on the time predictor can vary across assessments and across participants; as such, multilevel modeling can accommodate time-unstructured data when analyzing change over time. In the second framework for growth curve modeling, structural equation modeling (SEM; e.g., Kline, 2015; Meredith & Tisak, 1990), the times of assessment are included through a matrix of factor loadings. However, in the standard growth curve model, this matrix of factor loadings is not allowed to vary across individuals. As such, without further adjustment, the SEM framework cannot accommodate time-unstructured data when analyzing change over time (Coulombe et al., 2016).

Fortunately, methods have been developed to accommodate time-unstructured data in the SEM framework. One possibility is to use a missing-data approach (Bauer, 2003; Curran, 2003), where there are as many time variables as there are different times of assessment. Another more convenient method is to use *definition variables* (Mehta & West, 2000), whereby time values stored in a variable in the dataset are used to define the factor loadings separately for each individual. With either of these methods, the SEM framework to growth curve modeling can accommodate time-unstructured data, and

becomes essentially equivalent to the MLM approach. Sterba (2014) shows how to fit nonlinear growth curve models (including polynomial, piecewise and exponential models) with time-unstructured data using definition variables (see also Liu, Liu, Li, & Zhao, 2015). She also provides corresponding software syntax. Definition variables have been implemented in software packages such as Mplus (Muthén & Muthén, 2012) and the open-source software OpenMx (OpenMx Development Team, 2016).

Since MLM can accommodate time-unstructured data without adjustment while SEM cannot, readers might wonder: Why might researchers want to estimate growth curve models through SEM rather than MLM when the data are time-unstructured data? Coulombe et al. (2016) give a list of advantages of SEM over MLM, which include: (1) the ability to control for measurement error through latent variables; (2) more natural accommodating of multiple dependent variables; and (3) easier handling of missing data on predictor variables. In addition, Sterba (2014) mentions that in SEM, it is possible to estimate the functional form of the change over time directly from the sample data, whereas this is not yet implemented in MLM.

Whether researchers opt for MLM or SEM for modeling change over time, the analysis will be able to accommodate time-unstructured data when modeling the fixed and random effects. However, both approaches also require the researcher to estimate residual variances for the dependent variable at each measurement occasion, that is, the variance in the dependent variable at each occasion that cannot be accounted for by the model of change (i.e., by the fixed and random effects). The researcher also needs to specify a model for the residual covariances, that is, the covariance between residuals at different occasions. While methods that can accommodate time-unstructured data have

been developed for the modeling of fixed and random effects, such methods currently do not exist for the modeling of residual (co)variances. Moreover, the consequences of failing to account for the variability in times of assessment when estimating residual (co)variances—even when such variability is correctly accommodated when estimating the fixed and random effects—are not known. Therefore, the goal of the current study is to examine the consequences of treating time-unstructured data as though they were time-structured when estimating residual (co)variances in the absence of any other misspecification. A secondary goal of the current study is to evaluate the performance of one potential method to accommodate time-unstructured data when estimating residual (co)variances.

Alternative approaches with time-unstructured data. Before moving on to the next section, which discusses the consequences of ignoring variability in times of assessment across participants, I mention three other analytical approaches available to researchers for growth curve modeling with time-unstructured data. First, researchers can use multilevel structural equation modeling (MSEM; Mehta & Neale, 2005), which combines the strengths of multilevel modeling and structural equation modeling. Second, researchers can use traditional (single-level) regression, but correct the standard errors of the hypothesis tests to account for the fact that the same participants contributed to more than one observation (McNeish, Stapleton, & Silverman, in press). Finally, researchers can use generalized estimating equations (GEE; McNeish et al., in press), which corrects both the parameter estimates and the standard errors to account for the repeated measures within participants. Unlike MLM, SEM, and MSEM, the last two methods mentioned in this paragraph (cluster-robust standard errors and generalized estimating equations) do

not provide estimates of the random effects, but require fewer assumptions and can provide output with which researchers are already familiar, such as the R^2 effect size (McNeish et al., in press).

Consequences of Treating Time-Unstructured Data as Time-Structured

There are currently no studies that have investigated the consequences of ignoring variability in times of assessment across participants when estimating residual (co)variances in a growth curve model. However, some recent studies have examined the consequences of ignoring such variability in times of assessment when estimating the fixed and random effects. In this section, I review these consequences of treating time-unstructured data as though they were time-structured when modeling change over time.

Some of the current knowledge on the consequences of incorrectly treating time-unstructured data as time-structured was gained through sensitivity analyses, where researchers analyzed a single dataset in two ways, once with a time-unstructured analysis and once with time-structured analysis, and compared the results. Comparing the results of the two analyses gives an indication of how results from a single study can change depending on the analysis chosen. Coulombe et al. (2016) report two sensitivity analyses, and I follow their exposition. In their sensitivity analysis, Singer and Willett (2003, Ch. 5) estimated a growth curve model twice, each with a different time variable: (1) in the time-unstructured analysis, they used the child's actual age at the time of assessment; and (2) in the time-structured analysis, they assumed that every child was exactly 6.5, 8.5, and 10.5 years old at the three assessments, which were the ages at which the researchers had originally planned to assess the children. Although they did not have access to the population values, they found that, compared to the time-unstructured analysis, the time-

structured analysis yielded larger estimates of the linear slope and of the intercept and slope variances. In a similar sensitivity analysis, Mehta and West (2000) used assessment (wave) number rather than age to track time, and found that relative to known population values, the intercept variance was overestimated and the covariance between the intercept and linear slope was closer to 0. Taken together, these sensitivity analyses suggest that incorrectly pairing a time-unstructured dataset with a time-structured analysis can lead to different estimates of the fixed and random effects than would be obtained through the corresponding time-unstructured analysis.

More recently, methodologists have systematically evaluated the consequences of treating time-unstructured data as time-structured through Monte Carlo simulation studies. Aydin, Leite, and Algina (2014) simulated longitudinal datasets with linear change over time in which there was variability in times of assessment at each measurement occasion. Times of assessment at each occasion followed either a uniform distribution or a skewed distribution. They found that, in all conditions examined, the time-structured analysis yielded unbiased estimates of the mean intercept and linear slope (the fixed effects) and covariance between the intercept and slope. However, the time-structured analysis underestimated intercept and slope variances (random effects) when the times of assessment followed a skewed distribution around each measurement occasion. In a similar study, Coulombe et al. (2016) examined the effects of ignoring variability in times of assessment when estimating linear growth curve models that varied in number of assessments per person. They found that when there were only 3 assessments per person and sample size was small (particularly when $n \leq 100$), the intercept and slope variances were overestimated, and the intercept-slope covariance was

underestimated. This overestimation of the intercept and slope variances is in contrast to Aydin et al. (2014) but is in line with Singer and Willet (2003) and Mehta and West (2000). As was the case in Aydin et al., the fixed effects were generally unbiased, except when a large amount of variation in times of assessment was ignored and the times of assessment followed a skewed distribution around each measurement occasion, in which case the mean intercept was slightly overestimated. Coulombe et al. also found that ignoring variation in times of assessment was often associated with convergence problems, whereas the time-unstructured analysis nearly always converged on a proper solution (see also Liu et al., 2015).

At least one simulation study compared the time-unstructured and time-structured analyses when the model is complex and change over time is not linear. Specifically, Liu et al. (2015) sought to determine the consequences of ignoring variation in times of assessment in a piecewise growth curve model, where change over time is modeled as a function of two linear splines. They manipulated the amount of variation in times of assessment at each measurement occasion, the change in slope at the knot, and sample size. They found that estimates were more variable from one sample to the next using the time-structured analysis instead of the time-unstructured analysis, and the time-structured analysis yielded increasingly less precise estimates as the times of assessment became more variable; no such difficulty was observed with the time-unstructured analysis. They also noted sometimes large convergence problems (as did Coulombe et al., 2016). The only situation when the time-structured and time-unstructured analyses were comparable was when there was very little variability in the times of assessment (all assessments were within $1/12^{th}$ of one unit on the time scale around each measurement occasion). In

that situation, namely when times of assessment are all within $1/12^{th}$ of one time unit of each measurement occasion, Liu et al. recommend—for mere simplicity—to use the time-structured analysis rather than the time-unstructured analysis. However, since there is no apparent cost to using a time-unstructured analysis when the data are time-unstructured and their recommendation is based on a single study, I deem it more prudent to use a time-unstructured analysis whenever one has time-unstructured data; this is similar to the recommendation by Nezlek (2008) of using multilevel modeling rather than single-level regression whenever in the presence of clustered data, regardless of the amount of clustering (as indexed by the intraclass correlation).

Taken together, the simulation studies described above (Aydin et al., 2014; Coulombe et al., 2016; Liu et al., 2015) suggest that ignoring variation in times of assessment in growth curve modeling can yield biased estimates of the growth factor variances and covariance (the random effects), and potentially of the fixed effects. A time-structured analysis coupled with a time-unstructured dataset can also lead to convergence problems and to parameter estimates that are more variable from one random sample to the next.

Consequences of Misspecifying Residual (Co)Variances

One of the assumptions of the growth curve model is that the level-1 error matrix is correctly specified (McNeish et al., in press). The focus of the current study is on the consequences of misspecifying the level-1 error matrix by ignoring variability in times of assessment at each measurement occasion, namely by using an insufficiently precise time variable. Most of the previous research in this area, however, has focused on the consequences of misspecifying the structure of the level-1 matrix (for example, by using

a homogeneous diagonal matrix when the population matrix is first-order autoregressive). In this section, I review research that has investigated the consequences of misspecifying the residual (co)variances when modeling change over time. First, I describe a few sensitivity analyses that have compared growth curve models that differed only in their level-1 error matrix. Then, I review several simulation studies that have examined the consequences of misspecifying the level-1 error matrix.

Single-study investigations: Sensitivity analyses. Several authors have encouraged researchers to think carefully about which level-1 matrix structure to choose, often in the context of tutorials. In one such tutorial, Harring and Blozis (2014) use SAS to show how to deviate from the default homogeneous diagonal structure in the context of nonlinear multilevel models (see also Wolfinger, 1993). In particular, they implement several level-1 error structures that include correlated residuals and/or heterogeneous variances over time for a series of nonlinear growth models, including quadratic, exponential, and piecewise growth models. Littell, Pendergast, and Natarajan (2000) also provide a tutorial for specifying different level-1 error matrices in SAS, but they also conduct a sensitivity analysis to show that the estimates of the fixed effects, along with their hypothesis tests, are sensitive to the choice of the level-1 error structure. They show that when the chosen level-1 error structure does not provide good fit to the sample data (in their case, a first-order autoregressive structure), the test statistics corresponding to the fixed effects greatly differed from the ones obtained with better-fitting level-1 error structures. They also show that for certain contrasts, the estimate of the contrast itself remains invariant across level-1 error structures, but the standard error—and therefore the test—of the contrast again depends on which level-1 error structure is chosen. This effect

of the level-1 matrix on standard errors is in line with an example provided by Singer and Willett (2003), who found that standard errors for the fixed effects can be reduced by using a more parsimonious level-1 error matrix than a fully unconstrained (i.e., unstructured) matrix, yielding higher power to detect the fixed effects.

In trying to increase the number of level-1 matrices available to researchers interested in change over time, Grimm and Widaman (2010) developed two new level-1 error structures. Both structures are based on growth curve reliability, i.e. the ratio of reliable (latent) variance to residual variance. The first structure assumes invariant growth curve reliability over time, while the second structure assumes that the growth curve reliability changes linearly with time. In both structures, residual correlations are assumed to be 0, but residual variances are allowed to vary over time (similar to a heterogeneous diagonal matrix). In their study, Grimm and Widaman performed sensitivity analyses using data from several samples where they compared the results obtained when modeling the level-1 error matrix either as a homogeneous diagonal matrix, a heterogeneous diagonal matrix, or one of the two matrices based on growth curve reliability. Similar to previous studies, they found that the fixed effects were unaffected by the choice of the level-1 matrix, but the estimated covariance between the growth factors varied across level-1 matrices.

Combined, the sensitivity analyses reported here suggest that the choice of the level-1 matrix in growth curve models can affect the estimates of the fixed and random effects. Moreover, the choice of the level-1 matrix might have an even larger impact on the hypothesis test of the fixed effects, which are typically of interest to substantive researchers.

Systematic investigations: Simulation studies. Sensitivity analyses often cannot make use of population values, so any difference between the correctly and incorrectly specified models may or may not be to the advantage of the correct model. For example, even the correct, time-unstructured analysis can yield biased estimates in certain situations, particularly in small samples with few time points (Coulombe et al., 2016). In contrast, simulation studies allow for the comparison between the parameter estimates obtained from a given analysis and the corresponding population values. Further, simulation studies are based on hundreds or thousands of random samples, making them less prone than sensitivity analyses to conclusions based on characteristics unique to only a few datasets. In the next paragraphs, I review several simulation studies that have examined the consequences of misspecifying the level-1 error matrix (for simulation studies focusing on the misspecification of the level-2 matrix, see Barr, Levy, Scheepers, & Tily, 2013; Litière, Alonso, & Molenberghs, 2007).

***Misspecifying the structure of the level-1 matrix*.** Up to now in the methodological literature, most simulation studies that have addressed the consequences of misspecifying the level-1 error matrix in growth curve modeling have focused on misspecifying the structure of the matrix (for example, modeling heterogeneous variances as homogeneous). In one such simulation study, Ferron, Dailey, and Yi (2002) investigated the effects of misspecifying the level-1 matrix in a linear growth curve model. In that study, residuals were generated using a first-order autoregressive structure, but were modeled using a homogeneous diagonal matrix. They found that the fixed effects were estimated with little bias in most conditions, except when change over time was nonlinear; in that case, bias was further worsened when the interval between

assessments varied over the course of the study. Further, the hypothesis test of the linear slope was sometimes liberal. The incorrect simplification of the level-1 matrix had even more deleterious effects on the random effects, with the intercept variance being overestimated in all conditions, particularly when the autocorrelation between residuals was large. The slope variance was also overestimated, more so when there were fewer assessments per individual. The intercept-slope covariance was also found to be systematically negative, despite being 0 in the population, and this bias was worse when there were few assessments per individual and the autocorrelation between residuals was large. In sum, both the fixed and random effects—but mostly the random effects—are affected by assuming a homogeneous structure when the population matrix is AR(1), and few assessments and large residual autocorrelation both worsen these undesirable effects. In another simulation study with nearly identical results, misspecifying an AR(1) level-1 matrix (among other structures) as a homogeneous diagonal matrix in linear growth curve models was again not associated with any bias in the fixed effects, but the random effects were again biased (Shi, 2009). In that study, both the misspecified and the correctly specified models overestimated the intercept variance, but bias was worse when the level-1 matrix was misspecified and the autocorrelation was large. Again replicating Ferron et al. (2002), Shi (2009) reports overestimated slope variance and underestimated intercept-slope covariance.

Jacqmin-Gadda, Sibillot, Proust, Molina, and Thiébaut (2007) also examined the effects of specifying a simpler level-1 error structure than the structure that generated the data. Specifically, in their simulation, they estimated growth curve models with a homogeneous diagonal error structure when the population error structure was AR(1) or

heteroscedastic (among others). They found that, when the residual variances were increasing linearly with time, the coverage rates for the fixed effects were generally adequate (i.e., the 95% confidence intervals for the fixed effects contained the true population value in about 95% of random samples). However, when the population level-1 error matrix was AR(1), some coverage rates were lower than 95% (sometimes below 80%), particularly when the growth curve model had random intercepts but fixed slopes (i.e., the variance of the slopes was fixed to 0). One novelty provided by this study is the distinction between random-slopes and fixed-slopes models; I use only random-slopes models in the current study, but I discuss this and other potential factors to manipulate in future studies in the discussion.

To date, Kwok, West, and Green (2007) provided what is probably the most comprehensive simulation work pertaining to the misspecification of the level-1 error matrix. These authors were interested in the effects of misspecifying the level-1 error matrix in linear and quadratic growth curve models by using either a matrix similar in structure to the true matrix but with fewer parameters (underspecification), a matrix similar in structure to the true matrix but with more parameters (overspecification), or a matrix with an altogether different structure than the true matrix (general misspecification). They found that underspecification and general misspecification of the level-1 error matrix were associated with lower power to detect the fixed effects relative to the correctly specified analysis, because the random effects were overestimated, leading to overestimated standard errors of the fixed effects. Conversely, they found that overspecification of the level-1 matrix was associated with higher power to detect the fixed effects relative to the correctly specified analysis, because the random effects were

underestimated, leading to underestimated standard errors of the fixed effects. This is in contrast to the example reported by Singer and Willett (2003), in which using an unstructured level-1 matrix rather than a more parsimonious structure was associated with lower power to detect the fixed effects. However, Singer and Willett used a single dataset, whereas Kwok et al. drew their conclusion through Monte Carlo simulation. The final recommendation provided by Kwok et al. is to err on the side of overspecification rather than underspecification of the level-1 error matrix.

***Ignoring correlated residuals*.** Some research has focused specifically on the effects of incorrectly ignoring correlated residuals in growth curve models. When such correlations were ignored, Sivo, Fan, and Witta (2005) found that the fixed effects (intercept and linear slope) were generally spared from bias. Conversely, the random effects can suffer from bias; in their study, the intercept variance was overestimated, which in turn led to the intercept-slope correlation being underestimated. This bias became worse as the size of the correlation between errors increased. These authors recommend that researchers plan in advance, before the analysis stage, to compare models that differ in their level-1 error structure. Supplementing Monte Carlo simulation with analytical work has shown that ignoring residual correlations in a structural equation model (framed as a hypothetical two-wave longitudinal study) is associated with an overestimation of the latent variances, residual variances, and the regression weight between the two latent factors (Reddy, 1992). The factor loadings can be underestimated as well; though not discussed in the original study, this underestimation of factor loadings could potentially come into play in latent-basis (also known as fully latent) growth curve models, where the functional form of the change over time is estimated from the data

through the factor loadings (e.g., Preacher, Wichman, MacCallum, & Briggs, 2008; Sterba, 2014).

Ignoring heterogeneous variances. A few studies have focused explicitly on the modeling of heterogeneous variances, namely variances that vary across groups or time points. Results across several studies suggest that accommodating heterogeneous variances can improve predictions for future observations (for example, in the context of presidential elections in each state, and of feedlot selection in agricultural research; Boscardin & Gelman, 1996; Cernicchiaro, Renter, Xiang, White, & Bello, 2013). In the context of growth mixture models, where different subgroups (or classes) of individuals differ in their change over time (e.g., Preacher et al., 2008), Enders and Tofighi (2008) found that incorrectly assuming that the residual variances do not vary across classes can lead to biased and less precise parameter estimates, and those difficulties increase as the misspecification becomes larger (i.e., as the residual variances become more discrepant across classes). In sum, assuming that residual variances do not vary across groups or time when such variances are heterogeneous in the population can reduce accuracy of predictions and yield biased parameter estimates.

Misspecifying the level-1 matrix through information criteria. Previous simulation work has also explored the consequences of misspecifying the level-1 matrix when such misspecification is "well intentioned," in that the wrong structure has been selected because it was the structure that provided the best model fit. Gomez, Schaajle, and Fellingham (2005) showed through simulation that when the incorrect level-1 error structure is selected through the AIC or BIC (two information criteria; see below), the tests of the fixed effects can be conservative (the Type I error rates were systematically

below the nominal level of .05). In a similar study, Vallejo, Ato, and Valdés (2008) have found that when the structure of the level-1 error matrix is misspecified by relying on information criteria in small samples ($n \leq 60$), the test of the fixed effects can also be liberal in some conditions (i.e., the Type I error rate is greater than .05). Either way, these studies show that misspecifying the level-1 error structure can lead to biased tests of the fixed effects, at least when the misspecification was favored by information criteria.

Summary. To summarize the effects of misspecifying the residual (co)variances in growth curve models, I focus on findings from simulation studies. In general, fixed effects tend to be estimated without bias when the level-1 error matrix is misspecified, but the misspecification can affect the test of the fixed effects, generally by yielding less powerful tests of the fixed effects. On the other hand, random effects are frequently affected by the misspecification of the level-1 matrix. More often than not, misspecifying the residual (co)variances is associated with overestimated intercept and slope variances, and underestimated intercept-slope correlation. These biases appear worse when there are few assessments per participant, and if a nonzero autocorrelation between residuals is ignored, then these biases also get worse as the magnitude of the autocorrelation increases.

Detecting Misspecification of Residual (Co)Variances With Time-Unstructured Data

One aim of the current study is to determine whether the fit indexes available with time-unstructured data (deviance and information criteria) are sensitive to the misspecification introduced by ignoring variation in times of assessment when modeling the residual (co)variances. Those fit indexes will also be used to determine whether the

new partially time-unstructured method developed here yields better fit than the standard time-structured analysis, which ignores all variation in times of assessment.

Much previous research on the detection of misspecified residual (co)variances has relied exclusively on the traditional fit indexes available in the SEM framework (such as the root mean square error of approximation [RMSEA], comparative fit index [CFI], or Tucker-Lewis index [TLI]; e.g., Wu & West, 2010). For example, Sivo et al. (2005) found that SEM fit indexes sometimes indicate similar or even *better* fit when a nonzero autocorrelation among errors is ignored. However, with time-unstructured data, only information criteria are available to evaluate the fit of a growth curve model, even when the analysis is conducted in the SEM framework (Sterba, 2014). For this practical reason, I focus only on information criteria in this study.

In this section, I first compare from a theoretical standpoint the two information criteria most often available with growth curve models, the AIC and the BIC (see below). Next, I review recent evidence that suggests that those fit indexes are sensitive to the misspecification introduced in a growth curve model by using a time-structured analysis when the data are time-unstructured. Then, I describe several studies that have evaluated the (lack of) sensitivity of the AIC and BIC to misspecified level-1 matrices in growth curve models.

