Continuous Time Dynamic Structural Equation Models

Tihomir Asparouhov and Bengt Muthén

Version 5

October 18, 2024

1 Introduction

In this note we describe the residual dynamic structural equation model (RDSEM) where the times of observations are continuous variables. The RDSEM model described in Asparouhov, Hamaker and Muthén (2018) assumes that variables are observed along a discrete time frame and we aim to generalize that model to continuous times of observations. Structural models in the context of continuous time series models have been discussed in Deboeck and Preacher (2016), Driver et al. (2017), Oravecz et al. (2011), Chow et al. (2007), Boker (2002) among others. We generally follow those prior developments as far as the continuous times of observations impact the modeling.

The RDSEM framework also includes an algorithm for approximating continuous times of observations with a discrete time frame. This algorithm requires a time interval specification and is referred to in the Mplus language as TINTERVAL. The timeline is divided into sequential time periods of the same length. For example, if the time interval is a 2-hour period, the timeline in hours is divided in periods (0,2], (2,4], (4,6], etc. Each observation is then assigned to the time period the observation occurred. Time intervals without any observations are then assigned missing values. The smaller the time interval is the more accurate the approximation is. However, the smaller the time interval is the more missing data is inserted into the original data file,

which has some drawbacks. First the data set balloons as missing data is inserted between the observed values to account for the distance between the observations. Large data sets are much more computationally demanding. All of the missing data is imputed in the Bayesian MCMC estimation. Second, the inserted missing data essentially acts as unknown values/parameters in the estimation. The more such values are present in the MCMC estimation the slower the convergence will be. In many cases, the convergence is so slow that it is impractical to estimate within a finite number of MCMC iterations. It is therefore often the case that a larger time period must be selected to obtain convergence, but the larger periods of course provide a much cruder approximation to the continuous times of observations. Another drawback of the TINTERVAL procedure is the situation where multiple observations are taken within the same time interval. In such cases, observations are moved to empty periods which also reduces the precision of the approximation.

Using an autoregressive model that can accommodate the precise times of observation would therefore have two benefits. First, the analysis data set will not be expanded by inserting missing data. Second, the times of observations would not be approximated. There are some additional benefits. A univariate model AR(1) model for $Y_1, Y_2, Y_3, ...$ with autoregressive coefficient of r generally implies that the model for $Y_2, Y_4, Y_6, ...$ is also an AR(1) with an autoregressive coefficient of r^2. This observation implies that when TINTERVAL is doubled we would expect the autoregressive coefficient to be squared or at least not be statistically significant different from that square. In many data sets with continuous times of observations, however, this does not happen. One explanation for this practical contradiction is that the model is not an AR(1) model (possibly an AR(2) model). Another possible explanation is that the TINTERVAL approximation is too crude. When this happens, we typically conclude that doubling TINTERVAL leads to poor approximation and thus it is preferable to use the model with the smaller TINTERVAL. Generally, we lower TINTERVAL until the contradiction disappears. That however may lead to large amounts of missing data and various computational issues. Models based on continuous times avoid such issues and doubling the distance between observations always implies squaring of the autocorrelation.

Another benefit of using continuous time models is the fact that the model can be used to determine the autocorrelation as a function of the distance between the observations $\Delta t = t_j - t_{j-1}$, where t_j is the time when the $j-$th observation was obtained. In the bivariate or multivariate case

this function can be used to determine when the cross-lag impact is the greatest. Cross-lag effects in bivariate and multivariate models are zero when $\Delta t \approx 0$ because observations that are taken approximately at the same time are perfect predictors of each other and thus there is no room for cross-lag prediction. These are also 0 at $\Delta t \approx \infty$. Thus, the cross-lag effects are maximized (by absolute value) somewhere in the middle. This functional variation of the cross-lag effect in terms of time is referred to as unfolding and dissipation of the effects, Hecht and Zitzmann (2021). Finding the precise Δt where the cross-lag effect is maximized has a substantive importance. For example, it can be used to determine practical questions such as: how long it takes for additional physical activity to ease depression symptoms. One possibility is that statistical significance (although rare) can only be detected at around the time of peaking. Knowledge of the peaking interval can be useful also in designing ecological momentary assessment (EMA) and experience sampling (ESM) studies, see Dormann and Griffin (2015).

It should be noted here that in most situations, TINTERVAL works well when used with sufficiently small intervals. The discretization of time simplifies the model and the estimation, which allows us to easily increase model complexity. The primary concern regarding TINTERVAL is how fine the discretization should be and this is precisely where continuous time modeling can be helpful, i.e., in determining the proper value for the TINTERVAL approximation.

In this paper we describe the model implemented in Mplus 8.11. This model generally follows the development in Driver et al. (2017) but not precisely. One of the main differences is that the model we describe here is CT-RDSEM (continuous time RDSEM) and not CT-DSEM. The difference between such frameworks is that in RDSEM the autoregressive structure is contained entirely in the residuals and all regression effects are limited to contemporaneous effects. The RDSEM and DSEM models, however, are equivalent in some important cases such as the VAR model, and thus certain DSEM models can be estimated within the CT-RDSEM framework as well. The development in this paper is limited to AR(1) continuous models which correspond to first order differential equations. In addition, the current implementation in Mplus is limited to observed variable models and does not include latent variables.