Two information criteria: AIC and BIC. Two popular information criteria currently provided by MLM and SEM packages are the Akaike Information Criterion (AIC; Akaike, 1974) and the Bayesian Information Criterion (BIC; Schwartz, 1978). Dziak, Coffman, Lanza, and Li (2012) and Vrieze (2012) provide in-depth comparisons of the AIC and BIC. The two indexes only differ in the penalty that they impose for the

number of parameters (i.e., the complexity) of a model, with the BIC imposing a more severe penalty for increasingly complex models than the AIC. Because of this, the two indexes differ in the error they are likely to commit (Dziak et al., 2012): Using the AIC to select a model leads to the risk of choosing too complex a model (i.e., overfitting), whereas using the BIC leads to the risk of choosing too parsimonious a model (i.e., underfitting). When trying to decide among level-1 error structures, Littell et al. (2000) prefer the BIC over the AIC, because their goal is the "parsimonious modeling" of the level-1 matrix.

Vrieze (2012) distinguishes the two indexes in terms of consistency and efficiency. Asymptotically (i.e., in increasingly large samples), the BIC is said to be consistent, meaning that the BIC will converge to its true value and select the true model if the true model is among the candidate models; conversely, the AIC is not consistent. In contrast, the AIC is said to be efficient, meaning that when the true model is not among the candidate models, the AIC will select the model that minimizes the errors of prediction; conversely, the BIC is not efficient. Vrieze recommends using the AIC over the BIC when the true model is likely to be more complex than the candidate models. However, he also concludes by saying that most true psychological models probably are, in fact, complex. Overall, Vrieze's comment seems to suggest that in most situations, AIC may be a better choice than BIC to select among competing models. As discussed below, simulation studies that have compared the AIC and BIC in the context of growth curve models also tend to favor the AIC over the BIC.

Detecting misfit introduced by ignoring variation in times of assessment. One important question is whether the fit indexes that researchers have at their disposal when

conducting a time-unstructured analysis (log-likelihood or deviance and information criteria) are able to detect that a time-unstructured analysis should be preferred over a time-structured analysis. Certainly, these fit indexes have been used in the past to select among competing multilevel models: Sterba (2014) uses the log-likelihood, AIC, and BIC to compare nonlinear growth curve models estimated with time-unstructured data, and she uses likelihood ratio tests to compare nested models, while Harring and Blozis (2014) also use the deviance and AIC to choose an appropriate level-1 error structure in nonlinear multilevel models.

Fortunately, some evidence suggests that the fit indexes are in fact able to detect the misfit introduced by the ignoring of variation in times of assessment. Liu et al. (2015) compared the BIC obtained in piecewise growth curve models when variation in times of assessment are either accommodated (through MLM) or ignored (through SEM). Overall, the BIC was able to distinguish between the time-structured and time-unstructured analyses: the BIC indicated worse fit in the time-structured analysis when larger variations in times of assessment were ignored, whereas the BIC of the time-unstructured analysis was unaffected by the amount of variation in times of assessment. (In the case of the time-unstructured analysis, the BIC was only affected by sample size, increasing in value as sample size increases.) These authors also found that the results were "identically similar" for the AIC; this is in contrast to other research that has found differences in the performance of the AIC and BIC in detecting misspecification of the level-1 error matrix (see section "Detecting Misfit Introduced by Misspecifying the Level-1 Error Matrix" below).

In their simulation study, Aydin et al. (2014) found that, when variation in times of assessment are ignored in a linear growth curve model, the AIC is lower (indicates better fit) when variation in times of assessment is minimal, and the AIC gets larger (indicates worse fit) as variation in times of assessment increases. In other words, the AIC was sensitive enough to distinguish between small and large misspecifications. These authors also performed a sensitivity analysis, comparing time-structured analyses to time-unstructured analyses in the presence of time-unstructured data, and all three fit indexes investigated (the AIC, BIC, and adjusted BIC) indicated better fit (had lower values) with the time-unstructured analyses. This was the case whether or not predictors other than time were included in the model. In short, consistent with Liu et al. (2015), the evidence provided by Aydin et al. suggests that information criteria are sensitive to the ignoring of variation in times of assessment.

Detecting misfit introduced by misspecifying the level-1 error matrix. No prior research has investigated whether fit indexes can detect the misfit introduced by ignoring variability in times of assessment when modeling the level-1 error matrix, but some research has examined the ability of fit indexes to detect misspecification of the level-1 matrix. However, much of this research has focused exclusively on traditional fit indexes—which are not available in a time-unstructured analysis—and has excluded information criteria (see, e.g., Chen, 2007; Heene, Hilbert, Draxler, Ziegler, & Bühner, 2011; Grimm & Widaman, 2010; Hu & Bentler, 1998; Shi, 2009; Sivo et al., 2005; Wu & West, 2010). Fan, Thompson, and Wang (1999) investigated the effect of model misspecification on 10 fit indexes commonly found in the SEM framework, and give two reasons for not including the AIC in their list of fit indexes: (1) the AIC is not easy to

relate to other fit indexes, since it is on a very different scale than the other fit indexes; and (2) the AIC can be used to compare the fit of competing models (whether they are nested or not), but unlike fit indexes found in SEM, the AIC cannot be used to evaluate absolute fit.

In general, the research that looked at the ability of the AIC and BIC to detect misspecification of the level-1 matrix has found that neither index is adequately sensitive to that type of misspecification, but that AIC might perform slightly better than BIC for that task. Keselman, Algina, Kowalchuk, and Wolfinger (1998) compared the success rate of the AIC and BIC in selecting the correct level-1 error structure in growth curve models. Overall, the AIC performed better than the BIC, but neither index showed adequate performance. The average success rate for the AIC was 47%, and the average success rate for the BIC when success was > 0% was 35%; however, in more than half the conditions investigated, the BIC never selected the correct structure. Ferron et al. (2002) report similar (though less extreme) difficulties for the AIC and BIC to select the correct level-1 structure. In their simulation study, Ferron et al. also found that AIC was better than the BIC at selecting the correct structure: On average, the success rate for AIC was 79%, whereas it was 66% for the BIC. Success rate increased with increasing number of time points and sample size, and success tended to be low when there were few time points, particularly when sample size was also small.

In their simulation study, Gomez et al. (2005) report even more extreme difficulties for the AIC and BIC to select the correct level-1 error structure in growth curve models than Ferron et al. (2002) and Keselman et al. (1998). Overall, success rate was low for both indexes. Success rates were higher when sample size was larger and

when the correct level-1 error structure was simpler. However, while the AIC outperformed the BIC when the correct structure was complex (in line with previous research, e.g. Ferron et al., 2002; Keselman et al., 1998; Vallejo et al., 2008), the BIC tended to outperform the AIC when the correct structure was simpler. Vallejo et al. (2008) examined the success rate of the AIC and BIC in selecting the correct level-1 error structure in linear growth curve models in small samples ($n \leq 60$), and as with other studies, the AIC tended to perform better than the BIC, with neither criterion displaying adequate success rates. On average, the AIC led to the correct error structure 68% of the time, whereas the average success rate was 48% for the BIC. Vallejo et al. found that both indexes had higher success rates as the sample size increased.

Summary. Overall, the fit indexes available in time-unstructured analyses seem to be able to distinguish between a time-unstructured analysis and a time-structured analysis when the data are time-unstructured, but these same fit indexes are not as effective for detecting a mismatch between the population and modeled level-1 matrices. When the AIC and BIC are used to select among competing models that differ only in their level-1 matrix, average success rate for both the AIC and BIC tend to be quite low, often below 50%. That being said, in general, the AIC tends to achieve higher success rates than the BIC. Success rate for these fit indexes increases with increasing sample size and number of assessments per participant.

Current Study

There are two main goals to the current study. First, I seek to determine the consequences of ignoring variability in times of assessment when modeling the level-1 error matrix in linear growth curve models in terms of convergence, parameter bias,

power to detect the linear slope, and model fit. Second, I develop and evaluate a novel, partially time-unstructured method that can be used to accommodate some (but not all) of the variation in times of assessment in modeling the level-1 error matrix. This new method is the first method that can be used to retain some of the variation in times of assessment when time is truly continuous (i.e., when most participants are assessed at different times).

The idea behind the novel method developed here is simple. This method makes use of the fact that two *different* time variables can be used in a growth curve model, one that is used as a predictor of the dependent variable (which is used to estimate the fixed and random effects), and one that is used to determine the parameters and/or the dimensions of the level-1 error matrix. In the presence of time-unstructured data, it is often necessary to use two different variables; for example, if each time value is observed only once, then it is impossible to estimate a separate residual variance for each time point, whether or not the researcher believes that the residual variance changes over time. The partially time-unstructured analysis proposed here uses a time variable that retains all of the variation in values as the predictor of the dependent variable, but a modified time variable for the level-1 error matrix where time values are grouped into intervals (i.e., similar time values are lumped together and treated as a single time point). A similar approach has been used before in different contexts in an attempt to reduce the dimension of level-1 matrices (Jamrozik, Kistemaker, Dekkers, & Schaeffer, 1997; Jamrozik & Schaeffer, 1997; Rekaya, Carabano, & Toro, 1999). I extend this approach to the analysis of time-unstructured data, and I formally evaluate the performance of this method through Monte Carlo simulation.

Up to now, the selection of the time variable in modeling the level-1 matrix in growth curve models has received little attention. In fact, there is a common misconception in the methodological literature that the presence of variation in times of assessment is unimportant in modeling the level-1 matrix. In their treatment of the choice of the level-1 matrix in nonlinear growth curve models, Harring and Blozis (2014, p. 373) write that "[w]ithout a loss of generality, we assume a balanced design where the timing of the repeated measures is common to *m* individuals, […]", implying that both the parameters and the dimensions of the level-1 matrix do not depend on whether time values vary across participants. Similarly, Grimm and Widaman (2010) develop two new residual structures based on growth curve reliability, and both structures (invariant reliability over time, linearly-changing reliability over time) depend on time, yet these authors do not address the construction of an appropriate time variable when the data are time-unstructured. Perhaps even more telling, Hox (2010, p. 100) writes in his popular introductory text on multilevel modeling that "[… with] an unstructured model for residual errors across time; *all possible variances and covariances are estimated*." (emphasis added), even though he uses assessment number rather than individual time values to determine the parameters of the level-1 matrix—in other words, his unstructured matrix actually estimates the *minimum* number of variances and covariances that could be estimated with an unstructured matrix, namely, the number resulting from assuming that there are no more times of assessment than the number of assessments. With this study, I hope to bring attention to this common misconception, and I show that the choice of the time variable used to model the residual (co)variances has consequences

that researchers should care about, even when they do not have any substantive interest in the residual (co)variances per se.

Chapter 2

Method

Simulation Design

Table 1 shows the factors that were varied in the simulation. There are three numbers of assessments per person (3, 6, or 9), which affects the dimension of the level-1 error matrix; three sample sizes (50, 200, or 500); two structures for the population and modeled level-1 error matrix (heterogeneous diagonal or first-order autoregressive [AR(1)]); and three levels of misspecification of the level-1 error matrix with respect to time (none, intermediate, or maximal).

Factor	Values
Number of assessments per person	3, 6, 9
Sample size	50, 200, 500
Population & Modeled level-1 matrix	Heterogeneous diagonal, AR(1)
Misspecification with respect to time	None, Intermediate, Maximal

Table 1. Simulation factors.

Population values for the mean intercept and linear slope, intercept and slope variances, intercept and slope covariance, and residual variance were chosen to be representative of published research, and are the same values used in previous research on the analysis of time-unstructured data (Coulombe et al., 2016). Values at time 0 are *T* scores (see also Hertzog, Lindenberger, Ghisletta, & von Oertzen, 2006; Hertzog, von Oertzen, Ghisletta, & Lindenberger, 2008; von Oertzen, Hertzog, Lindenberger, & Ghisletta, 2010). This means that the population mean intercept is 50, and intercept

variance is 100. The mean nonzero slope is 2, corresponding to a change of one fifth of a standard deviation from one assessment to the next (cf. Feingold, 2009). The slope variance was chosen so as to be smaller than the intercept variance and was set to 16, meaning that the intercept variance is 6.25 times larger than the slope variance. The covariance between the intercept and slope was set to 12, which corresponds to an intercept-slope correlation of +0.30.

Regardless of whether the level-1 error matrix was heterogeneous diagonal or AR(1), the residual variance was set to 100 at time 0, implying a low growth curve reliability of 50%, reflecting levels commonly found in the growth curve literature. When the level-1 error matrix is first-order autoregressive, the residual variance is constant over time, and the correlation between residuals for assessments separated by one unit on the time scale is ρ = +0.5. When the level-1 error matrix is heterogeneous diagonal, the residual variance is set to increase linearly over time, increasing by 3 for each one-unit increase on the time scale (see also Jacqmin-Gadda et al., 2007). With time values varying from 1 to up to 36 when there are nine assessments per person (see "Analysis of Datasets" below), residual variances vary from 100 up to 205.

Data Generation

One thousand datasets were generated in each combination of number of assessments (3 levels), sample size (3 levels), and level-1 error matrix (2 levels), yielding 3×3×2 = 18 different combinations and 1,000×18 = 18,000 different datasets. Each dataset was then analyzed in three different ways by varying the extent of the misspecification of the level-1 error matrix (bottom row of Table 1; also see next section).

Data were generated in multivariate fashion, one vector of scores at a time, as was done in Coulombe et al. (2016). In particular, data were generated according to the structural equation modeling approach to growth curve modeling, using the following equations:

$$\boldsymbol{y_i} = \boldsymbol{\Lambda_{yi}\eta_i} + \boldsymbol{\varepsilon_i} \tag{7}$$

where

$$\boldsymbol{\eta_i} = \boldsymbol{\alpha} + \boldsymbol{\zeta_i} \tag{8}$$

$$\boldsymbol{\varepsilon_i} \sim MVN(\mathbf{0}, \boldsymbol{\Theta_\varepsilon});\ \boldsymbol{\zeta_i} \sim MVN(\mathbf{0}, \boldsymbol{\Psi}) \tag{9}$$

In this set-up, with *t* assessments per person and an intercept and slope factor, $\boldsymbol{y_i}$ is the $(t \times 1)$ vector of scores; $\boldsymbol{\Lambda_{yi}}$ is a $(t \times 2)$ matrix of factor loadings, where the *i* subscript indicates that the times of assessment can vary across individuals; $\boldsymbol{\eta_i}$ is a (2×1) vector of factor scores for the intercept and linear slope factors obtained from the sum of $\boldsymbol{\alpha}$, a (2×1) vector of factor means, and $\boldsymbol{\zeta_i}$, a (2×1) vector of factor-level residuals; and $\boldsymbol{\varepsilon_i}$ is a $(t \times 1)$ vector of occasion-specific residuals. For example, the matrices used when the level-1 error matrix is heterogeneous diagonal and an individual is assessed at times 1, 7, and 9 (with the time variable centered at time = 1) are the following:

$$\boldsymbol{\Lambda_{yi}} = \begin{bmatrix} 1 & 0 \\ 1 & 6 \\ 1 & 8 \end{bmatrix}$$

$$\boldsymbol{\alpha} = \begin{bmatrix} 50 \\ 2 \end{bmatrix}$$

$$\boldsymbol{\varepsilon_i} \sim MVN(\mathbf{0}, \boldsymbol{\Theta_\varepsilon}), \text{ where } \boldsymbol{\Theta_\varepsilon} = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 118 & 0 \\ 0 & 0 & 124 \end{bmatrix}$$

$$\boldsymbol{\zeta_i} \sim MVN(\mathbf{0}, \boldsymbol{\Psi}), \text{ where } \boldsymbol{\Psi} = \begin{bmatrix} 100 & 12 \\ 12 & 16 \end{bmatrix}$$

When the level-1 error matrix is AR(1), Figure 2 shows that the off-diagonal entries of the matrix are obtained by raising ρ = +0.5 to a power equal to the difference in times of assessment (e.g., 7-1 = 6) and multiplying by the residual variance (here, 100). In this example, this means that the corresponding level-1 error matrix is:

$$\boldsymbol{\Theta}_{\boldsymbol{\varepsilon}} = \begin{bmatrix} 100 & 1.5625 & 0.390625 \\ 1.5625 & 100 & 25 \\ 0.390625 & 25 & 100 \end{bmatrix}$$

All datasets were generated and analyzed using R v3.2.2 (R Core Team, 2015). The creation of AR(1) matrices was achieved using the package CVTuningCov v1.0 (Wang, 2014), and sampling from multivariate normal distributions was achieved using the package MASS v7.3-43 (Venables & Ripley, 2002). In each dataset, every individual has the same number of assessments, with no missing data.

Listing of the computer code used to generate and analyze the data is presented in the Appendix.

Analysis of Datasets

Each dataset was analyzed with three different levels of misspecification of the level-1 matrix (see bottom row in Table 1): (1) with no misspecification, where the specific time values for the assessments were used in estimating the parameters of the level-1 matrix (yielding a fully time-unstructured analysis); (2) with intermediate misspecification, where the specific time values for the assessments were lumped two by two to estimate the parameters of the level-1 matrix (yielding a partially time-unstructured analysis with respect to the level-1 matrix); and (3) with maximal misspecification, where the assessment numbers rather than time values were used to estimate the parameters of the level-1 matrix (yielding a time-structured analysis with respect to the level-1 matrix). Table 2 shows the resulting values used in modeling the

level-1 error matrix in each of the three misspecification conditions. As shown in Table 2, each wave of assessments spans 4 units on the time scale, with each participant being assessed at one of the four times at each wave. As the level of the misspecification of the level-1 matrix is increased (going from left to right in Table 2), an increasing number of specific time values are assumed to be the same.

Assessment Number	Time Values	Value Used in Each Misspecification Category		
		No Misspecification	**Intermediate Misspecification**	**Maximal Misspecification**
1	1	1	1	1
	2	2	1	1
	3	3	2	1
	4	4	2	1
2	5	5	3	2
	6	6	3	2
	7	7	4	2
	8	8	4	2
3	9	9	5	3
	10	10	5	3
	11	11	6	3
	12	12	6	3
4	13	13	7	4
	14	14	7	4
	15	15	8	4
	16	16	8	4
5	17	17	9	5
	18	18	9	5
	19	19	10	5
	20	20	10	5
6	21	21	11	6
	22	22	11	6
	23	23	12	6
	24	24	12	6
7	25	25	13	7
	26	26	13	7
	27	27	14	7
	28	28	14	7
8	29	29	15	8
	30	30	15	8
	31	31	16	8
	32	32	16	8
9	33	33	17	9
	34	34	17	9
	35	35	18	9
	36	36	18	9

Table 2. Time values used in estimating the level-1 error matrix in each misspecification condition.

Figure 1a-c shows the posited level-1 matrices in the no-misspecification, intermediation-misspecification, and maximal-misspecification conditions when there are three assessments per person and the structure of the matrix is heterogeneous diagonal. Figure 2a-c shows the same matrices when the structure is AR(1). As shown in Figures 1-2, going from (a) no misspecification to (b) intermediate misspecification to (c) maximal misspecification, an increasing amount of individual differences in times of assessment is ignored in modeling the level-1 error matrix. When the matrix is heterogeneous diagonal (Figure 1), increasing the extent of the misspecification of the level-1 matrix also decreases the number of parameters (i.e., the number of residual variances) to be estimated. Conversely, when the matrix is AR(1) (Figure 2), the number of estimated parameters (2) remains constant across levels of misspecification.

(a) No misspecification

		Wave 1				Wave 2				Wave 3			
	Time	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Wave 1	**1**	σ_1^2											
	2	0	σ_2^2										
	3	0	0	σ_3^2									
	4	0	0	0	σ_4^2								
Wave 2	**5**	0	0	0	0	σ_5^2							
	6	0	0	0	0	0	σ_6^2						
	7	0	0	0	0	0	0	σ_7^2					
	8	0	0	0	0	0	0	0	σ_8^2				
Wave 3	**9**	0	0	0	0	0	0	0	0	σ_9^2			
	10	0	0	0	0	0	0	0	0	0	σ_{10}^2		
	11	0	0	0	0	0	0	0	0	0	0	σ_{11}^2	
	12	0	0	0	0	0	0	0	0	0	0	0	σ_{12}^2

(b) Intermediate misspecification

		Wave 1		Wave 2		Wave 3	
	Time	**1-2**	**3-4**	**5-6**	**7-8**	**9-10**	**11-12**
Wave 1	**1-2**	σ_1^2					
	3-4	0	σ_2^2				
Wave 2	**5-6**	0	0	σ_3^2			
	7-8	0	0	0	σ_4^2		
Wave 3	**9-10**	0	0	0	0	σ_5^2	
	11-12	0	0	0	0	0	σ_6^2

(c) Maximal misspecification

		Wave 1	Wave 2	Wave 3
	Time	**1-4**	**5-8**	**9-12**
Wave 1	**1-4**	σ_1^2		
Wave 2	**5-8**	0	σ_2^2	
Wave 3	**9-12**	0	0	σ_3^2

Figure 1. Estimated heterogeneous diagonal error matrix with three assessments per person with (a) no misspecification, (b) intermediate misspecification, and (c) maximal misspecification. Non-numerical entries are parameters to be estimated. Subscripts are index (row and column) numbers and not time values.

(a) No misspecification

		Wave 1				Wave 2				Wave 3			
	Time	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Wave 1	**1**	σ^2											
	2	$\rho^1\sigma^2$	σ^2										
	3	$\rho^2\sigma^2$	$\rho^1\sigma^2$	σ^2									
	4	$\rho^3\sigma^2$	$\rho^2\sigma^2$	$\rho^1\sigma^2$	σ^2								
Wave 2	**5**	$\rho^4\sigma^2$	$\rho^3\sigma^2$	$\rho^2\sigma^2$	$\rho^1\sigma^2$	σ^2							
	6	$\rho^5\sigma^2$	$\rho^4\sigma^2$	$\rho^3\sigma^2$	$\rho^2\sigma^2$	$\rho^1\sigma^2$	σ^2						
	7	$\rho^6\sigma^2$	$\rho^5\sigma^2$	$\rho^4\sigma^2$	$\rho^3\sigma^2$	$\rho^2\sigma^2$	$\rho^1\sigma^2$	σ^2					
	8	$\rho^7\sigma^2$	$\rho^6\sigma^2$	$\rho^5\sigma^2$	$\rho^4\sigma^2$	$\rho^3\sigma^2$	$\rho^2\sigma^2$	$\rho^1\sigma^2$	σ^2				
Wave 3	**9**	$\rho^8\sigma^2$	$\rho^7\sigma^2$	$\rho^6\sigma^2$	$\rho^5\sigma^2$	$\rho^4\sigma^2$	$\rho^3\sigma^2$	$\rho^2\sigma^2$	$\rho^1\sigma^2$	σ^2			
	10	$\rho^9\sigma^2$	$\rho^8\sigma^2$	$\rho^7\sigma^2$	$\rho^6\sigma^2$	$\rho^5\sigma^2$	$\rho^4\sigma^2$	$\rho^3\sigma^2$	$\rho^2\sigma^2$	$\rho^1\sigma^2$	σ^2		
	11	$\rho^{10}\sigma^2$	$\rho^9\sigma^2$	$\rho^8\sigma^2$	$\rho^7\sigma^2$	$\rho^6\sigma^2$	$\rho^5\sigma^2$	$\rho^4\sigma^2$	$\rho^3\sigma^2$	$\rho^2\sigma^2$	$\rho^1\sigma^2$	σ^2	
	12	$\rho^{11}\sigma^2$	$\rho^{10}\sigma^2$	$\rho^9\sigma^2$	$\rho^8\sigma^2$	$\rho^7\sigma^2$	$\rho^6\sigma^2$	$\rho^5\sigma^2$	$\rho^4\sigma^2$	$\rho^3\sigma^2$	$\rho^2\sigma^2$	$\rho^1\sigma^2$	σ^2

(b) Intermediate misspecification

		Wave 1		Wave 2		Wave 3	
	Time	**1-2**	**3-4**	**5-6**	**7-8**	**9-10**	**11-12**
Wave 1	**1-2**	σ^2					
	3-4	$\rho^1\sigma^2$	σ^2				
Wave 2	**5-6**	$\rho^2\sigma^2$	$\rho^1\sigma^2$	σ^2			
	7-8	$\rho^3\sigma^2$	$\rho^2\sigma^2$	$\rho^1\sigma^2$	σ^2		
Wave 3	**9-10**	$\rho^4\sigma^2$	$\rho^3\sigma^2$	$\rho^2\sigma^2$	$\rho^1\sigma^2$	σ^2	
	11-12	$\rho^5\sigma^2$	$\rho^4\sigma^2$	$\rho^3\sigma^2$	$\rho^2\sigma^2$	$\rho^1\sigma^2$	σ^2

(c) Maximal misspecification

		Wave 1	Wave 2	Wave 3
	Time	**1-4**	**5-8**	**9-12**
Wave 1	**1-4**	σ^2		
Wave 2	**5-8**	$\rho^1\sigma^2$	σ^2	
Wave 3	**9-12**	$\rho^2\sigma^2$	$\rho^1\sigma^2$	σ^2

Figure 2. Estimated AR(1) error matrix with three assessments per person with (a) no misspecification, (b) intermediate misspecification, and (c) maximal misspecification. In each case, two parameters are to be estimated (ρ and σ). Subscripts are index (row and column) numbers and not time values.

All analyses were performed using the R package nlme v3.1-121 (Pinheiro, Bates, DebRoy, Sarkar, & R Core Team, 2015). Only the level-1 error matrix was misspecified; no other component of the analysis was misspecified. This means that while a miscoded time variable was used to estimate the level-1 error matrix in the intermediate- and maximal-misspecification conditions, the correct time variable was used to estimate the fixed effects (mean intercept, mean linear slope) and random effects (intercept and slope variances, intercept-slope correlation).

Outcomes

In each condition of the simulation, I examine the following outcomes:

Proportion of admissible solutions. The proportion of admissible solutions is the proportion of datasets that yield a converging solution with no negative variances, out-of-bounds correlations, or non-positive-definite matrices. Only the datasets that yielded admissible solutions were used in pooling the results within each condition of the simulation. Outcomes were examined in conditions where there were at least 10 admissible solutions.

Parameter bias. Parameter bias is computed as

$$B = \frac{\bar{\hat{\theta}} - \theta}{\theta}$$

where $\bar{\hat{\theta}}$ is the average parameter estimate across replications, and θ is the population value for that parameter. Parameter bias was computed for all estimated parameters, namely: mean intercept and linear slope; intercept and slope variances; intercept and slope correlation; residual variance at each of the 36 time values when the level-1 matrix is heterogeneous diagonal; the residual variance and autocorrelation parameter ρ when the level-1 matrix is AR(1).

Power. Power to detect linear change over time was computed as the proportion of replications in each condition where the *t*-test of the linear slope is significant at an alpha level of .05.

Model fit. In each condition, I examine the proportion of replications in which model fit was better (1) in the no-misspecification analysis relative to the maximal-misspecification analysis; (2) in the no-misspecification analysis relative to the intermediate-misspecification analysis; and (3) in the intermediate-misspecification analysis relative to the maximal-misspecification analysis. Such proportions of better fit were computed for all three fit indexes reported by the package nlme and commonly reported by other multilevel modeling packages: deviance (i.e., the log-likelihood multiplied by -2), the AIC, and the BIC.

Chapter 3

Results

Proportion of Admissible Solutions

Figure 3 shows the convergence rate as a function of sample size (*x* axis), extent of misspecification (line style and points), type of level-1 error matrix (line color), and number of time points (panels). Results were quite different when the level-1 error matrix was autoregressive as opposed to heterogeneous diagonal. First, considering when the level-1 error matrix is AR(1) (blue lines in Figure 3), extent of misspecification of the timing of assessments in the level-1 matrix does not have much of an impact on the convergence rate. When there are few assessments per person (here, 3; leftmost panel in Figure 3), convergence rate is lower, particularly when sample size is small: Convergence rate is slightly above 60% when sample size is 50, and around 90% when sample size is 200. When there are 3 time points and sample size is 500, no misspecification and an intermediate misspecification both yield convergence rates close to 100%, but convergence rate is slightly lower (88%) when misspecification is maximal. When there are 6 or 9 assessments per person (last two columns in Figure 3), convergence rate is always close to 100% when the level-1 matrix is AR(1).

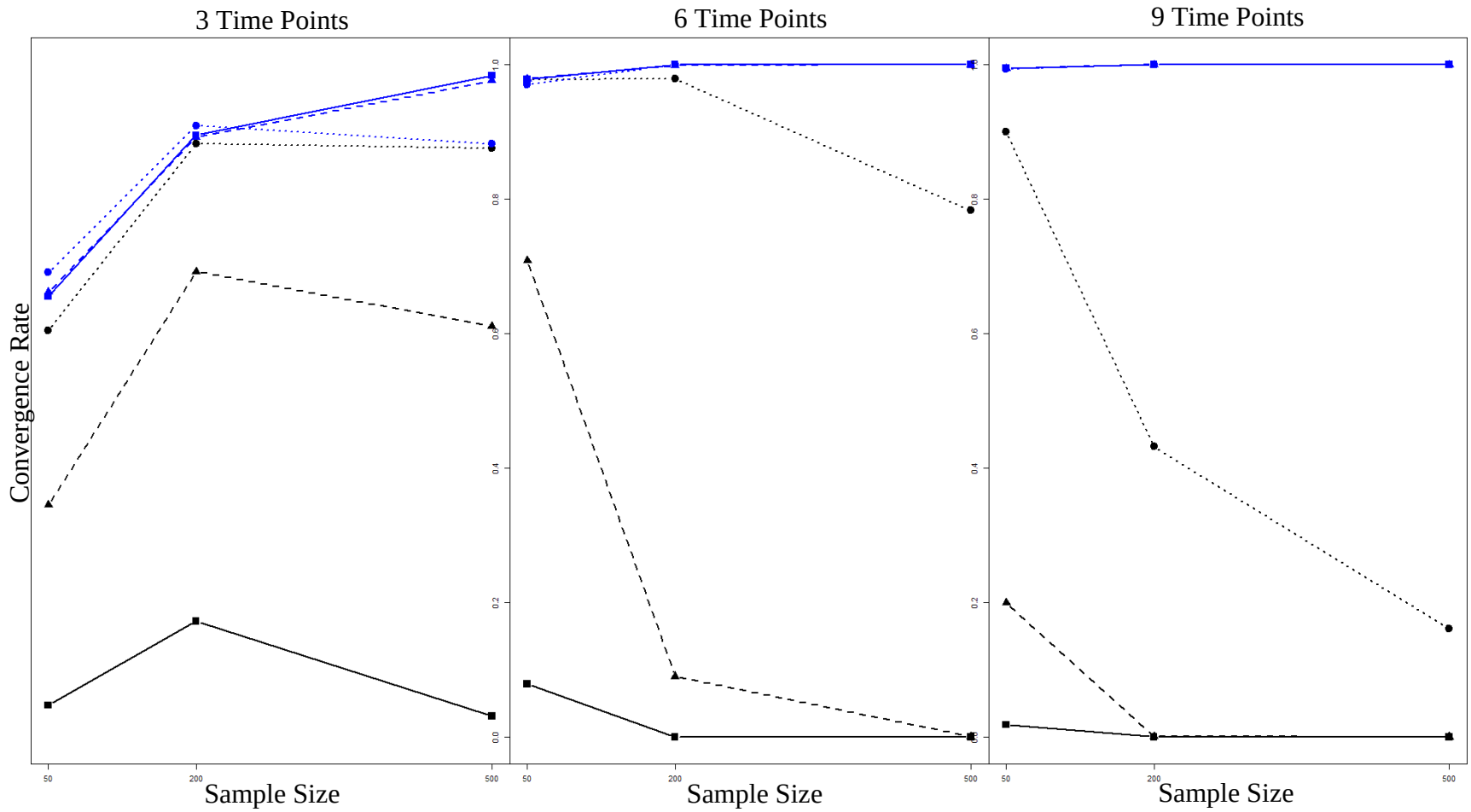

Figure 3. Convergence rate as a function of sample size (*x* axis), extent of misspecification (solid line ■ = no misspecification; dashed line ▲ = intermediate misspecification; dotted line ● = maximal misspecification), type of level-1 error matrix (black line = heterogeneous diagonal; blue = AR(1)), and number of time points (columns).

Conversely, when the level-1 error matrix is heterogeneous diagonal (black lines in Figure 3), the extent of misspecification of the level-1 matrix has a sizeable impact on convergence rate. Across sample sizes and numbers of time points, analyses with maximal misspecification of the heterogeneous diagonal error matrix tend to converge most easily to an admissible solution, followed by analyses with intermediate misspecification, finally by analyses with no misspecification. When there are many time points (6 or 9), convergence rates tend to *decrease* with increasing sample sizes, regardless of the extent of the misspecification of the level-1 error matrix; convergence rates are particularly low when both sample size and number of time points are large (9 time points with $n \geq 200$; see rightmost panel in Figure 3), with convergence rates below 50% regardless of the extent of misspecification of the level-1 matrix. When there is no misspecification of the level-1 error matrix, convergence rate is always low (< 20%), regardless of sample size or number of time points.

There are 7 conditions in which less than 10 replications converged to an admissible solution (i.e., the convergence rate is less than 1%). As shown in Figure 3, all seven instances concern conditions where the level-1 error matrix is heterogeneous diagonal, misspecification is either absent or intermediate, and both sample size (≥ 200) and number of time points (≥ 6) are large. Such frequent convergence problems are not present when the level-1 error matrix is AR(1) (blue lines in Figure 3). Only those conditions with a minimum of 10 converged replications are displayed in the following results.

Parameter Bias

Mean intercept and slope. Figure 4 shows parameter bias (in percent) for the mean intercept (row 1) and mean linear slope (row 2) as a function of sample size (*x* axis), extent of misspecification (line style and points), type of level-1 error matrix (line color), and number of time points (panels). As shown in Figure 4, both the mean intercept and mean slope are generally estimated with minimal bias (< 2%) in conditions with at least 10 admissible replications, with two exceptions in the bottom row of the figure. First, mean slope is underestimated by over 3% in the case of no misspecification with 3 time points and 500 observations; however, as shown in Figure 3, convergence rate in this condition was less than 5% so the bias is estimated with much less precision than in most other conditions. Second, mean slope is overestimated (11% bias) when the level-1 error matrix is heterogeneous diagonal and is correctly specified (no misspecification), and there are 9 time points per person and sample size is 50 (isolated point in rightmost panel of the bottom row in Figure 4); however, this was another case where fewer than 5% of the replications converged (as was shown in Figure 3 above), yielding few and perhaps idiosyncratic analyses. I revisit this issue briefly in the discussion.

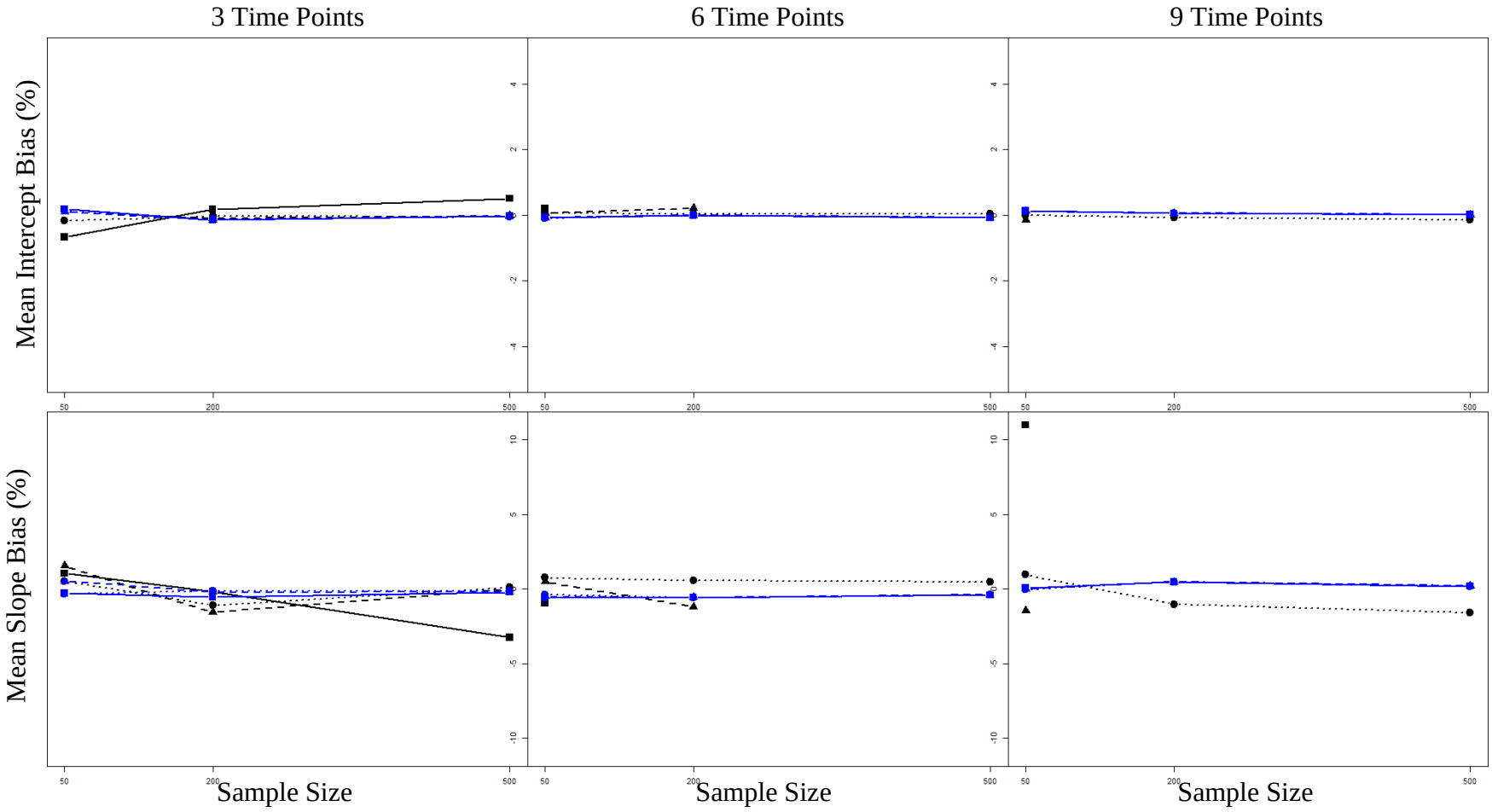

Figure 4. Mean intercept and slope bias (in %) as a function of sample size (*x* axis), extent of misspecification (solid line ■ = no misspecification; dashed line ▲ = intermediate misspecification; dotted line ● = maximal misspecification), type of level-1 error matrix (black line = heterogeneous diagonal; blue = AR(1)), and number of time points (columns).

Intercept and slope variances and correlation. Figure 5 shows parameter bias (in percent) for the intercept variance (row 1), slope variance (row 2), and intercept-slope correlation (row 3) as a function of the manipulated factors. Note that the vertical scale indicating extent of bias has a much wider range in rows 1 and 3 of Figure 5 than in row 2. When the level-1 error matrix is AR(1) (blue lines in Figure 5) and there are only 3 time points, the intercept variance is overestimated when sample size is 50, more so when misspecification is maximal (65% bias) than intermediate or absent (< 20% bias). In those conditions, slope variance is also slightly overestimated, but less so than intercept variance, with bias being less than 3% in the no-misspecification and intermediate-misspecification conditions, and around 9% when misspecification is maximal. With samples larger than 50, only a maximal misspecification leads to an overestimation of the intercept variance (by around 40%) and slope variance (by around 5%) when there are 3 time points. Consequently, as will be considered in more detail below, in this maximal misspecification condition the correlation between intercept and slope is underestimated when there are 3 time points. When there are more than 3 time points, intercept and slope variances are estimated without bias in all conditions.

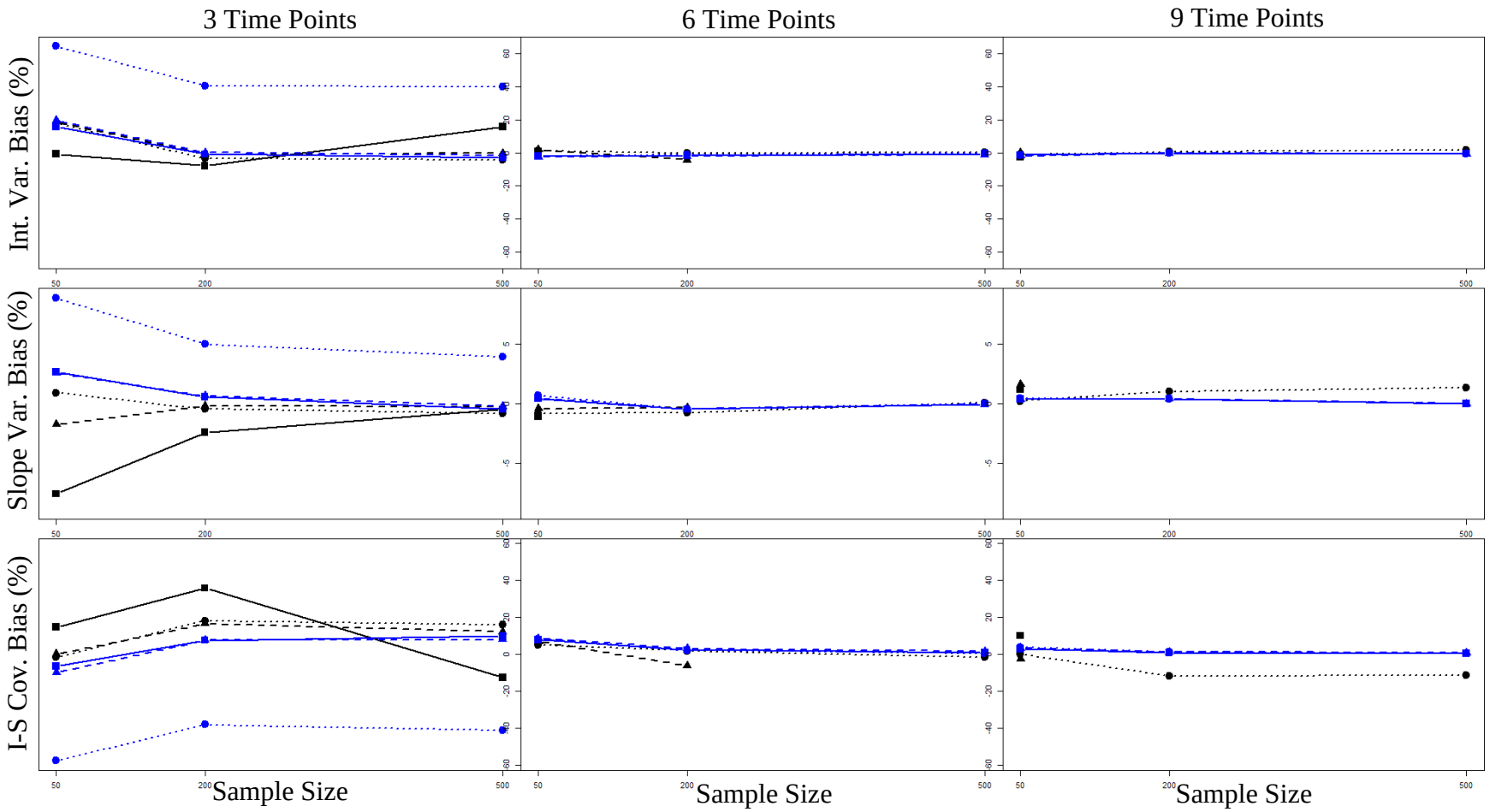

Figure 5. Bias (in %) of intercept and slope variances and correlation as a function of sample size (*x* axis), extent of misspecification (solid line ■ = no misspecification; dashed line ▲ = intermediate misspecification; dotted line ● = maximal misspecification), type of level-1 error matrix (black line = heterogeneous diagonal; blue = AR(1)), and number of time points (columns).

When the level-1 error matrix is heterogeneous diagonal (black lines in Figure 5) and there are 3 time points, intercept variance is slightly overestimated in small samples (n = 50) when misspecification is intermediate or maximal. Conversely, in those conditions intercept variance is generally estimated without much bias when the level-1 error matrix is specified correctly, with the exception of a positive bias (16%) when sample size is 500. As was the case when the level-1 matrix was AR(1), slope variance generally suffers less bias than intercept variance also when the level-1 matrix is heterogeneous diagonal, with a slight negative bias (-8%) when there are 3 time points, sample size is small (n = 50), and the level-1 matrix is correctly specified. As mentioned above, with more than 3 time points both intercept and slope variances are estimated without bias.

Figure 5, row 3 shows parameter bias for the intercept-slope correlation. When the level-1 error matrix is AR(1) (blue lines in Figure 5) and there are 3 time points, the intercept-slope correlation is underestimated when misspecification is maximal (by as much as almost 60% when sample size is 50). Conversely, bias is lesser (< 10%) when misspecification is either intermediate or absent. When there are more than 3 time points and the level-1 matrix is AR(1), the intercept-slope correlation is generally estimated without bias.

When the level-1 error matrix is heterogeneous diagonal (black lines in Figure 5), analyses with either an intermediate or maximal misspecification of the level-1 matrix overestimate the intercept-slope correlation (with bias varying from 12% to 18%) when there are 3 time points and sample size is at least 200. When there are 6 time points, both the intermediate-misspecification and the maximal-misspecification analyses yield a

slightly overestimated intercept-slope correlation (by 5 and 7%, respectively) when sample size is 50, and the analyses with intermediate misspecification slightly underestimate (by 6%) the correlation when sample size is 200. With 9 time points, the analysis with maximal misspecification underestimates the correlation when sample size is 200 and 500 by 12% and 11%, respectively. When the level-1 error matrix is correctly specified (solid lines in Figure 5), the correlation in the analyses that did converge is more often than not overestimated, particularly when there are only 3 time points. Conversely, when there are only 3 time points and sample size is 500, the correct analysis underestimates the correlation (by 13%). However, in some of those conditions—particularly when the heterogeneous diagonal matrix is correctly specified—most of the replications did not converge (as was shown in Figure 3 above).

Residual variances with heterogeneous diagonal matrix. Figure 6 shows the mean residual variance for each of the time values (*x* axis) as a function of number of time points per individual (line color) and sample size (line style) when the level-1 error matrix is correctly specified, with the expected residual variance superimposed (thick red line). Figure 7 shows the corresponding bias (in percent) for the residual variance at each time value.

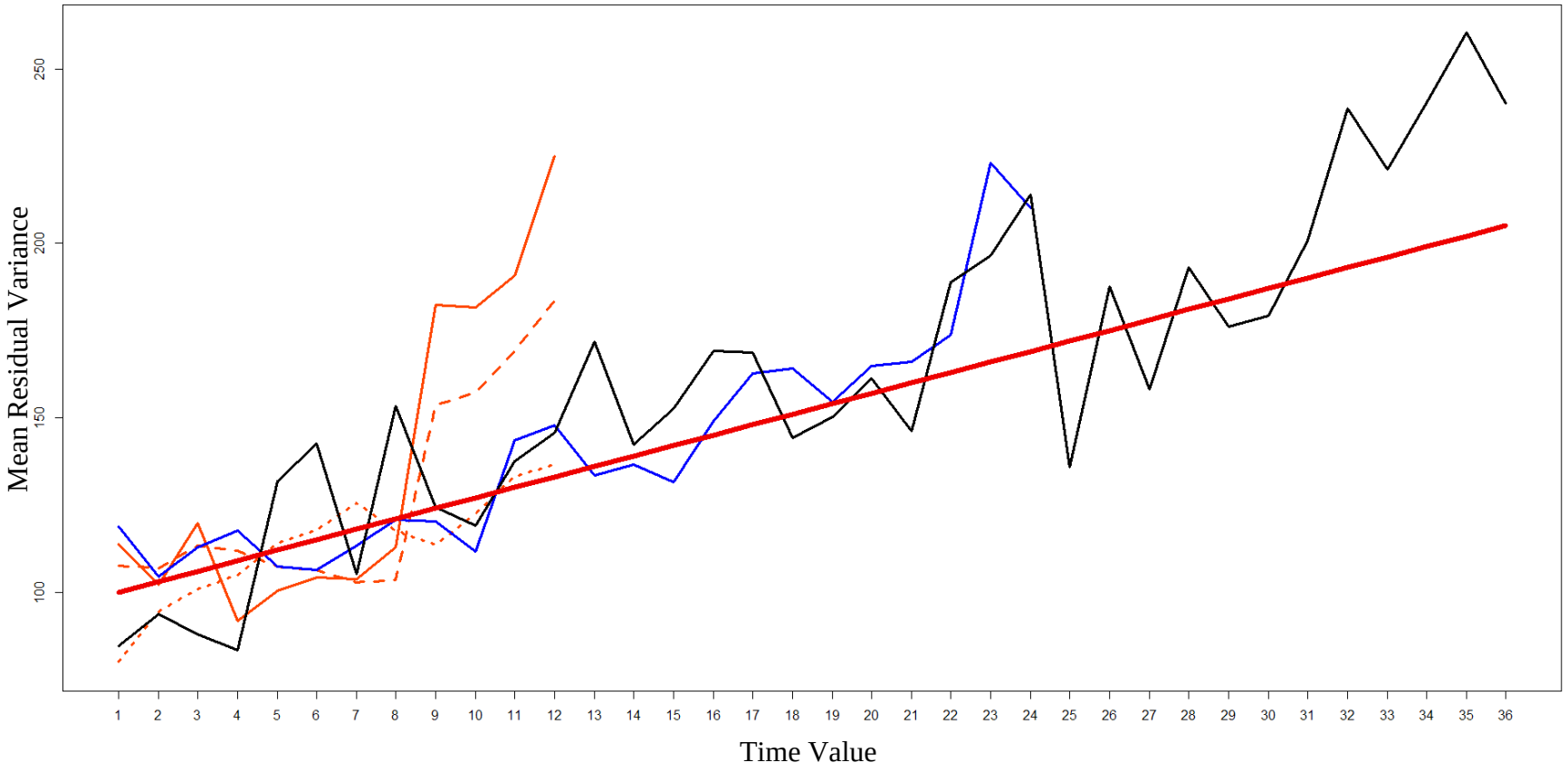

Figure 6. Mean residual variance for each time value (*x* axis) as a function of number of time points (orange line = 3, blue line = 6, black line = 9) and sample size (solid line = 50, dashed line = 200, dotted line = 500) when the heterogeneous diagonal error matrix is correctly specified, with the expected residual variance superimposed (thick red line).

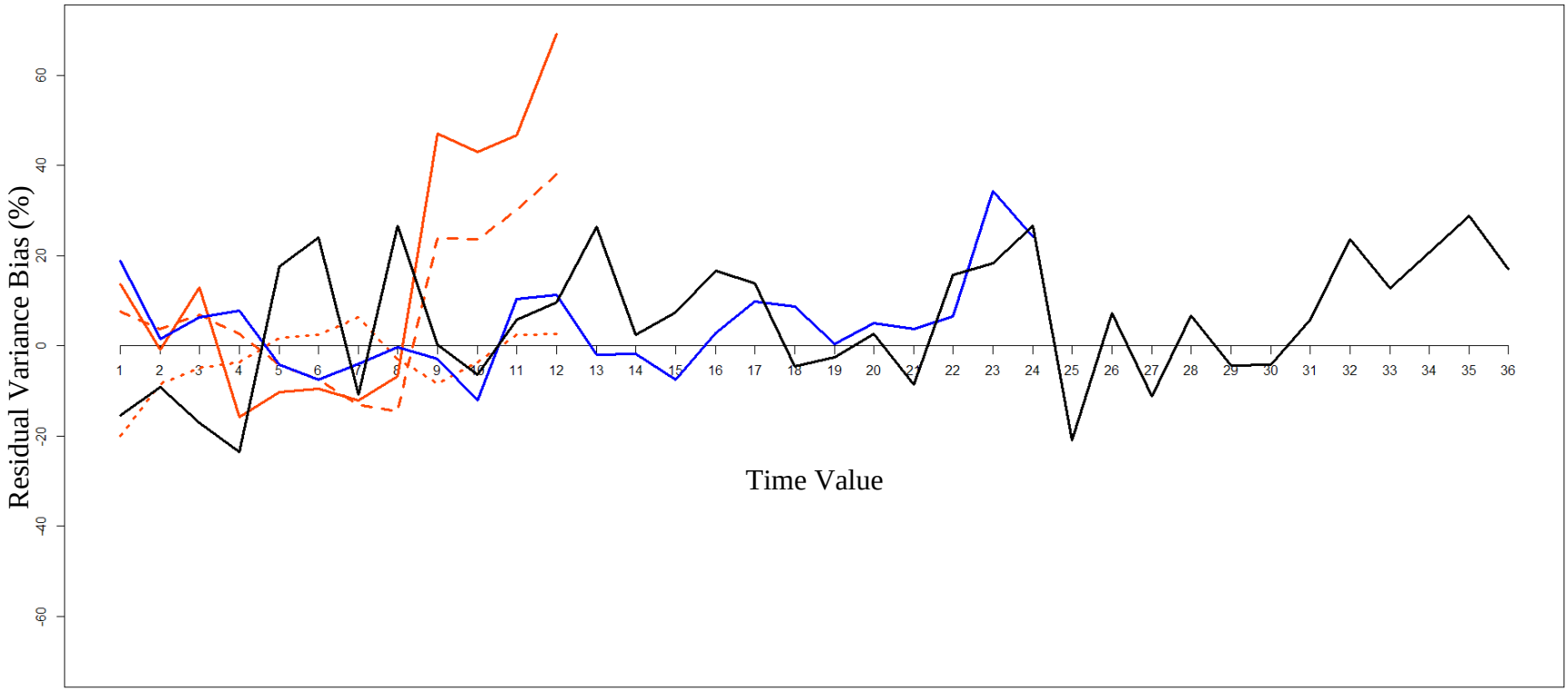

Figure 7. Residual variance bias (in %) for each time value (*x* axis) as a function of number of time points (orange line = 3, blue line = 6, black line = 9) and sample size (solid line = 50, dashed line = 200, dotted line = 500) when the heterogeneous diagonal error matrix is correctly specified.

As shown in Figure 6, residual variances are correctly found to increase over time across all sample sizes and numbers of time points when the heterogeneous diagonal matrix is correctly specified. Figure 7 shows that when there are 3 time points per individual (orange lines), bias moves around 0, except towards the later time values (9-12), where residual variance is overestimated when sample size is less than 500. When there are 6 time points and sample size is 50 (blue line in Figure 7), a similar pattern is observed, where bias hovers around 0 except for the later time values (23-24), where residual variance is overestimated. Finally, when there are 9 time points per individual and sample size is 50 (black line in Figure 7), residual variances are more often than not overestimated. However, as mentioned above and as shown in Figure 3, the majority of replications in this condition did not converge to an admissible solution.

Residual variance and autocorrelation with AR(1) matrix. Figure 8 shows parameter bias (in percent) for the residual variance (row 1) and autocorrelation (row 2) as a function of sample size (*x* axis), extent of misspecification (line style and points), and number of time points (panels) when the matrix is AR(1). As shown in the last two columns of row 1 in Figure 8, the residual variance is estimated without bias when there are more than 3 time points. When there are 3 time points (leftmost column in Figure 8), the residual variance is estimated with minimal bias ($\leq 5\%$) when misspecification is absent or intermediate, but when misspecification is maximal the residual variance is underestimated by 26% when sample size is 50, and by 19% and 20% when sample size is 200 and 500, respectively.

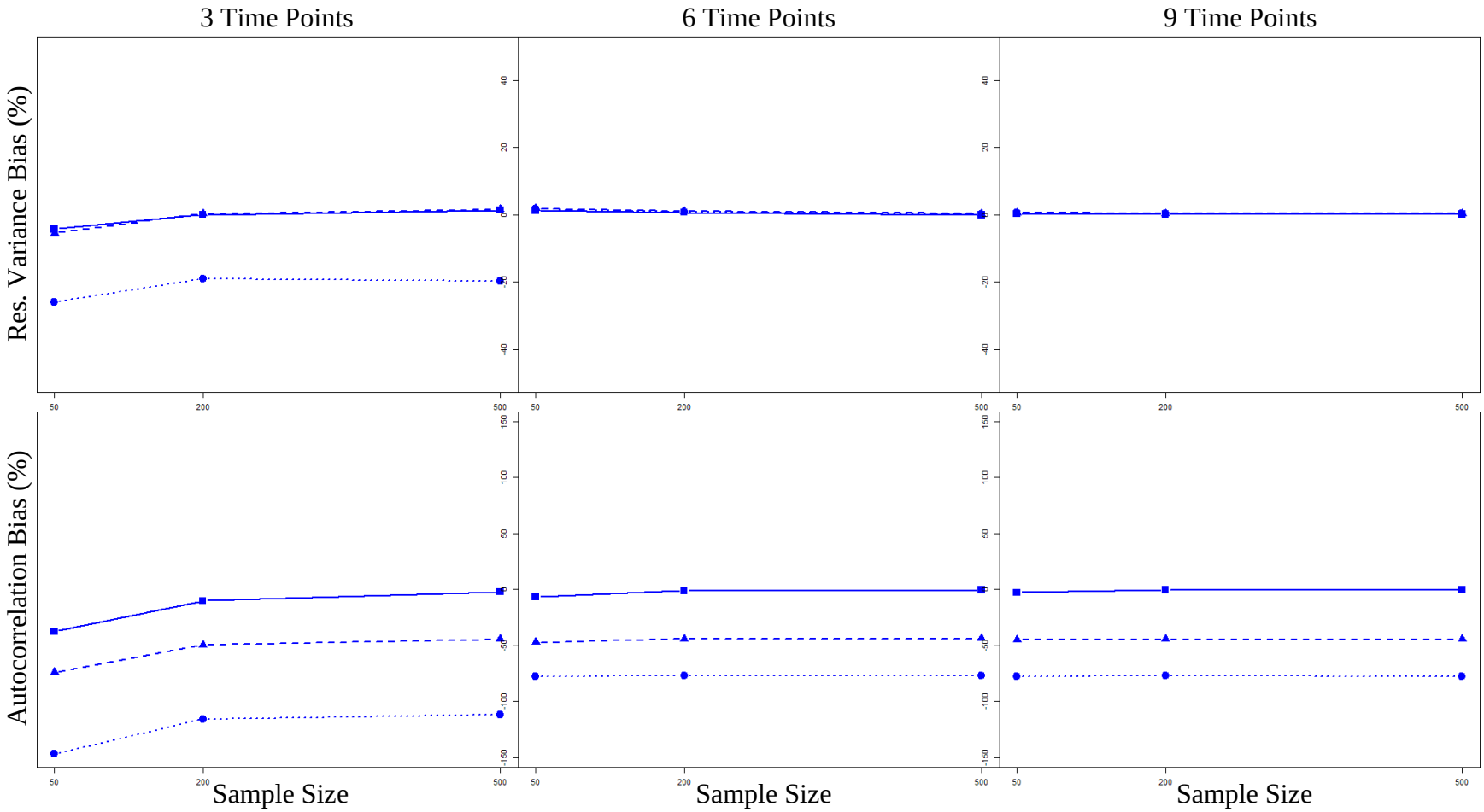

Figure 8. Residual variance and autocorrelation bias (in %) as a function of sample size (*x* axis), extent of misspecification (solid line ■ = no misspecification; dashed line ▲ = intermediate misspecification; dotted line ● = maximal misspecification), and number of time points (columns) when the level-1 error matrix is AR(1).

Figure 8, row 2 shows a large effect of the extent of the misspecification of the level-1 error matrix on the estimate of the autocorrelation parameter in the AR(1) matrix. When the level-1 matrix is correctly specified, the autocorrelation is estimated with minimal bias, except when sample size is 50 and there are either 3 or 6 time points, in which case the autocorrelation is underestimated (by 38% with 3 time points, and by 7% with 6 time points). Conversely, when the misspecification of the level-1 matrix is either intermediate or maximal, the autocorrelation is underestimated, more so when misspecification is maximal rather than intermediate. When misspecification is intermediate, the autocorrelation is underestimated by 74% when there are 3 time points and sample size is 50, and slightly less so (by 44-50%) with other sample sizes and numbers of time points. When misspecification is maximal, the autocorrelation is greatly underestimated across conditions, particularly when there are 3 time points, where bias is greater than 100%; nonetheless, the autocorrelation is also underestimated (by 77-78%) when there are 6 or 9 time points.

Power

Figure 9 shows power to detect the mean linear slope as a function of the manipulated factors. In all conditions investigated, power is always above .89, and power is always 1.00 when sample size is at least 200. When there are 3 time points and sample size is 50, power varies between .89 and .91. When there are 6 time points and sample size is 50, power varies between .94 and .95 when the matrix is heterogeneous diagonal, and power is .92 when the matrix is AR(1). Finally, when there are 9 time points and sample size is 50, power varies between .90 and .94 when the matrix is heterogeneous diagonal, and power is .92 when the matrix is AR(1).

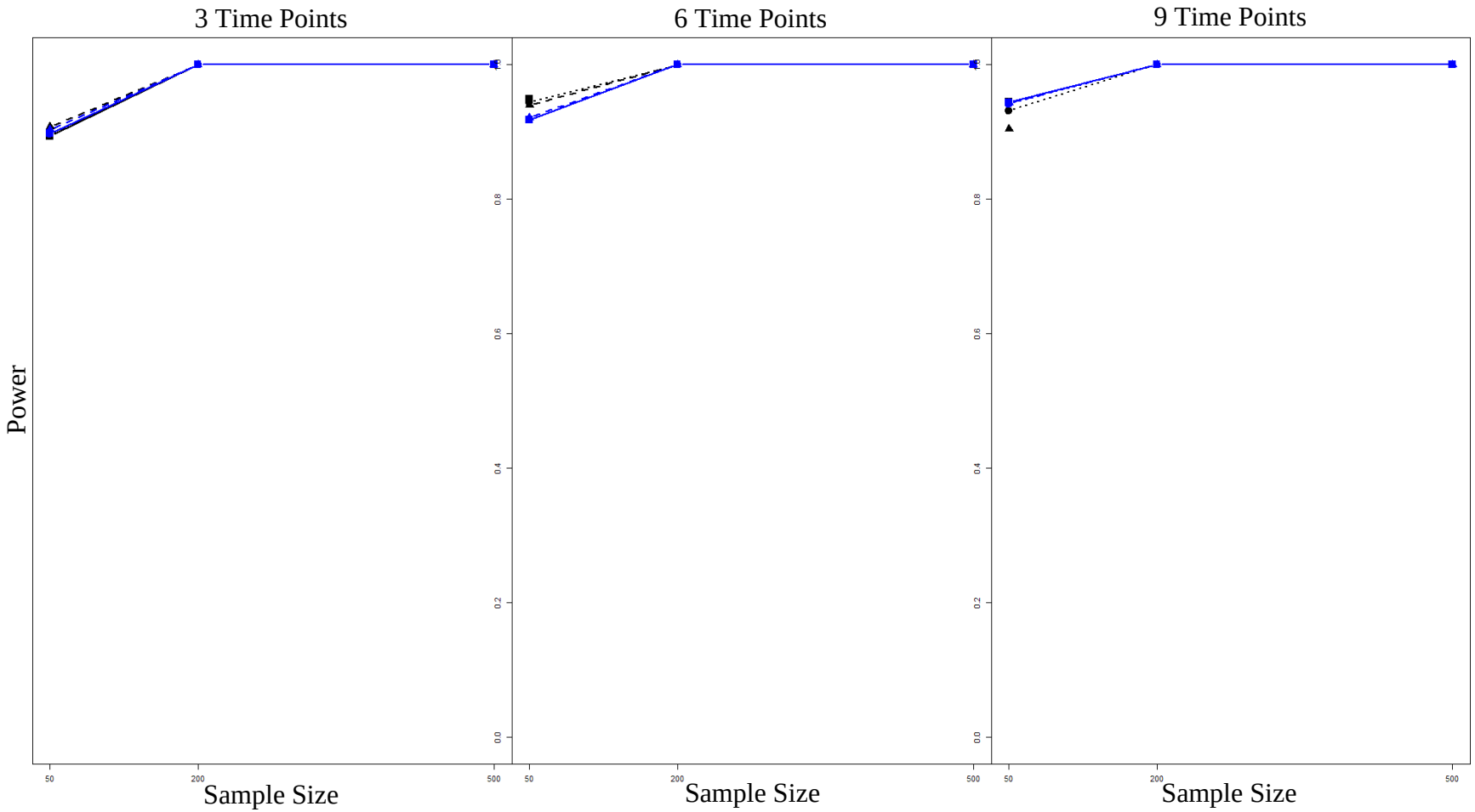

Figure 9. Power to detect the linear slope as a function of sample size (*x* axis), extent of misspecification (solid line ■ = no misspecification; dashed line ▲ = intermediate misspecification; dotted line ● = maximal misspecification), type of level-1 error matrix (black line = heterogeneous diagonal; blue = AR (1)), and number of time points (columns).

Model Fit

Figure 10 shows the proportion of replications in which three fit indexes (deviance, row 1; AIC, row 2; BIC, row 3) indicate better fit when modeling the level-1 error matrix with no misspecification relative to maximal misspecification (solid lines), no misspecification relative to intermediate misspecification (dashed lines), and intermediate misspecification relative to maximal misspecification (dotted lines), as a function of sample size (*x* axis), type of level-1 error matrix (line color), and number of time points (panels). When the level-1 error matrix is heterogeneous diagonal (black lines in Figure 10), two of the three fit indexes (the AIC and BIC; rows 2 and 3) tend to favor the analyses where misspecification relative to the level-1 matrix is *largest*. In other words, the success rate for both the AIC and the BIC is always (well) below 50% across conditions. In fact, the success rate of the BIC is always 0% across conditions; this means that the BIC systematically favors the analysis with the largest misspecification. The AIC selects the correct analysis a maximum of 12% of the time across conditions, except when there are 3 time points, sample size is 500, and the comparison is between the intermediate-misspecification and maximal-misspecification analyses, where the success rate is 32%. When there are 9 time points, the success rate of the AIC is always 0%. In contrast to both the AIC and BIC, the deviance (row 1) always selects the analysis that most closely represents the data-generation process across conditions (i.e., the success rate is 100% in all conditions investigated). Overall, only the deviance can be used to select the most appropriate analysis when the level-1 error matrix is heterogeneous diagonal.

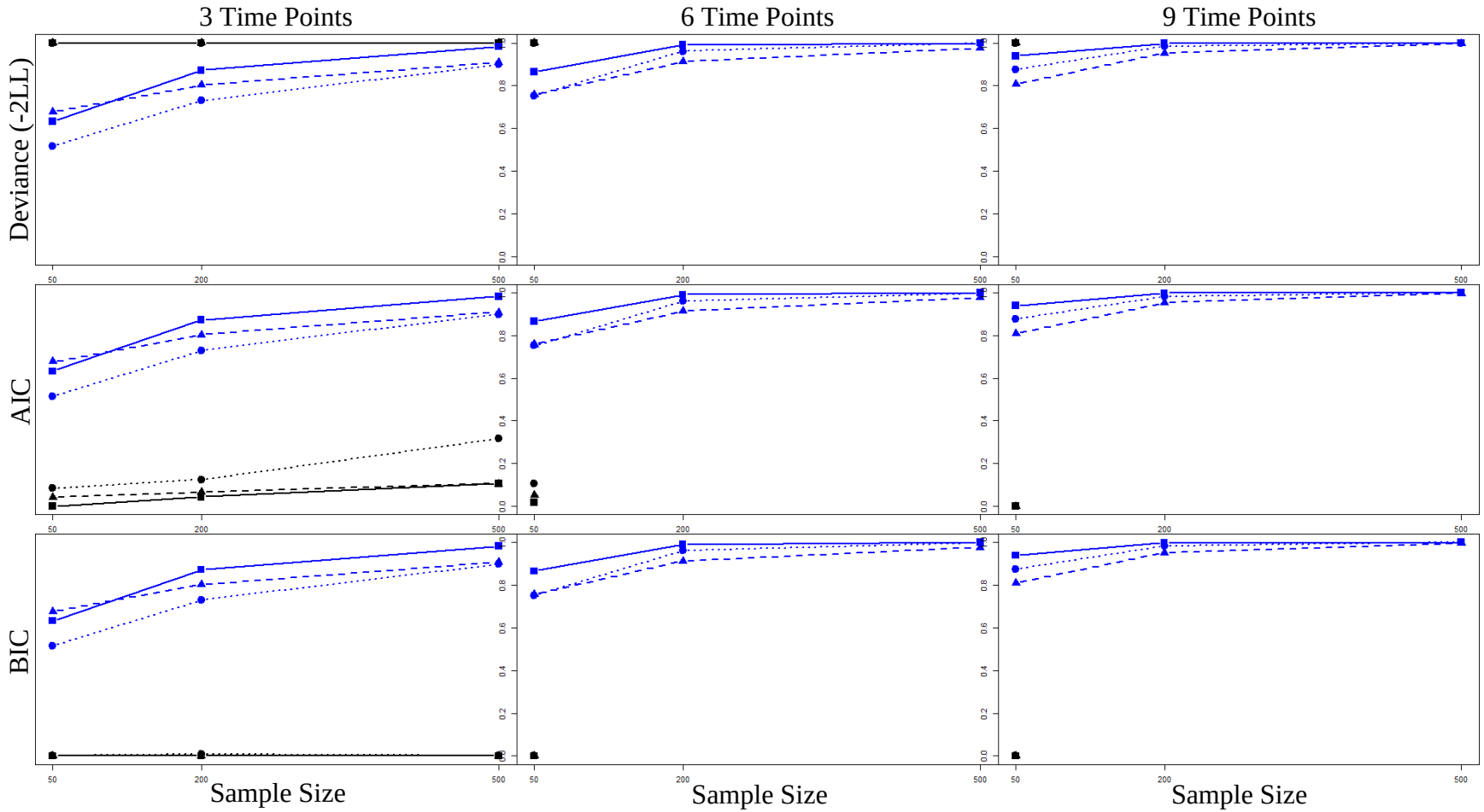

Figure 10. Proportion of replications in which fit indexes select the most appropriate analysis as a function of the misspecification comparison (solid line ■ = none vs. maximal; dashed line ▲ = none vs. intermediate; dotted line ● = intermediate vs. maximal), sample size (*x* axis), type of level-1 error matrix (black line = heterogeneous diagonal; blue = AR(1)), and number of time points (columns).

Unlike conditions where the level-1 error matrix is heterogeneous diagonal, all three fit indexes behave the same across conditions when the matrix is AR(1) (blue lines in Figure 10). Also in contrast to conditions where the matrix is heterogeneous diagonal, all three fit indexes are able to select the analysis that most closely represents the data-generation process a majority of the time (> 50%). The three fit indexes always have a success rate above 63%, except when there are 3 time points, sample size is 50, and the comparison is between the intermediate-misspecification and maximal-misspecification analyses, where the success rate is 51%. When there are at least 6 time points, the success rate of the fit indexes is always over 75%. The success rate of the three fit indexes increases with increasing sample sizes and numbers of time points. In short, while only the deviance can be used to select the appropriate analysis when the level-1 error matrix is heterogeneous diagonal, all three fit indexes can be used when the matrix is AR(1).

Chapter 4

Discussion

There were two main goals to the current study. First, I wanted to establish the consequences of ignoring some or all of the variation in times of assessment when modeling the residual (co)variances, namely the level-1 error matrix, in growth curve modeling. Second, I wanted to evaluate a novel method that attempts to strike a balance between using assessment number and using individual time values when modeling the level-1 matrix. This new method should allow researchers to retain some of the variation in times of assessment even when the data are truly time-unstructured (i.e., the times of assessment are completely different from one person to the next). With those two aims, this study should bridge a gap in the methodological literature. Importantly, this study should also bring attention to the fact that researchers make a choice of how much or how little variation in times of assessment they retain when modeling residual (co)variances in growth curve modeling.

Convergence

Previous studies have noted convergence problems associated with ignoring variation in times of assessment in growth curve modeling (Aydin et al., 2014; Coulombe et al., 2016; Liu et al., 2015) and with misspecifying the structure of the level-1 matrix (Gomez et al., 2005; Gromm & Widaman, 2010). This study extends those findings to situations where variation in times of assessment is ignored when modeling the level-1 matrix. In this study, the effect of ignoring variation in times of assessment when modeling the level-1 matrix on convergence depended on the structure of the matrix. When the matrix was AR(1) (namely, when the residual variance was constant over time

but there was nonzero residual autocorrelation), convergence was generally high regardless of the type of analysis, except that convergence rate was lower when small samples were combined with few time points. Conversely, when the matrix was heterogeneous diagonal (namely, when there were no residual correlations but residual variance was linearly increasing over time), the time-structured analysis converged most often, followed by the partially time-unstructured analysis, finally by the fully time-unstructured analysis. In fact, in many conditions the time-unstructured analysis never converged, despite time having taken only a few discrete values rather than being truly continuous in this study (see Table 2).

The difference between the AR(1) and heterogeneous diagonal structures in terms of estimation difficulties is likely due to the effect of the type of analysis on the number of parameters being estimated. When the matrix is AR(1), accommodating time-unstructured data either partially or fully is not associated with any increase in the number of parameters being estimated relative to a time-structured analysis (see Figure 2). However, when the matrix is heterogeneous diagonal, accommodating time-unstructured data can be associated with a dramatic increase in number of model parameters going from a time-structured to partially time-structured to fully time-unstructured analysis (see Figure 1). In particular, a separate residual variance is estimated for each distinct time value used, so the number of residual variances estimated increases as the number of distinct time values increases. Until research is conducted with more matrix structures (and until appropriate methods are developed to accommodate time-unstructured data with level-1 matrices), researchers can expect to run into

convergence problems if accommodating time-unstructured data when modeling the level-1 matrix means increasing the number of model parameters.

One unexpected finding was the low convergence rate for all three types of analyses when the matrix was heterogeneous diagonal and a large sample was coupled with 9 assessments per individual. On the one hand, the fully and partially time-unstructured analyses estimate a large number of residual variances, which likely contributes to estimation difficulties. On the other hand, the time-structured analysis estimates the minimum number of residual variances given a heterogeneous structure, but the model is misspecified by assuming that the residual variances of assessments collected relatively close in time (here, within 4 units on the time scale; see Table 2) are the same even though residual variance is actually increasing over time. Unfortunately, incorrectly modeling heterogeneous variances as homogeneous (which would be one solution in this situation) can reduce accuracy of predictions (Boscardin & Gelman, 1996; Cernicchiaro et al., 2013) and yield biased estimates of the fixed effects, particularly if the residual variance also varies across groups of participants (Enders & Tofighi, 2008; Jacqmin-Gadda et al., 2007). Therefore, the pervasive estimation difficulties across all three analyses in large samples with many assessments point to the need for more complete methods than the partial method developed here to appropriately accommodate time-unstructured data in level-1 matrices.

Bias

In line with the previous literature reviewed in this study (e.g., Ferron et al., 2002; Shi, 2009; Sivo et al., 2005), misspecifying the level-1 matrix by ignoring variation in times of assessment did not have much effect on parameter bias for the fixed effects

(mean intercept and linear slope). Power to detect the linear slope was also unaffected by the type of analysis and was generally high across conditions. Conversely, and again in line with previous research, this misspecification of the level-1 matrix sometimes yielded biased estimates of the random effects. Bias in the random effects was generally restricted to samples with only 3 assessments per individual, regardless of the structure of the level-1 matrix or the type of analysis. Similar to previous research, when the matrix was AR(1), the time-structured analysis yielded overestimated intercept and slope variances, and the intercept-slope correlation was in turn underestimated. With the correct time-unstructured analysis, bias was much reduced. Interestingly, the new partially time-unstructured analysis also helped reduce bias in the random effects relative to the standard time-structured analysis, and performed as well as the fully time-unstructured analysis.

When the matrix was AR(1), I also investigated bias in the residual variance and autocorrelation. The residual variance was generally estimated without bias regardless of the type of analysis, except when there were only 3 time points, in which case the time-structured analysis underestimated the residual variance at all sample sizes. As was the case for the random effects, the partially time-unstructured analysis performed just as well as the fully time-unstructured analysis in terms of residual variance bias. In sharp contrast, notable bias existed for the autocorrelation parameter, and this bias depended very much on the analysis chosen. Only the fully time-unstructured analysis performed satisfactorily in estimating the residual autocorrelation. Otherwise, there was a clear gradation of the analyses in terms of autocorrelation bias across all sample sizes and numbers of time points, with the partially time-unstructured analysis yielding sizable

bias, and the time-structured analysis yielding even greater bias. This gradation makes sense, because going from fully time-unstructured to partially time-unstructured to time-structured analyses, assessments increasingly far apart in time are treated as though they were still separated by only 1 unit on the time scale. All in all, when the residual autocorrelation is of interest, there is a distinct advantage to accommodating variation in times of assessment as much as possible, through a fully time-unstructured analysis if feasible, or through the partially time-unstructured analysis proposed here if the fully time-unstructured analysis is not possible (e.g., if there are too many distinct time values in the sample, or if the analysis fails to converge). I mention this finding again when I discuss the partially time-unstructured analysis below.

When the matrix was heterogeneous diagonal, some unexpected results were observed in regards to bias in the random effects. The random effects were generally spared from bias (barring some overestimation of the intercept-slope correlation; see Figure 5), except when using the *correct*, fully time-unstructured analysis in conjunction with only 3 assessments per person. In that case, some erratic patterns of bias were observed (see solid lines in Figure 5), for example with bias in both the intercept variance and intercept-slope correlation both increasing *and* decreasing with increasing sample sizes. I attribute these erratic patterns to the extreme convergence difficulties faced in those conditions when using the fully time-unstructured analysis. Very few iterations converged in those conditions (less than 1% convergence rate), so it is possible that the datasets that did lead to converging solutions with the fully time-unstructured analysis converged specifically because they had different parameter estimates, on average, than those datasets that did not lead to converging solutions; in other words, it is possible that

differences in average parameter estimates are the reason why those analyses converged in the first place. In that case, increasing the number of iterations so as to obtain a larger number of converged solutions would still lead to using only datasets that yield, for one reason or another, admissible solutions. Overall, the results regarding conditions in which the vast majority of replications did not converge should be taken lightly, especially since in practice, odds are that the analysis will not even lead to a solution that can be published as is.

Model Fit

Previous research found that information criteria are sensitive to misspecification introduced by ignoring variation in times of assessment (Aydin et al., 2014; Liu et al., 2015), but those same criteria are not as sensitive to misspecification due to using the wrong structure for the level-1 matrix (Ferron et al., 2002; Gomez et al., 2005; Keselman et al., 1998; Vallejo et al., 2008). In the current study, I used the correct matrix structure (AR[1] or heterogeneous diagonal), but I ignored some or all of the variation in times of assessment. Therefore, the information criteria should have been expected to select the correct models, with a preference for the fully time-unstructured analysis, followed by the partially time-unstructured analysis, finally by the time-structured analysis. This is what I found when the matrix is AR(1). When the matrix is AR(1), all three fit indexes behave similarly, and success rate is always greater than 50% for all comparisons (fully time-unstructured vs. partially time-unstructured, fully time-unstructured vs. time-structured, and partially time-unstructured vs. time-structured). As found in previous studies, success rate increases with sample size and number of assessments per participant (e.g., Ferron et al., 2002; Gomez et al., 2005; Vallejo et al., 2008).

In contrast, when the matrix is heterogeneous diagonal, only deviance selects the correct level-1 matrix a majority of the time, with a success rate of 100% in all conditions (as anticipated by Singer & Willett, 2003). As in previous studies (Ferron et al., 2002; Gomez et al., 2005; Keselman et al., 1998; Vallejo et al., 2008), the AIC achieved higher success rates than the BIC in all conditions, but success rate was always low for both AIC and BIC. When the matrix was heterogeneous diagonal, the highest success rate achieved by the AIC in this study was 32%, while the success rate for the BIC was always 0%; this is in line with previous studies, in which the BIC also achieved a 0% success rate in many conditions (Keselman et al., 1998).

The reason why the AIC and BIC behave similarly to the deviance when comparing models with AR(1) matrices but not heterogeneous diagonal matrices has to do with the penalty that the AIC and BIC—but not the deviance—impose for model complexity (i.e., the number of parameters estimated in the model; Dziak et al., 2012; Vrieze, 2012). As was shown in Figure 2, increasing the extent of the misspecification of the level-1 matrix when the matrix is AR(1) does not change the number of parameters, so the comparisons are made between models that differ only in their fit and not in their number of parameters. Therefore, it would be natural to expect that the patterns of success rates for the AIC and BIC would parallel the ones for the deviance when the matrix is AR(1). Conversely, when the matrix is heterogeneous diagonal, increasing the extent of the misspecification simultaneously *decreases* the number of parameters (Figure 1). This means that when the matrix is heterogeneous diagonal, the AIC and BIC are used to compare models that differ in both their fit and their number of parameters, with the model with the smallest misspecification also being the more complex model.

Consequently, it appears that in this study the penalty for model complexity in the AIC and BIC superseded the gain in model fit obtained by accommodating varying times of observation in the level-1 matrix. Future research should examine whether there exists an amount of variation in times of assessment at which information criteria tend to favor a time-unstructured analysis over a time-structured analysis.

Based on the results of this study, recommendations can be derived regarding which fit indexes can be used to distinguish between time-structured and time-unstructured level-1 matrices in growth curve models. When the matrix is AR(1) (more generally, when the competing level-1 matrices have the same number of parameters), then the deviance and information criteria tend to be accurate in detecting the increase in model fit achieved by moving towards a time-unstructured analysis. Conversely, when the matrix is heterogeneous diagonal (more generally, when the competing level-1 matrices differ in their number of parameters), then only the deviance successfully detects the increase in fit achieved by accommodating the time-unstructured data. While the AIC did perform better than the BIC, neither information criterion can be recommended for this purpose when the matrix is heterogeneous diagonal.

Another option, not investigated in this study, to compare some level-1 matrices that differ in the extent to which they accommodate time-unstructured data is the likelihood ratio test. Perhaps surprisingly, the models shown in Figure 1a-c (but not Figure 2a-c) are nested. In particular, when the level-1 matrix is heterogeneous diagonal matrix, the time-structured matrix is nested within the partially time-unstructured matrix, which is itself nested within the fully time-unstructured matrix. This is not immediately apparent in Figure 1 because the matrices have been drawn so as not to repeat parameters

where not needed. However, the smaller matrices can be expanded to illustrate this nesting. Figure 11 shows the time-structured matrix from Figure 1c re-drawn so that its dimensions match the dimensions of the fully time-unstructured matrix from Figure 1a. (The corresponding partially time-unstructured matrix could be re-drawn in the same way.) As is now apparent by comparing Figure 11 and Figure 1a, one can constrain the residual variances belonging to the same wave to equality to obtain a time-structured matrix from the fully time-unstructured matrix. Other researchers have used likelihood ratio tests to guide selection of level-1 matrices in growth curve models (e.g., Ferron et al., 2002; Gomez et al., 2005; Wolfinger, 1993); future research should examine whether likelihood ratio tests can be fruitful in determining whether the gain in model fit in fully or partially time-unstructured level-1 matrices is sufficient to justify the addition of parameters required by these matrices.

		Wave 1				**Wave 2**				**Wave 3**			
	Time	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Wave 1	**1**	σ_1^2											
	2	0	σ_1^2										
	3	0	0	σ_1^2									
	4	0	0	0	σ_1^2								
Wave 2	**5**	0	0	0	0	σ_2^2							
	6	0	0	0	0	0	σ_2^2						
	7	0	0	0	0	0	0	σ_2^2					
	8	0	0	0	0	0	0	0	σ_2^2				
Wave 3	**9**	0	0	0	0	0	0	0	0	σ_3^2			
	10	0	0	0	0	0	0	0	0	0	σ_3^2		
	11	0	0	0	0	0	0	0	0	0	0	σ_3^2	
	12	0	0	0	0	0	0	0	0	0	0	0	σ_3^2

Figure 11. Expanded time-structured matrix from Figure 1c to illustrate that the time-structured matrix is a fully time-unstructured matrix where residual variances belonging to the same wave have been constrained to equality.

One difficulty with the fit indexes examined here is that none of the indexes can be used to assess absolute fit. In other words, the deviance, AIC, and BIC cannot be used to answer the question: Is my model a good representation of the process that generated my sample data? Instead, those indexes can only be used to compare competing models. Indexes of absolute fit have already been developed in the SEM framework, but such fit indexes cannot be computed with time-unstructured data (Sterba, 2014) or in the MLM framework in general. This lacuna points to a need for absolute fit indexes for time-unstructured analyses, which is a long-standing problem in the MLM framework that extends beyond time-unstructured data analysis. Future research should develop such indexes of absolute fit for MLM in general, and time-unstructured analyses specifically.

One promising possibility to obtain such indexes of absolute fit in time-unstructured analyses is to conduct the time-unstructured analysis normally, but also simultaneously conduct the corresponding Bayesian analysis with the sole aim of deriving one or several Bayesian indexes of absolute fit for that analysis. A related approach has already been adopted and seamlessly implemented in software in the past: In several versions of the multilevel modeling package lme4 (Bates, Maechler, Bolker, & Walker, 2015), the standard multilevel analysis was conducted, but Bayesian credible intervals (instead of frequentist confidence intervals) were provided for the random effects; users were not required to have any knowledge of the Bayesian approach to access those intervals. Extending this idea to the assessment of absolute fit in a time-unstructured analysis, the time-unstructured analysis could be conducted normally, but the technique of Bayesian posterior predictive checking (or some other Bayesian index of absolute fit) could be performed simultaneously in the background. With posterior

predictive checking, researchers could obtain a so-called Bayesian *p*-value, with *p*-values between .05 and .95 generally indicating adequate fit, and *p*-values below .05 or above .95 indicating poor fit (e.g., Gelman, Carlin, Stern, & Rubin, 2004; Lynch & Western, 2004; Muthén & Asparouhov, 2012). Posterior predictive checking could also be used to determine which aspects of the sample data (e.g., the minimum value) are correctly or incorrectly reproduced by the model. One potential drawback of this approach is that Bayesian analyses require simulation, which might render the time-unstructured analysis more time-consuming if indexes of absolute fit are requested. The Bayesian approach is beyond the scope of this paper, but Kruschke (2014) provides an accessible introduction for social scientists, and Gelman, Meng, and Stern (1996) and Gelman et al. (2004) provide detailed accounts of posterior predictive checking.

Partially Time-Unstructured Analysis

One main contribution of the present research is to develop a method that allows researchers to retain some of the variation even when the data are truly time-unstructured. The method is simple, and is based on the grouping of similar time values into intervals if a truly time-unstructured level-1 matrix fails or is impossible to implement.

I found in this study that the partially time-unstructured analysis provided advantages relative to the standard time-structured analysis, and often even behaved similarly to the fully time-unstructured analysis. When the matrix was AR(1), the partially time-unstructured analysis yielded more accurate (less biased) estimates of the random effects and of the residual variance than the time-structured analysis, with bias levels similar to the ones obtained in the fully time-unstructured analysis. When the matrix was heterogeneous diagonal, the partially time-unstructured analysis greatly aided

convergence of the analysis relative to the fully time-unstructured analysis. Such advantages become particularly important when time is truly continuous rather than merely discrete as it was in this study, because in those situations the fully time-unstructured matrices presented here are not identified and simply cannot be implemented. Because time was not truly continuous here, the advantages of the partially time-unstructured analysis may have been underestimated in this study.

The partially time-unstructured analysis as developed in this study is not without its drawbacks, however. When the matrix was AR(1), there was a marked difference in autocorrelation bias between the three analyses, with the partially time-unstructured analysis improving on the time-structured analysis, yet without achieving the ideal performance of the fully time-unstructured analysis. In addition, when the matrix was heterogeneous diagonal, *all* three analyses displayed unacceptably low convergence rates in large samples with many time points. I do not want to recommend that researchers reduce the sample size or number of assessments in their study, or that they attempt at all costs to assess participants at exactly the same time. Instead, in my view, these difficulties point to a need for the development of (non-partial) methods that will allow researchers to achieve fully time-unstructured analyses even when time is truly continuous.

At least two methods are possible to achieve fully time-structured analyses even when time is truly continuous, but such methods have yet to be evaluated formally. The first method extends the Bayesian structural equation modeling approach (Muthén & Asparouhov, 2012), originally developed in the context of confirmatory factor analysis, to multilevel modeling. In this method, the continuous time variable is used to determine

the dimensions of the level-1 matrix, but small-variance prior distributions with zero mean are imposed on the entries of the matrix to render the matrix identified. One caveat with this method is that the posterior distributions of the underidentified (co)variances will resemble the prior distribution. Yet another caveat with this method is that time values might still need to be lumped into intervals. However, this method has yet to be formally described and implemented in growth curve models.

The second method to achieve a fully time-unstructured analysis when time is truly continuous involves explicitly modeling the residual (co)variances as a function of the continuous time variable. Previous research has used linear models to link the logarithm of the level-1 residual variance—but not the covariances—to linear combinations of predictors (e.g., Bello, Steibel, & Tempelman, 2010; Cernicchiaro et al., 2013; Foulley, Quaas, & d'Arnoldi, 1998; Robert-Granié, Heude, & Foulley, 2002; Verbyla, 1993). Conceivably, such models could be extended so as to predict residual covariances in addition to the residual variances, and to include the continuous time variable as a predictor of the residual (co)variances. One potential difficulty with this approach is the possibility of predicting covariances that yield correlations that are out of bounds (> 1 or < -1) once standardized. However, this second method too has yet to be evaluated formally and implemented in growth curve models.

Regardless of the specific choice of the method used to accommodate time-unstructured data in the level-1 matrix, this study highlights the fact that analysts get to choose which time variable they use in modeling the residual (co)variances in a growth curve model, a choice that has been largely misconstrued as automatic up to now. Because the time variable used to construct the level-1 matrix is a choice, that choice

should be reported by researchers. A straightforward way to describe the level-1 matrix to readers is to simply show the full matrix, as I did in Figures 1 and 2. Another standard way to communicate the form of a growth curve model, at least in the SEM framework, is through path diagrams (see, e.g., Kline, 2015). Currently, there exists no standard notation for residual (co)variances based on time values other than assessment number. Figure 12 shows one such possible notation for a growth curve model with 3 time points, with definition variables to define the factor loadings of the linear slope, and with heterogeneous variances (in red). Here, the resulting matrix is the partially time-unstructured heterogeneous diagonal matrix shown in Figure 1b. Figure 12 makes it clear that overall, participants contributed up to 3 scores on the dependent variable, but that 2 residual variances rather than 1 are estimated for each measurement occasion. Notations other than the one shown in Figure 12 are possible, and could be extended to include nonzero residual correlations as well. Note that I do not use diamonds to indicate that time values stored in variables are used to define the heterogeneous variances, because diamonds are the notation proposed by Mehta and West (2000) to represent definition variables, whose values take the place of (non-estimated) parameters in the model; time values in (partially) time-unstructured matrices are used to determine the dimensions of the matrix and the parameters to be estimated, but not to fix parameters to given values.

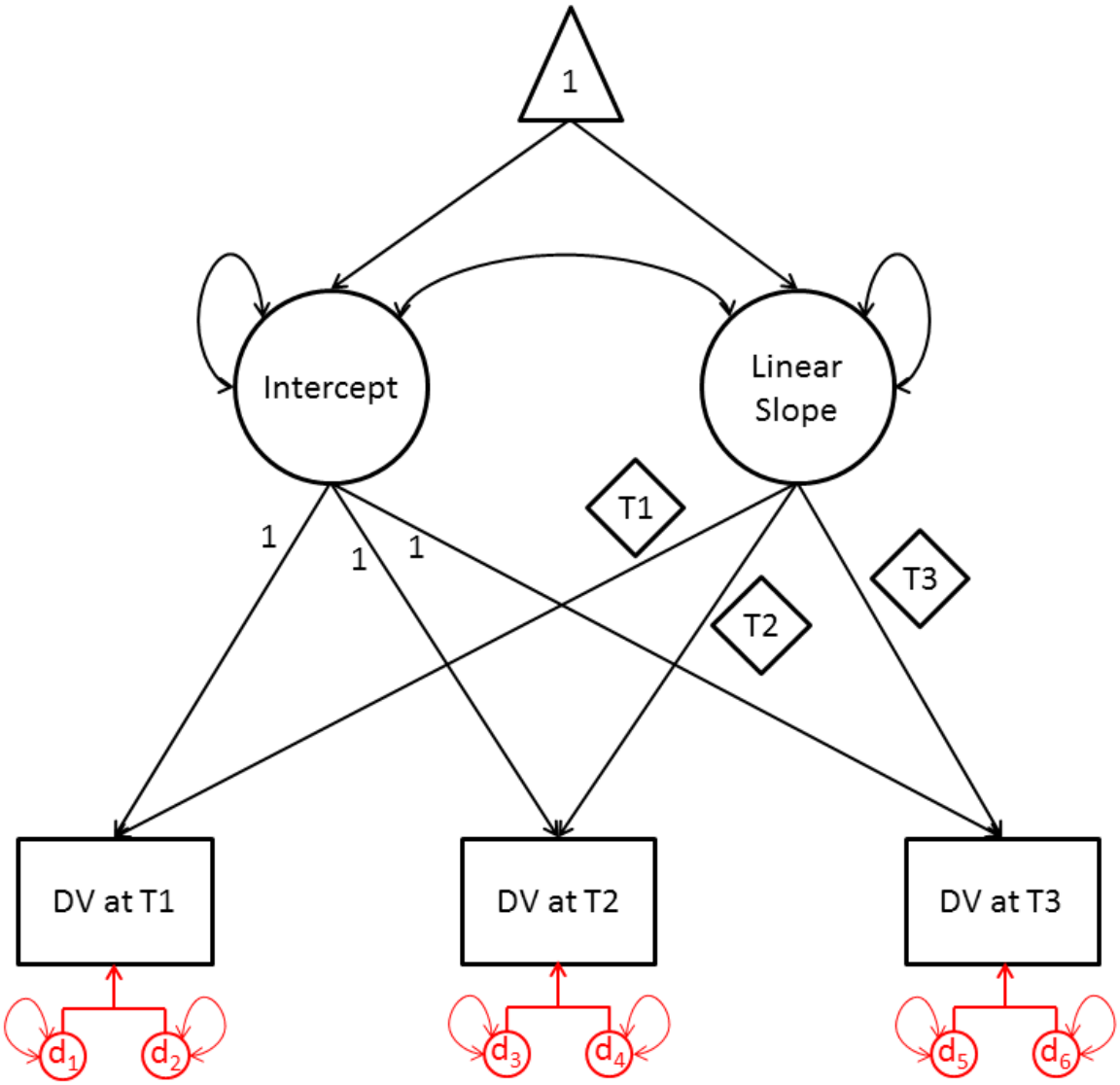

Figure 12. Possible notation for the partially time-unstructured heterogeneous diagonal matrix shown in Figure 1b, with 2 residual variances per measurement occasion (in red).

Limitations and Future Directions

The current research has some limitations, and could be extended in future investigations of growth curve modeling with time-unstructured data. Of note, time was not truly continuous in this study. Instead, time was treated as a discrete, interval-scale

variable that could take on a limited number of integer values. In some applied studies, data could be truly time-unstructured, whereby every single participant is assessed at different times. In such a situation, the consequences of ignoring variation in times of assessment when modeling the residual (co)variances could be even worse than the deleterious effects found in this study. In addition, when the data are truly time-unstructured, the fully time-unstructured matrices presented here are underidentified and therefore cannot be used to conduct the analysis. As a result, this study might have underestimated the usefulness of the new partially time-unstructured analysis presented here for applied researchers. Future studies could focus on longitudinal studies where the data are truly time-unstructured, perhaps while evaluating one or both of the two non-partial methods described above under "Partially Time-Unstructured Analysis".

In this simulation, manipulated factors included extent of misspecification of the level-1 matrix (type of analysis), population and modeled level-1 matrix, sample size, and number of assessments per individual. Several other factors could have been manipulated as well. For example, the distribution of the time values at each measurement occasion was uniform (for instance, participants were equally likely to be assessed at times 1, 2, 3, or 4 at their first assessment). Previous studies have often found larger differences between time-structured and time-unstructured analyses when time values follow a skewed distribution at each measurement occasion (Aydin et al., 2014; Coulombe et al., 2016). I also included no predictor other than time. However, previous research has found that the effects of misspecifying the level-1 matrix in growth curve models can differ for level-1 and level-2 predictors (Gomez et al., 2005; Vallejo et al., 2008), and those effects can also depend on whether the time variable interacts with another predictor (Jacqmin-

Gadda et al., 2007). Future research could quantify the effects of ignoring variation in times of assessment in level-1 matrices when level-1 and level-2 predictors other than time are included in the model, for instance in terms of parameter bias or power to detect nonzero change over time.

This study was the first study to investigate the consequences of incorrectly ignoring variation in times of assessment when modeling residual (co)variances. As a starting point, I focused on a simple linear growth curve model in this study. This should be extended to growth curve models where change is not linear. Nonlinear change over time has been investigated in previous simulation studies pertaining both to the analysis of time-unstructured data (Liu et al., 2015) and to the misspecification of residual (co)variances (Enders & Tofighi, 2008; Ferron et al., 2002; Kwok et al., 2007). Similarly, I have restricted my attention to growth curve models where both the intercept and linear slope are random (i.e., participants are allowed to vary in their intercept and slope). However, some evidence suggests that misspecifying the level-1 matrix is associated with worse bias in the fixed effects when slopes are fixed rather than random (Jacqmin-Gadda et al., 2007). This evidence points to the possibility that the general absence of bias in the fixed effects in this study might not generalize to models where the slopes have been fixed. Importantly, researchers can often resort to fixing the slopes when they encounter convergence difficulties (Singer & Willett, 2003); this is particularly important in the context of the modeling of residual (co)variances with time-unstructured data, because the current study shows that such analyses can often run into convergence problems, perhaps especially when the residual variance changes over time.

This study has focused on the comparison between time-structured and partially and fully time-unstructured analyses in the absence of any other model misspecification. It would be interesting to determine whether transitioning from a time-structured level-1 matrix to a partially or fully time-unstructured level-1 matrix can help detecting other model misspecifications, or perhaps even *reducing* the negative effects of other model misspecifications. Wu and West (2000) have found that maximizing the number of parameters estimated in the level-1 matrix (i.e., saturating the level-1 matrix) improves the ability of traditional SEM fit indexes to detect misspecification in the fixed effects (for example, when only a linear slope is included in the model but change over time follows a quadratic trajectory). Based on this finding, they recommend saturating the level-1 matrix in order to detect other model misspecifications. Unfortunately, this strategy currently cannot be employed with time-unstructured data: As I mentioned above, indexes of absolute fit have not yet been extended to time-unstructured data analyses; moreover, increasing the number of parameters in the level-1 matrix to accommodate time-unstructured data was associated with convergence problems in this study, and if time is truly continuous, such matrices are simply not identified. At the very least, indexes of absolute fit for time-unstructured analyses will need to be developed before future research can determine whether using a partially (or fully) time-unstructured level-1 matrix can aid in the detection of other model misspecifications.

One limitation concerns specifically the partially (and fully) time-unstructured matrices introduced in this study. While there is a direct equivalence between the MLM and SEM approaches to growth curve modeling under certain conditions (e.g., Bauer, 2003; Coulombe et al., 2016; Curran, 2003; Mehta & West, 2000), MLM and SEM

software packages, on the other hand, are not necessarily equivalent (Preacher et al., 2008). Currently, the partially time-unstructured analysis detailed here is straightforward to implement in many MLM software packages (and even in general-purpose packages like SPSS), but difficult to implement in SEM software packages (like Mplus). This means that the possibility of implementing the time-unstructured analyses presented in this paper can be limited by the availability of software packages to applied researchers. Until packages dedicated to the analysis of time-unstructured data are developed, researchers interested in estimating partially or fully time-unstructured residual variance-covariance matrices will likely have to do so using MLM software.

Chapter 5

Conclusion

In conclusion, the current study is the first study to examine the consequences of ignoring variability in times of assessment when modeling residual variances and covariances in growth curve modeling. This study also illustrates how residual variance-covariance matrices can be adjusted to accommodate time-unstructured data fully or partially. All in all, this study suggests that accommodating time-unstructured data when modeling residual variances and covariances can be important, perhaps especially when residuals are autocorrelated. Further, until non-partial methods are devised to fully accommodate time-unstructured data, the partially time-unstructured matrices described in this study can provide an improvement over the standard time-structured matrices under several conditions.

APPENDIX

Code Used to Generate and Analyze the Data

```
#########################################################
################### DATA GENERATION ###################


#### initialize session ####
rm(list=ls()) #remove all existing object


#### set working directory ####
setwd("c:/.../dissertation/simul")


#### set simulation factors ####
nrep <- 1000
simul <- list()
simul$t <- c(3,6,9)
simul$n <- c(50, 200, 500)
simul$matrix <- c("het", "ar1")
simul$mis <- c("none", "int", "max")


#minimum conv rate for condition to be used in results
simul$conv.min <- .010 #1%, i.e. 10 iterations



#### set population values ####
pop <- list()
pop$alpha <- c(50, 2) #mean intercept & slope
pop$psi <- matrix(c(100, 12, 12, 16), 2, 2)
pop$cor <- pop$psi[1,2]/(pop$psi[1,1]*pop$psi[2,2])^0.5
pop$eps <- 100
pop$rho <- +0.5 #rho in AR(1) matrix
pop$var.increase <- 3 #increase in residual variance from one time value to the next



#### functions to generate heterogeneous and AR(1) matrices of residuals for the full
sample ####
#each column = a person, each row = a wave number

#heterogeneous diagonal
generateHetMatrix <- function(timepts, t, n)
{
  .var <- 100+3*((1:(t*4)-1))

  .matrix <- matrix(nrow=t, ncol=n, byrow=F)
  for (time in as.numeric(names(table(timepts))))
  {
    .indexes <- timepts[,] == time
    .matrix[ .indexes ] <- rnorm(sum(.indexes), 0, sqrt(.var[time]))
  }

  return(.matrix)
}

#AR(1)
library(CVTuningCov)
generateAR1Matrix <- function(timepts, t, n)
{
  matrix(rnorm(t*n,0,sqrt(pop$eps)), nrow=t, ncol=n, byrow=F)
  mvrnorm(n, mu=c(0,0), Sigma=pop$psi)

  #create AR(1) matrix (using CVTuningCov::AR1())
```

```
  .Sigma <- AR1(p=36, rho=pop$rho)*pop$eps #AR1() produces a correlation matrix; multiply
by variance to get vcov matrix

  #generate residuals
  .matrix.full <- t(mvrnorm(n, mu=rep(0, 36), Sigma=.Sigma)) #transposed via t() to get
persons in columns, waves in rows

  #need to select residuals for the timepts that were observed for each person
(currently, each person has 36 residuals instead of t)
  #might not be most efficient, but will do one person at a time
  .matrix <- matrix(nrow=t, ncol=n, byrow=F)
  for (col in 1:ncol(.matrix.full))
  {
    .matrix[,col] <- .matrix.full[timepts[,col], col] #keep only row indexes from timepts
for that person (column)
  }

  return(.matrix)
}


#### generate datasets ####
library(MASS)
d <- list() #to store datasets
cond <- data.frame() #to keep track of the conditions for each dataset
vector() -> cond$i -> cond$t -> cond$n -> cond$mat

for (t in simul$t)
{
  .colnames <- c(paste("y", 1:t, sep=""), paste("t", 1:t, sep=""))

  #factor loadings (lambda)
  lambda<-matrix(NA, t, 2)
  lambda[,1] <- 1

  for (n in simul$n)
  {
    for (mat in simul$matrix)
    {
      for (r in 1:nrep)
      {
        #factor loadings slope (timepoints)
        timepts <- matrix(nrow=t, ncol=n, byrow=F)
        for(wave in 1:t)
        {
          timepts[wave,] <- sample(x=(1:4)+4*(wave-1), size=n, replace=TRUE)
        }

        .d<-matrix(NA,0,t*2) #(empty) vector of scores for one person


        #level-2 residuals for full sample: zeta (sample from bivariate normal)
        zeta <- mvrnorm(n, mu=c(0,0), Sigma=pop$psi) #equation 7 in master's thesis

        #level-1 residuals vector (epsilon) for full sample; each column = a person, each
row = a wave number
        epsilon <- switch(mat,
                          homog=matrix(rnorm(t*n,0,sqrt(pop$eps)), nrow=t, ncol=n,
byrow=F), #equation 7
                          het=generateHetMatrix(timepts=timepts, t=t, n=n),
                          ar1=generateAR1Matrix(timepts=timepts, t=t, n=n)
                            )


        #each person in sample for repetition r
        for (i in 1:n)
        {
          y <- vector()
          #factor loadings
          lambda[,2] <- timepts[,i]
```

```
          #eta
          eta <- c(pop$alpha[1], pop$alpha[2]) + zeta[i,] #equation 6

          #compute vector of scores for observation
          y <- lambda%*%eta + epsilon[,i] #equation 5
          .d <- rbind(.d, c(as.vector(y), lambda[,2]) )
        }

        .d<-as.data.frame(.d) #n rows, Ts columns
        colnames(.d) <- .colnames
        d[[length(d)+1]] <- .d #stores rth dataset in list of datasets
        cond[nrow(cond)+1, ] <- c(nrow(cond)+1, t, n, mat)
      }
    }
  }
}


#### reshape datasets for multilevel format (from wide to long) ####
d.wide <- d
d <- lapply(
  X = d,
  FUN = function(x) {
    t <- ncol(x)/2
    .d <- reshape(x, varying=list(colnames(x)[(t+1):ncol(x)], colnames(x)[1:t]),
v.names=c("time", "dv"), timevar="wave", times=1:t, direction="long")
    .d <- .d[order(.d$id, .d$time), ]
    return(.d)
  }
)



#### create time.int variable for intermediate misspecification ####
createTimeInt <- function(d)
{
  d[d$time %in% 1:2, "time.int"] <- 1
  d[d$time %in% 3:4, "time.int"] <- 2
  d[d$time %in% 5:6, "time.int"] <- 3
  d[d$time %in% 7:8, "time.int"] <- 4
  d[d$time %in% 9:10, "time.int"] <- 5
  d[d$time %in% 11:12, "time.int"] <- 6
  d[d$time %in% 13:14, "time.int"] <- 7
  d[d$time %in% 15:16, "time.int"] <- 8
  d[d$time %in% 17:18, "time.int"] <- 9
  d[d$time %in% 19:20, "time.int"] <- 10
  d[d$time %in% 21:22, "time.int"] <- 11
  d[d$time %in% 23:24, "time.int"] <- 12
  d[d$time %in% 25:26, "time.int"] <- 13
  d[d$time %in% 27:28, "time.int"] <- 14
  d[d$time %in% 29:30, "time.int"] <- 15
  d[d$time %in% 31:32, "time.int"] <- 16
  d[d$time %in% 33:34, "time.int"] <- 17
  d[d$time %in% 35:36, "time.int"] <- 18

  return(d)
}
d <- lapply(d, createTimeInt)


##########################################################
```

```
#########################################################
################ ANALYZE DATASETS 3 WAYS ################


#### factory() function (http://stackoverflow.com/questions/4948361/how-do-i-save-
warnings-and-errors-as-output-from-a-function) ####
#### to be able to save warnings & errors from multilevel analyses
factory <- function(fun)
  function(...) {
    warn <- err <- NULL
    res <- withCallingHandlers(
      tryCatch(fun(...), error=function(e) {
        err <<- conditionMessage(e)
        NULL
      }), warning=function(w) {
        warn <<- append(warn, conditionMessage(w))
        invokeRestart("muffleWarning")
      })
    list(res, warn=warn, err=err)
  }



#### with lapply() and factor(), but only by batches ####
#18,000 datasets (18 "conditions") to analyze in 3 ways
setwd("C:/.../Dissertation")
library(nlme)



##### NONE (fully time-unstructured) #####
#none.het.t3.n50
.cond <- which(cond$mat == "het" & cond$t == 3 & cond$n == 50)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|time), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.none.het.t3.n50.rds")
rm(ana); gc()


#none.het.t3.n200
.cond <- which(cond$mat == "het" & cond$t == 3 & cond$n == 200)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|time), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.none.het.t3.n200.rds")
rm(ana); gc()


#none.het.t3.n500
.cond <- which(cond$mat == "het" & cond$t == 3 & cond$n == 500)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|time), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.none.het.t3.n500.rds")
rm(ana); gc()


#none.het.t6.n50
.cond <- which(cond$mat == "het" & cond$t == 6 & cond$n == 50)
ana <- lapply(d[.cond],
```

```
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|time), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.none.het.t6.n50.rds")
rm(ana); gc()


#none.het.t6.n200
.cond <- which(cond$mat == "het" & cond$t == 6 & cond$n == 200)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|time), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.none.het.t6.n200.rds")
rm(ana); gc()


#none.het.t6.n500
.cond <- which(cond$mat == "het" & cond$t == 6 & cond$n == 500)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|time), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.none.het.t6.n500.rds")
rm(ana); gc()


#none.het.t9.n50
.cond <- which(cond$mat == "het" & cond$t == 9 & cond$n == 50)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|time), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.none.het.t9.n50.rds")
rm(ana); gc()


#none.het.t9.n200
.cond <- which(cond$mat == "het" & cond$t == 9 & cond$n == 200)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|time), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.none.het.t9.n200.rds")
rm(ana); gc()


#none.het.t9.n500
.cond <- which(cond$mat == "het" & cond$t == 9 & cond$n == 500)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|time), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.none.het.t9.n500.rds")
rm(ana); gc()



#ar1
```

```
#none.ar1.t3.n50
.cond <- which(cond$mat == "ar1" & cond$t == 3 & cond$n == 50)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~time), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.none.ar1.t3.n50.rds")
rm(ana); gc()


#none.ar1.t3.n200
.cond <- which(cond$mat == "ar1" & cond$t == 3 & cond$n == 200)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~time), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.none.ar1.t3.n200.rds")
rm(ana); gc()


#none.ar1.t3.n500
.cond <- which(cond$mat == "ar1" & cond$t == 3 & cond$n == 500)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~time), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.none.ar1.t3.n500.rds")
rm(ana); gc()


#none.ar1.t6.n50
.cond <- which(cond$mat == "ar1" & cond$t == 6 & cond$n == 50)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~time), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.none.ar1.t6.n50.rds")
rm(ana); gc()


#none.ar1.t6.n200
.cond <- which(cond$mat == "ar1" & cond$t == 6 & cond$n == 200)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~time), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.none.ar1.t6.n200.rds")
rm(ana); gc()


#none.ar1.t6.n500
.cond <- which(cond$mat == "ar1" & cond$t == 6 & cond$n == 500)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~time), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.none.ar1.t6.n500.rds")
rm(ana); gc()
```

```
#none.ar1.t9.n50
.cond <- which(cond$mat == "ar1" & cond$t == 9 & cond$n == 50)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~time), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.none.ar1.t9.n50.rds")
rm(ana); gc()


#none.ar1.t9.n200
.cond <- which(cond$mat == "ar1" & cond$t == 9 & cond$n == 200)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~time), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.none.ar1.t9.n200.rds")
rm(ana); gc()


#none.ar1.t9.n500
.cond <- which(cond$mat == "ar1" & cond$t == 9 & cond$n == 500)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~time), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.none.ar1.t9.n500.rds")
rm(ana); gc()




##### INTERMEDIATE (partially time-unstructured) #####

#int.het.t3.n50
.cond <- which(cond$mat == "het" & cond$t == 3 & cond$n == 50)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|time.int), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.int.het.t3.n50.rds")
rm(ana); gc()


#int.het.t3.n200
.cond <- which(cond$mat == "het" & cond$t == 3 & cond$n == 200)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|time.int), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.int.het.t3.n200.rds")
rm(ana); gc()


#int.het.t3.n500
.cond <- which(cond$mat == "het" & cond$t == 3 & cond$n == 500)
ana <- lapply(d[.cond],
              factory(
```

```
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|time.int), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.int.het.t3.n500.rds")
rm(ana); gc()


#int.het.t6.n50
.cond <- which(cond$mat == "het" & cond$t == 6 & cond$n == 50)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|time.int), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.int.het.t6.n50.rds")
rm(ana); gc()


#int.het.t6.n200
.cond <- which(cond$mat == "het" & cond$t == 6 & cond$n == 200)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|time.int), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.int.het.t6.n200.rds")
rm(ana); gc()


#int.het.t6.n500
.cond <- which(cond$mat == "het" & cond$t == 6 & cond$n == 500)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|time.int), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.int.het.t6.n500.rds")
rm(ana); gc()


#int.het.t9.n50
.cond <- which(cond$mat == "het" & cond$t == 9 & cond$n == 50)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|time.int), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.int.het.t9.n50.rds")
rm(ana); gc()


#int.het.t9.n200
.cond <- which(cond$mat == "het" & cond$t == 9 & cond$n == 200)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|time.int), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.int.het.t9.n200.rds")
rm(ana); gc()


#int.het.t9.n500
.cond <- which(cond$mat == "het" & cond$t == 9 & cond$n == 500)
ana <- lapply(d[.cond],
```

```
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|time.int), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.int.het.t9.n500.rds")
rm(ana); gc()



#ar1
#int.ar1.t3.n50
.cond <- which(cond$mat == "ar1" & cond$t == 3 & cond$n == 50)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~time.int), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.int.ar1.t3.n50.rds")
rm(ana); gc()


#int.ar1.t3.n200
.cond <- which(cond$mat == "ar1" & cond$t == 3 & cond$n == 200)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~time.int), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.int.ar1.t3.n200.rds")
rm(ana); gc()


#int.ar1.t3.n500
.cond <- which(cond$mat == "ar1" & cond$t == 3 & cond$n == 500)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~time.int), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.int.ar1.t3.n500.rds")
rm(ana); gc()


#int.ar1.t6.n50
.cond <- which(cond$mat == "ar1" & cond$t == 6 & cond$n == 50)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~time.int), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.int.ar1.t6.n50.rds")
rm(ana); gc()


#int.ar1.t6.n200
.cond <- which(cond$mat == "ar1" & cond$t == 6 & cond$n == 200)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~time.int), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.int.ar1.t6.n200.rds")
rm(ana); gc()
```

```
#int.ar1.t6.n500
.cond <- which(cond$mat == "ar1" & cond$t == 6 & cond$n == 500)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~time.int), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.int.ar1.t6.n500.rds")
rm(ana); gc()


#int.ar1.t9.n50
.cond <- which(cond$mat == "ar1" & cond$t == 9 & cond$n == 50)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~time.int), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.int.ar1.t9.n50.rds")
rm(ana); gc()


#int.ar1.t9.n200
.cond <- which(cond$mat == "ar1" & cond$t == 9 & cond$n == 200)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~time.int), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.int.ar1.t9.n200.rds")
rm(ana); gc()


#int.ar1.t9.n500
.cond <- which(cond$mat == "ar1" & cond$t == 9 & cond$n == 500)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~time.int), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.int.ar1.t9.n500.rds")
rm(ana); gc()




##### MAXIMAL (time-structured) #####
#max.het.t3.n50
.cond <- which(cond$mat == "het" & cond$t == 3 & cond$n == 50)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|wave), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.max.het.t3.n50.rds")
rm(ana); gc()


#max.het.t3.n200
.cond <- which(cond$mat == "het" & cond$t == 3 & cond$n == 200)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|wave), data=x, control=lmeControl(returnObject = TRUE)) }
```

```
              )
)
saveRDS(ana, file="ana.max.het.t3.n200.rds")
rm(ana); gc()


#max.het.t3.n500
.cond <- which(cond$mat == "het" & cond$t == 3 & cond$n == 500)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|wave), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.max.het.t3.n500.rds")
rm(ana); gc()


#max.het.t6.n50
.cond <- which(cond$mat == "het" & cond$t == 6 & cond$n == 50)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|wave), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.max.het.t6.n50.rds")
rm(ana); gc()


#max.het.t6.n200
.cond <- which(cond$mat == "het" & cond$t == 6 & cond$n == 200)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|wave), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.max.het.t6.n200.rds")
rm(ana); gc()


#max.het.t6.n500
.cond <- which(cond$mat == "het" & cond$t == 6 & cond$n == 500)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|wave), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.max.het.t6.n500.rds")
rm(ana); gc()


#max.het.t9.n50
.cond <- which(cond$mat == "het" & cond$t == 9 & cond$n == 50)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|wave), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.max.het.t9.n50.rds")
rm(ana); gc()


#max.het.t9.n200
.cond <- which(cond$mat == "het" & cond$t == 9 & cond$n == 200)
ana <- lapply(d[.cond],
              factory(
```

```
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|wave), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.max.het.t9.n200.rds")
rm(ana); gc()


#max.het.t9.n500
.cond <- which(cond$mat == "het" & cond$t == 9 & cond$n == 500)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
weights=varIdent(form=~1|wave), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.max.het.t9.n500.rds")
rm(ana); gc()



#ar1
#max.ar1.t3.n50
.cond <- which(cond$mat == "ar1" & cond$t == 3 & cond$n == 50)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~wave), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.max.ar1.t3.n50.rds")
rm(ana); gc()


#max.ar1.t3.n200
.cond <- which(cond$mat == "ar1" & cond$t == 3 & cond$n == 200)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~wave), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.max.ar1.t3.n200.rds")
rm(ana); gc()


#max.ar1.t3.n500
.cond <- which(cond$mat == "ar1" & cond$t == 3 & cond$n == 500)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~wave), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.max.ar1.t3.n500.rds")
rm(ana); gc()


#max.ar1.t6.n50
.cond <- which(cond$mat == "ar1" & cond$t == 6 & cond$n == 50)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~wave), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.max.ar1.t6.n50.rds")
rm(ana); gc()


#max.ar1.t6.n200
```

```
.cond <- which(cond$mat == "ar1" & cond$t == 6 & cond$n == 200)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~wave), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.max.ar1.t6.n200.rds")
rm(ana); gc()


#max.ar1.t6.n500
.cond <- which(cond$mat == "ar1" & cond$t == 6 & cond$n == 500)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~wave), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.max.ar1.t6.n500.rds")
rm(ana); gc()


#max.ar1.t9.n50
.cond <- which(cond$mat == "ar1" & cond$t == 9 & cond$n == 50)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~wave), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.max.ar1.t9.n50.rds")
rm(ana); gc()


#max.ar1.t9.n200
.cond <- which(cond$mat == "ar1" & cond$t == 9 & cond$n == 200)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~wave), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.max.ar1.t9.n200.rds")
rm(ana); gc()


#max.ar1.t9.n500
.cond <- which(cond$mat == "ar1" & cond$t == 9 & cond$n == 500)
ana <- lapply(d[.cond],
              factory(
                function(x) { lme(dv ~ time, random=~time|id,
correlation=corAR1(form=~wave), data=x, control=lmeControl(returnObject = TRUE)) }
              )
)
saveRDS(ana, file="ana.max.ar1.t9.n500.rds")
rm(ana); gc()


##########################################################
```

```
#####################################################################
############# CREATE TABLE WITH ONE LINE PER OUTPUT    ##############
############# WITH PARAM. ESTIMATES AND OTHER OUTCOMES ##############


#### RETRIEVE RESULTS FOR EACH ITERATION
#need:
#param estimates (fixed effects: intercept, slope; variances: intercept, slope,
correlation; residual (co)variances)
#hyp test slope (for power)
#fit indexes
#converged to admissible solution (i.e., warning/error/apVar)

#need table with iteration #; mis; mat; t; n; conv; logLik; AIC; BIC; icept; slope;
slope.se; slope.p; var.icept; var.slope; var.resid; all var.weight.[t]; phi;
createOutputsTable <- function()
{
  .v <- vector()
  outputs <- data.frame(i=.v, mis=.v, mat=.v, t=.v, n=.v, conv=.v, LL=.v, AIC=.v, BIC=.v,
icept=.v, slope=.v, slope.se=.v, slope.p=.v, var.icept=.v, var.slope=.v, var.cor=.v,
var.resid=.v,
                        var.weight.1=.v,
                        var.weight.2=.v,
                        var.weight.3=.v,
                        var.weight.4=.v,
                        var.weight.5=.v,
                        var.weight.6=.v,
                        var.weight.7=.v,
                        var.weight.8=.v,
                        var.weight.9=.v,
                        var.weight.10=.v,
                        var.weight.11=.v,
                        var.weight.12=.v,
                        var.weight.13=.v,
                        var.weight.14=.v,
                        var.weight.15=.v,
                        var.weight.16=.v,
                        var.weight.17=.v,
                        var.weight.18=.v,
                        var.weight.19=.v,
                        var.weight.20=.v,
                        var.weight.21=.v,
                        var.weight.22=.v,
                        var.weight.23=.v,
                        var.weight.24=.v,
                        var.weight.25=.v,
                        var.weight.26=.v,
                        var.weight.27=.v,
                        var.weight.28=.v,
                        var.weight.29=.v,
                        var.weight.30=.v,
                        var.weight.31=.v,
                        var.weight.32=.v,
                        var.weight.33=.v,
                        var.weight.34=.v,
                        var.weight.35=.v,
                        var.weight.36=.v,
                        phi=.v
  )
  return(outputs)
}

library(nlme)

for (mis in c("none", "int", "max"))
{
  for (mat in c("het", "ar1"))
  {
    for (t in c(3, 6, 9))
    {
      for (n in c(50, 200, 500))
```

```
      {
        outputs <- createOutputsTable()
        ana <- readRDS(paste0("ana.", mis, ".", mat, ".t", t, ".n", n, ".rds"))

        for (i in 1:length(ana))
        {
          #fill in descriptives for that condition
          outputs[i, c("i", "mis", "mat", "t", "n")] <- c(i, mis, mat, t, n)

          #load ith analysis
          .ana <- ana[[i]][[1]]
          #if (!is.null(.ana)) {
          .out <- summary(.ana)

          ###determine convergence
          outputs[i, "conv"] <- is.null(ana[[i]]$warn) & is.null(ana[[i]]$err) &
length(.ana$apVar) != 1

          ###fit indexes
          outputs[i, c("LL", "AIC", "BIC")] <- c(as.numeric(.out$logLik), .out$AIC,
.out$BIC)

          ###param estimates
          outputs[i, c("icept", "slope", "slope.se", "slope.p", "var.icept", "var.slope",
"var.cor", "var.resid")] <-
            c(.out$tTable["(Intercept)", "Value"], .out$tTable["time", "Value"],
.out$tTable["time", "Std.Error"], .out$tTable["time", "p-value"],
VarCorr(.ana)["(Intercept)", "Variance"], VarCorr(.ana)["time", "Variance"],
VarCorr(.ana)["time", "Corr"], .out$sigma)

          ###var.weights if mat==het
          if (mat == "het")
          {
            .w <- attributes(.ana$modelStruct$varStruct)$weights #weights
            .w <- data.frame(t=as.numeric(names(.w)), weight=.w)
            .w <- subset(.w, !duplicated(.w$t)) #keep one var per t
            .w <- .w[order(.w$t), ] #reorder
            .w$var <- (1/.w$weight)^2*.ana$sigma^2
            #.w$sd <- 1/.w$weight

            row.names(.w) <- paste0("var.weight.", .w$t)
            .w <- t(.w)
            .w <- .w[row.names(.w) == "var", ]

            #merge
            outputs[i, which(names(outputs) %in% names(.w))] <- .w
          }


          ###phi if mat==ar1
          else if (mat == "ar1")
          {
            outputs[i, "phi"] <- coef(.ana$modelStruct$corStruct, unconstrained=F)
          }

        }

        #save outputs
        saveRDS(outputs, paste0("outputs.", mis, ".", mat, ".t", t, ".n", n, ".rds"))
        rm(outputs, ana); gc()
      }
    }
  }
}



#### combine all outputs ####
o <- createOutputsTable()
for (mis in c("none", "int", "max"))
{
```

```
  for (mat in c("het", "ar1"))
  {
    for (t in c(3, 6, 9))
    {
      for (n in c(50, 200, 500))
      {
          .o <- readRDS(paste0("outputs.", mis, ".", mat, ".t", t, ".n", n, ".rds"))
          o <- rbind(o, .o)
      }
    }
  }
}
#utils::View(o)

#save
saveRDS(o, "c:/users/patri_000/google drive/dissertation/simulated datasets/o.rds")



######################################################################
######################################################################
############## CREATE TABLE WITH ONE LINE PER CONDITION ##############
##############  WITH PARAMETER BIAS AND OTHER OUTCOMES  ##############


#### RESULTS BY CONDITIONS ####
outputs <- readRDS("o.rds")

#fix class of variables in o
for (col in c(4:5, 10:17)) class(outputs[,col]) <- "numeric"

#fix var.resid--currently shown as SD, need to square
outputs$var.resid <- outputs$var.resid^2

#change LL into deviance (-2LL), keep variable name as "LL"
outputs$LL <- -2*outputs$LL

#convergence **using outputs, not o [see next]**
o.all <- aggregate(conv ~ mis + mat + t + n, data=outputs, FUN=function(x)
sum(x)/length(x))

###exclude or include inadmissible solutions
#if keep only converged solutions, need to eliminate conditions with < 10 converged
iterations (i.e., conv < .010)
.conds.notEnoughConv <- subset(o.all, conv < simul$conv.min & conv > 0)
.conds.notEnoughConv$cond <- with(.conds.notEnoughConv, paste0(mis, mat, t, n))
.conds <- with(outputs, paste0(mis, mat, t, n)); .conv <- outputs[, "conv"];
.conv[which(.conds %in% .conds.notEnoughConv$cond)] <- FALSE

sub <- list(excl.inad = .conv == TRUE, incl.inad = rep(TRUE, nrow(outputs)))$excl.inad
#toggle
o <- subset(outputs, subset=sub)



#bias
o.bias <- aggregate(cbind(icept, slope, var.icept, var.slope, var.cor, var.resid) ~ mis +
mat + t + n, data=o, FUN=mean)
o.bias$icept <- (o.bias$icept - pop$alpha[1])/pop$alpha[1] * 100
o.bias$slope <- (o.bias$slope - pop$alpha[2])/pop$alpha[2] * 100
o.bias$var.icept <- (o.bias$var.icept - pop$psi[1,1])/pop$psi[1,1] * 100
o.bias$var.slope <- (o.bias$var.slope - pop$psi[2,2])/pop$psi[2,2] * 100
o.bias$var.cor <- (o.bias$var.cor - pop$cor)/pop$cor * 100
o.bias$var.resid <- (o.bias$var.resid - pop$eps)/pop$eps * 100

#merge
o.all <- merge(x=o.all, y=o.bias, all.x=TRUE, all.y=TRUE, by=c("mis", "mat", "t", "n"));
rm(o.bias)
```

```
#p-value (i.e., power)
o.power <- aggregate(slope.p ~ mis + mat + t + n, data=o, FUN=function(x)
sum(x<.05)/length(x))
o.all <- merge(x=o.all, y=o.power, all.x=TRUE, all.y=TRUE, by=c("mis", "mat", "t", "n"));
rm(o.power)


#var.weight.t
for (t in 1:36)
{
  .v <- aggregate(formula=as.formula(paste0("var.weight.", t, " ~ mis + mat + t + n")),
data=o, FUN=mean, na.rm=TRUE)
  o.all <- merge(x=o.all, y=.v, all.x=TRUE, all.y=TRUE, by=c("mis", "mat", "t", "n"))
}


#phi
o.phi <- aggregate(phi ~ mis + mat + t + n, data=o, FUN=mean)
o.phi$phi <- (o.phi$phi - pop$rho)/pop$rho * 100
o.all <- merge(x=o.all, y=o.phi, all.x=TRUE, all.y=TRUE, by=c("mis", "mat", "t", "n"));
rm(o.phi)

#eliminate conditions with < 10 iterations (i.e., conv < .010)
#o.all[o.all$conv < .010, 6:ncol(o.all)] <- NA


###fit
o.fit <- aggregate(cbind(LL, AIC, BIC) ~ mis + mat + t + n, data=o, FUN=mean)
o.all <- merge(x=o.all, y=o.fit, all.x=TRUE, all.y=TRUE, by=c("mis", "mat", "t", "n"));
rm(o.fit)

o.fv.fit <- subset(o, select=c("i", "mis", "mat", "t", "n", "LL", "AIC", "BIC"))

#need to match iterations by i*mat*t*n
#reshape from long to wide
o.fit.wide <- reshape(o.fv.fit,
  timevar="mis",
  idvar=c("i", "mat", "t", "n"),
  direction="wide"
) #not all cases are complete bc some iterations converged with one matrix type and
didn't with the other
  #but incomplete cases are useful if they have at least 2 of the three misspecifications
available
rm(o.fv.fit)

#get differences
o.fit.wide$LL.noneMinusInt <- with(o.fit.wide, LL.none - LL.int)
o.fit.wide$LL.noneMinusMax <- with(o.fit.wide, LL.none - LL.max)
o.fit.wide$LL.intMinusMax <- with(o.fit.wide, LL.int - LL.max)

o.fit.wide$AIC.noneMinusInt <- with(o.fit.wide, AIC.none - AIC.int)
o.fit.wide$AIC.noneMinusMax <- with(o.fit.wide, AIC.none - AIC.max)
o.fit.wide$AIC.intMinusMax <- with(o.fit.wide, AIC.int - AIC.max)

o.fit.wide$BIC.noneMinusInt <- with(o.fit.wide, BIC.none - BIC.int)
o.fit.wide$BIC.noneMinusMax <- with(o.fit.wide, BIC.none - BIC.max)
o.fit.wide$BIC.intMinusMax <- with(o.fit.wide, BIC.int - BIC.max)

#get proportions better fit using the sign of the differences
o.fit.wide$LL.noneMinusInt.BetterFit <- o.fit.wide$LL.noneMinusInt < 0
o.fit.wide$LL.noneMinusMax.BetterFit <- o.fit.wide$LL.noneMinusMax < 0
o.fit.wide$LL.intMinusMax.BetterFit <- o.fit.wide$LL.intMinusMax < 0

o.fit.wide$AIC.noneMinusInt.BetterFit <- o.fit.wide$AIC.noneMinusInt < 0
o.fit.wide$AIC.noneMinusMax.BetterFit <- o.fit.wide$AIC.noneMinusMax < 0
o.fit.wide$AIC.intMinusMax.BetterFit <- o.fit.wide$AIC.intMinusMax < 0

o.fit.wide$BIC.noneMinusInt.BetterFit <- o.fit.wide$BIC.noneMinusInt < 0
o.fit.wide$BIC.noneMinusMax.BetterFit <- o.fit.wide$BIC.noneMinusMax < 0
o.fit.wide$BIC.intMinusMax.BetterFit <- o.fit.wide$BIC.intMinusMax < 0
```

```
#group by conditions
o.fit <- aggregate(
  cbind(
    LL.noneMinusInt,
    LL.noneMinusMax,
    LL.intMinusMax,
    AIC.noneMinusInt,
    AIC.noneMinusMax,
    AIC.intMinusMax,
    BIC.noneMinusInt,
    BIC.noneMinusMax,
    BIC.intMinusMax,
    LL.noneMinusInt.BetterFit,
    LL.noneMinusMax.BetterFit,
    LL.intMinusMax.BetterFit,
    AIC.noneMinusInt.BetterFit,
    AIC.noneMinusMax.BetterFit,
    AIC.intMinusMax.BetterFit,
    BIC.noneMinusInt.BetterFit,
    BIC.noneMinusMax.BetterFit,
    BIC.intMinusMax.BetterFit
  )
  ~ mat + t + n, data=o.fit.wide,
  FUN=function(x) if (class(x) == "logical")
return(sum(x[!is.na(x)])/length(x[!is.na(x)])) else return(mean(x, na.rm=TRUE)))
rm(o.fit.wide)

###end fit

#######################################################################




#########################################################
######################### PLOTS #########################


#### graphs/tables/etc. ####

.conv <- c("conv")
.fixef <- c("icept", "slope")
.var <- c("var.icept", "var.slope", "var.cor")
.ar1 <- c("var.resid", "phi")
.listparams <- list("conv", c("icept", "slope"), .var, "slope.p", .ar1)
#.listparams <- list(.ar1)
#.listparams <- list("conv")


#need to add .05 margin to the right of last panel
.margins <- c(1,0,1,0)+0.0

for (.params in .listparams)
{
  windows()
  par(mfrow=c(length(.params), 3), mar=.margins) #2 rows, 3 columns (3 timepts)


  for (.param in .params)
  {
    #set ylim for this param
    .ylim <- c(       floor( min(o.all[, .param], na.rm=TRUE)     ), ceiling(
       max(o.all[, .param], na.rm=TRUE)      )       )
    .ylim <- c(max(abs(.ylim),5)*-1, max(abs(.ylim),5))
    .xaxis <- 0
    if (.param %in% c("conv", "slope.p")) { .ylim <- c(0, 1); .xaxis <- NA }

    #set output data
    .output <- o.all
    if (.param %in% c("var.resid", "phi")) .output <- subset(o.all, mat=="ar1")
```

```
    for (.t in simul$t) #columns
    {
      if (.t == 9) par(mar=.margins+c(0,0,0,0.05)) else par(mar=.margins)

      #windows();
      plot(1, type="n", xlab=NA, ylab=NA, ylim=.ylim, xlim=c(50, 500), xaxt="n")#,
yaxt="n")#labels=F)
      axis(1, at=simul$n, labels=T, tick=T, pos=.xaxis)

      #need one line per mat*mis combination
      #mat type = color; mis extent = color darkness
      for (.mat in c("het", "ar1")) #color
      {
        for (.mis in c("none", "int", "max")) #symbol + dashed
        {
          .d <- subset(.output, mat==.mat & mis==.mis & t==.t)
          print(.mat); print(.mis); print(.d[, c("t", .param)])
          lines(x=.d[, "n"], y=.d[, .param], type="b", lty=switch(.mis, none=1, int=2,
max=3), pch=switch(.mis, none=15, int=17, max=19), col=switch(.mat, het="black",
ar1="blue"), lwd=2, cex=2)
        }
      }

    }
  }
}


#### fit graphs ####
#fit graphs don't have mis; the DVs are comparisons between two mis
#do 3 rows: LL, AIC, BIC
#6 lines per graph: 2 mat * 3 diffs (none-int, none-max, int-max); the diffs are the new
params (columns in o.fit)
#for: row (ie fit index); column (t); line color (mat); line type (which comparison)

windows()
par(mfrow=c(3, 3), mar=.margins) #2 rows, 3 columns (3 timepts)

for (.index in c("LL", "AIC", "BIC"))
{
  .ylim <- c(0, 1)

  for (.t in simul$t) #columns (i.e., individual plots)
  {
    if (.t == 9) par(mar=.margins+c(0,0,0,0.05)) else par(mar=.margins)

    plot(1, type="n", xlab=NA, ylab=NA, ylim=.ylim, xlim=c(50, 500), xaxt="n")#,
yaxt="n")#labels=F)
    axis(1, at=simul$n, labels=T, tick=T, pos=NA)

    #need one line per mat*comparison combination
    #mat type = color; comparison = symbol + dashed
    for (.mat in c("het", "ar1")) #color
    {
      for (.comp in c("noneMinusInt", "noneMinusMax", "intMinusMax")) #symbol + dashed
      {
        .d <- subset(o.fit, mat==.mat & t==.t)
        .dv <- paste(.index, .comp, "BetterFit", sep=".")
        print(.mat); print(.comp); print(.d[, c("t", .dv)])
        lines(x=.d[, "n"], y=.d[, .dv], type="b", lty=switch(.comp, noneMinusMax=1,
noneMinusInt=2, intMinusMax=3), pch=switch(.comp, noneMinusMax=15, noneMinusInt=17,
intMinusMax=19), col=switch(.mat, het="black", ar1="blue"), lwd=2, cex=2)
      }
    }

  }
}
```

```
#### var.weight graph ####
.d <- subset(o.all, subset=conv > simul$conv.min & mat=="het" & mis=="none",
select=c("conv", "mis", "mat", "t", "n", names(o.all)[grep(pattern="weight",
x=names(o.all))]))
.d$n <- as.character(.d$n)
windows()

#set ylim for this param
.ylim <- c(    floor( min(.d[, grep(pattern="var.weight", x=names(.d))], na.rm=TRUE)
       ), ceiling(    max(.d[, grep(pattern="var.weight", x=names(.d))], na.rm=TRUE)
       )      )

#plot window
plot(1, type="n", xlab=NA, ylab=NA, ylim=.ylim, xlim=c(1, 36), xaxt="n")#,
yaxt="n")#labels=F)
axis(1, at=1:36, labels=T, tick=T)#, pos=0)

#add all lines
for (.line in 1:nrow(.d))
{
  lines(x=1:(.d[.line, "t"]*4), y=.d[.line, (which(names(.d) ==
"var.weight.1")):(which(names(.d) == paste0("var.weight.", .d[.line, "t"]*4)))],
        type="l", lty=if (.d[.line, "n"] == 500) 3 else if (.d[.line, "n"] == 200) 2 else
if (.d[.line, "n"] == 50) 1, pch=20,
        col=if (.d[.line, "t"] == 9) "black" else if (.d[.line, "t"] == 6) "blue" else if
(.d[.line, "t"] == 3) "orangered",

        lwd=3, cex=2)
}

#add pop line
lines(x=1:36, y=100+3*(1:36-1), type="l", lty=1, pch=20, col="red2", lwd=6, cex=2)



#do bias instead of means
.d.bias <- .d
for (.t in 1:36)
{
  .col <- paste0("var.weight.", .t)
  .pop.var <- 100+(.t-1)*3
  .d.bias[, .col] <- (.d.bias[, .col] - .pop.var)/.pop.var * 100
}

#set ylim for this param
.ylim <- c(    floor( min(.d.bias[, grep(pattern="var.weight", x=names(.d.bias))],
na.rm=TRUE)    ), ceiling(    max(.d.bias[, grep(pattern="var.weight",
x=names(.d.bias))], na.rm=TRUE)       )       )
.ylim <- c(max(abs(.ylim),5)*-1, max(abs(.ylim),5))

#plot window
windows()
plot(1, type="n", xlab=NA, ylab=NA, ylim=.ylim, xlim=c(1, 36), xaxt="n")
axis(1, at=1:36, labels=T, tick=T, pos=0)

#add all lines
for (.line in 1:nrow(.d.bias))
{
  .bias <- .d.bias[.line, (which(names(.d.bias) == "var.weight.1")):(which(names(.d.bias)
== paste0("var.weight.", .d.bias[.line, "t"]*4)))]
  lines(x=1:(.d.bias[.line, "t"]*4), y=.bias,
        type="l", lty=if (.d.bias[.line, "n"] == 500) 3 else if (.d.bias[.line, "n"] ==
200) 2 else if (.d.bias[.line, "n"] == 50) 1, pch=20,
        col=if (.d.bias[.line, "t"] == 9) "black" else if (.d.bias[.line, "t"] == 6)
"blue" else if (.d.bias[.line, "t"] == 3) "orangered",
        lwd=3, cex=2)
  print(.d.bias[.line, "t"]); print(.d.bias[.line, "n"]); print(.bias)
}


#######################################################
```

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control, 19*, 716-723.

Aydin, B., Leite, W. L., & Algina, J. (2014). The consequences of ignoring variability in measurement occasions within data collection waves in latent growth models. *Multivariate Behavioral Research, 49*, 149-160.

Barr, D. J., Levy, R., Scheepers, C., & Tily, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language, 68*, 255-278.

Bates, D., Maechler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software, 67*, 1-48.

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

Bello, N. M., Steibel, J. P., & Tempelman, R. J. (2010). Hierarchical Bayesian modeling of random and residual variance–covariance matrices in bivariate mixed effects models. *Biometrical Journal, 52*, 297-313.

Boscardin, J. W., & Gelman, A. (1996). Bayesian computation for parametric models of heteroscedasticity in the linear model. In T. B. Fomby & R. C. Hill (Eds.), *Advances in Econometrics* (Vol. 11A). Available from http://www.stat.columbia.edu/~gelman/research/published/aiepub.pdf

Cernicchiaro, N., Renter, D. G., Xiang, S., White, B. J., & Bello, N. M. (2013). Hierarchical Bayesian modeling of heterogeneous variances in average daily

weight gain of commercial feedlot cattle. *Journal of Animal Science*, *91*, 2910-2919.

Chen, F. F. (2007). Sensitivity of goodness of fit indexes to lack of measurement invariance. *Structural Equation Modeling*, *14*, 464-504.

Coulombe, P., Selig, J. P., & Delaney, H. D. (2016). Ignoring individual differences in times of assessment in growth curve modeling. *International Journal of Behavioral Development*, *40*, 76-86.

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

Curran, P. J., Obeidat, K., & Losardo, D. (2010). Twelve frequently asked questions about growth curve modeling. *Journal of Cognition and Development, 11*, 121-136.

Dziak, J. J., Coffman, D. L., Lanza, S. T., & Li, R. (2012). *Sensitivity and specificity of information criteria* (Technical Report #12-119). Retrieved from https://methodology.psu.edu/media/techreports/12-119.pdf

Enders, C. K., & Tofighi, D. (2008). The impact of misspecifying class-specific residual variances in growth mixture models. *Structural Equation Modeling*, *15*, 75-95.

Fan, X., Thompson, B., & Wang, L. (1999). Effects of sample size, estimation methods, and model specification on structural equation modeling fit indexes. *Structural Equation Modeling*, *6*, 56-83.

Feingold, A. (2009). Effect sizes for growth-modeling analysis for controlled clinical trials in the same metric as for classical analysis. *Psychological Methods*, *14*, 43-53.

Ferron, J., Dailey, R., & Yi, Q. (2002). Effects of misspecifying the first-level error structure in two-level models of change. *Multivariate Behavioral Research*, *37*, 379-403.

Foulley, J.-L., Quaas, R. L., & d'Arnoldi, C. T. (1998). A link function approach to heterogeneous variance components. *Genetics Selection Evolution*, *30*, 27-43.

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2004). *Bayesian data analysis* (2nd ed.). Boca Raton, FL: CRC Press.

Gelman, A., Meng, X. L., & Stern, H. (1996). Posterior predictive assessment of model fitness via realized discrepancies. *Statistica Sinica, 6*, 733–807.

Gomez, E. V., Schaajle, B., & Fellingham, G. W. (2005). Performance of the Kenward—Roger method when the covariance structure is selected using AIC and BIC. *Communcations in Statistics: Simulation and Computation*, *34*, 377-392.

Grimm, K. J., & Widaman, K. F. (2010). Residual structures in latent growth curve modeling. *Structural Equation Modeling*, *17*, 424-442.

Harring, J. R., & Blozis, S. A. (2014). Fitting correlated residual error structures in nonlinear mixed-effects models using SAS PROC NLMIXED. *Behavioral Research Methods*, *46*, 372-384.

Heene, M., Hilbert, S., Draxler, C., Ziegler, M., & Bühner, M. (2011). Masking misfit in confirmatory factor analysis by increasing unique variances: A cautionary note on the usefulness of cutoff values of fit indices. *Psychological Methods*, *16*, 319-336.

Hertzog, C., Lindenberger, U., Ghisletta, P., & von Oertzen, T. (2006). On the power of multivariate latent growth curve models to detect correlated change. *Psychological Methods*, *11*, 244-252.

Hertzog, C., von Oertzen, T., Ghisletta, P., & Lindenberger, U. (2008). Evaluating the power of latent growth curve models to detect individual differences in change. *Structural Equation Modeling, 15*, 541-563.

Hox, J. J. (2010). *Multilevel Analysis: Techniques and Applications* (2nd ed.). New York: Routledge.

Hu, L., & Bentler, P. M. (1998). Fit indices in covariance structure modeling: Sensitivity to underparameterized model misspecification. *Psychological Methods, 3,* 424-453.

Jacqmin-Gadda, H., Sibillot, S., Proust, C., Molina, J.-M., & Thiébaut, R. (2007). Robustness of the linear mixed model to misspecified error distribution. *Computational Statistics & Data Analysis, 51,* 5142-5154.

Jamrozik, J., Kistemaker, G. J., Dekkers, J. C. M., & Schaeffer, L. R. (1997). Comparison of possible covariates for use in a random regression model for analyses of test day yields. *Journal of Dairy Science, 80,* 2550-2556.

Jamrozik, J., & Schaeffer, L. R. (1997). Estimates of genetic parameters for a test day model with random regressions for yield traits of first lactation Holsteins. *Journal of Dairy Science, 80,* 762-770.

Keppel, G. & Wickens, T. D. (2004). *Design and analysis: A researcher's handbook* (4th ed.). Upper Saddle River, NJ: Prentice Hall.

Keselman, H. J., Algina, J., Kowalchuk, R. K., & Wolfinger, R. D. (1998). A comparison of two approaches for selecting covariance structures in the analysis of repeated measurements. *Communications in Statistics: Simulation and Computation, 27,* 591-604.

Kline, R. B. (2015). *Principles and practice of structural equation modeling* (4th ed.). New York: Guilford.

Kruschke, J. K. (2014). *Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan* (2nd ed.). Burlington, MA: Elsevier.

Kwok, O. M., West, S. G., & Green, S. B. (2007). The impact of misspecifying the within-subject covariance structure in multiwave longitudinal multilevel models: A Monte Carlo study. *Multivariate Behavioral Research*, *42*, 557-592.

Litière, S., Alonso, A., & Molenberghs, G. (2007). Type I and Type II error under random effects misspecification in generalized linear mixed models. *Biometrics*, *63*, 1038-1044.

Littell, R. C., Pendergast, J., & Natarajan, R. (2000). Modelling covariance structure in the analysis of repeated measures data. *Statistics in Medicine*, *19*, 1793-1819.

Liu, Y., Liu, H., Li, H., & Zhao, Q. (2015). The effects of individually varying times of observations on growth parameter estimations in piecewise growth model. *Journal of Applied Statistics*, *42*, 1843-1860.

Lynch, S. M., & Western, B. (2004). Bayesian posterior predictive checks for complex models. *Sociological Methods and Research*, *32*, 301-335.

Maxwell, S. E., & Delaney, H. D. (2004). *Designing experiments and analyzing data: A model comparison perspective.* New York: Taylor & Francis.

McNeish, D., Stapleton, L. M., & Silverman, R. D. (in press). On the unnecessary ubiquity of hierarchical linear modeling. *Psychological Methods*.

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

Mehta, P. D., & West, S. G. (2000). Putting the individual back into individual growth curves. *Psychological Methods, 5,* 23-43.

Meredith, W., & Tisak, J. (1990). Latent curve analysis. *Psychometrika*, *55*, 107-122.

Muthén, B., & Asparouhov, T. (2012). Bayesian structural equation modeling: A more flexible representation of substantive theory. *Psychological Methods, 17,* 313-335.

Muthén, L. K., & Muthén, B. O. (2012). *Mplus: Statistical analysis with latent variables* (7th ed.). Los Angeles, CA: Muthén & Muthén.

Nezlek, J. B. (2008). An introduction to multilevel modeling for social and personality psychology. *Social and Personality Psychology Compass, 2,* 842-860.

OpenMx Development Team (2016). OpenMx (Version 2.5.2). Retrieved from http://openmx.psyc.virginia.edu/

Pinheiro, J., Bates, D., DebRoy, S., Sarkar, D., & R Core Team. (2015). nlme: Linear and nonlinear mixed effects models (Version 3.1-121). Retrieved from http://CRAN.R-project.org/package=nlme

Preacher, K. J., Wichman, A. L., MacCallum, R. C., & Briggs, N. E. (2008). *Latent growth curve modeling*. Thousand Oaks, CA: SAGE.

R Core Team (2015). R: A language and environment for statistical computing (Version 3.2.2). Retrieved from http://www.r-project.org

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

Reddy, S. K. (1992). Effects of ignoring correlated measurement error in structural equation models. *Educational and Psychological Measurement, 52,* 549-570.

Rekaya, R., Carabano, M. J., & Toro, M. A. (1999). Use of test day yields for the genetic evaluation of production traits in Holstein-Friesian cattle. *Livestock Production Science, 57*, 203-217.

Robert-Granié, C., Heude, B., & Foulley, J. L. (2002). Modelling the growth curve of Maine-Anjou beef cattle using heteroskedastic random coefficients models. *Genetics Selection Evolution, 34*, 423-446.

Schwartz, G. (1978). Estimating the dimension of a model. *Annals of Statistics, 6*, 461-464.

Shi, Y. (2009). *The impact of unmodeled time series processes in within-subject residual structure in conditional latent growth modeling: A Monte Carlo study* (Doctoral dissertation). Retrieved from http://ufdc.ufl.edu/UFE0024992/00001/pdf

Singer, J. D., & Willett, J. B. (2003). *Applied longitudinal data analysis: Modeling change and event occurrence*. New York: Oxford University Press.

Sivo, S., Fan, X., & Witta, L. (2005). The biasing effects of unmodeled ARMA time series processes on latent growth curve model estimates. *Structural Equation Modeling, 12*, 215-231.

Sterba, S. K. (2014). Fitting nonlinear latent growth curve models with individually varying time points. *Structural Equation Modeling, 21*, 630-647.

Vallejo, G., Ato, M., & Valdés, T. (2008). Consequences of misspecifying the error covariance structure in linear mixed models for longitudinal data. *Methodology: European Journal of Research Methods for the Behavioral and Social Sciences, 4*, 10-21.

Venables, W. N., & Ripley, B. D. (2002). *Modern applied statistics with S* (4th ed.). New York: Springer.

Verbyla, A. P. (1993). Modelling variance heterogeneity: Residual maximum likelihood and diagnostics. *Journal of the Royal Statistical Society* (Series B), *55*, 493-508.

von Oertzen, T., Hertzog, C., Lindenberger, U., & Ghisletta, P. (2010). The effect of multiple indicators on the power to detect inter-individual differences in change. *British Journal of Mathematical and Statistical Psychology*, *63*, 627-646.

Vrieze, S. I. (2012). Model selection and psychological theory: A discussion of the differences between the Akaike information criterion (AIC) and the Bayesian information criterion (BIC). *Psychological Methods*, *17*, 228-243.

Wang, B. (2014). CVTuningCov: Regularized estimators of covariance matrices with CV Tuning (Version 1.0). Retrieved from http://CRAN.R-project.org/package=CVTuningCov

Wolfinger, R. (1993). Covariance structure selection in general mixed models. *Communication in Statistics: Simulation and Computation*, *22*, 1079-1106.

Wu, W., & West, S. G. (2010). Sensitivity of fit indices to misspecification in growth curve models. *Multivariate Behavioral Research*, *45*, 420-452.