2 Continuous time autoregressive structure

The general CT-RDSEM model will be discussed in the next section. That model follows the framework of the DSEM and RDSEM models, i.e., variables are split as within-between and the within part has a dynamic component. In this section we focus on the technical issues that arise from the continuous times of observations and how to define autoregressive structure based on such continuous times. This is precisely where the difference between the CT-RDSEM model and the RDSEM model is. Because the CT-RDSEM framework places the autoregressive structure of the time series entirely in the residuals, this section presents some key aspects of the residual modeling before turning to the full model in the next section. This involves the drift matrix which is a central concept of continuous time analysis. We will use ε as the name of the variables because that is how the autoregressive structure fits in the general model. However, the models described here can also be understood and used with Y as the variable, i.e., the models we describe here apply to observed variables as well. If a CT-RDSEM model does not include structural relations, then $Y = \varepsilon$ and the model described here applies for the actual observed variables.

We begin by describing a single (one individual) time-series autoregressive univariate model based on continuous times of observations. Later, this model will be generalized to a multivariate model, a two-level model consisting of many time-series autoregressive models for a population of individuals, and then to a structural model which includes a variety of path analysis relations.

Suppose that ε_1, ε_2, ..., ε_T is a time series of zero mean variables observed at times t_1, t_2,...,t_T. The covariance structure of an AR(1) model can be defined as follows

$$Cov(\varepsilon_{j_1}, \varepsilon_{j_2}) = \sigma^2 Exp(\rho|t_{j_1} - t_{j_2}|) \tag{1}$$

This model has two parameters. The first parameter is σ^2, which represents time invariant variance of ε_j. The second parameter ρ is referred to as the drift parameter and is interpreted simply as the log of the correlation between two residuals that are 1 unit apart. This parameter is always negative.

The above model can equivalently be rewritten as a structural model

$$\varepsilon_1 = \xi_1 \sim N(0, \sigma^2) \tag{2}$$

and for $j > 1$

$$\varepsilon_j = r_j \varepsilon_{j-1} + \xi_j \tag{3}$$

$$r_j = Exp(\rho |t_j - t_{j-1}|) \tag{4}$$

$$Var(\xi_j) = (1 - r_j^2)\sigma^2. \tag{5}$$

The key things to understand in this model is that $Var(\xi_j)$ is not a model parameter. The model parameters are again σ^2 and ρ and these have the same interpretation as in the model formulation (1), i.e., the variance of the residual ξ_j is not a model parameter but the total variance of ε_j is. The variance of ε_j is time invariant but the variance of ξ_j is not. $Var(\xi_j)$ depends only on the $\Delta t_j = t_j - t_{j-1}$. The bigger the distance between the observations is the less predictive the previous observation is and therefore the bigger $Var(\xi_j)$ is. The autoregressive coefficient r_j is not time invariant as well but it also depends on the time only via Δt_j. The smaller the distance between the consecutive observations is the higher the correlation between them. The variance $Var(\xi_j)$ converges to 0 when $\Delta t_j = t_j - t_{j-1}$ converges to zero. In that case also r_j converges to 1, and ε_j converges to ε_{j-1} as the two are observed at nearly the same time. When Δt_j converges to infinity, r_j converges to 0 and $Var(\xi_j)$ converges to $Var(\varepsilon_j) = \sigma^2$.

Note that this dependence on time for the autoregressive coefficient and the residual variance is exactly the same for discrete time series. The only reason we are able to not specify a DSEM model without that dependence is because we insert missing values for the discrete time points when there is no observation. If we actually construct the model only with the observed values, then the implied DSEM model will have varying autoregressive parameters (and residual variances) that depend on precisely how many intervals have elapsed between the previous and the current observation.

It is important to note here that the autoregressive time dependence is not a time specific model such as the one used in Mplus with cross-random estimation where the time specific coefficients are time-specific random effects. In the continuous time model we describe, the autoregressive coefficient would be time invariant if the observations are equally distant. The drift parameter is time invariant and the model does not have a time-specific instability. The parameters' dependence on time is only to reflect the unequal spacing of the observations.

Another important feature of the above model is the fact that there is no initial condition. In RDSEM and DSEM frameworks at time point 1,

the model assumes an unobserved predictor coming from period 0, which is treated as a missing predictor and is imputed in the MCMC estimation. In the CT-RDSEM model, this is not necessary, and the model is formulated more elegantly.

The above model easily generalizes also to vector form. Suppose that ε_j is a vector of residuals. The structural model in this case is

$$\varepsilon_1 = \xi_1 \sim N(0, \Sigma) \tag{6}$$

and for $j > 1$

$$\varepsilon_j = R_j \varepsilon_{j-1} + \xi_j \tag{7}$$

$$R_j = Exp(D|t_j - t_{j-1}|) \tag{8}$$

$$Var(\xi_j) = \Sigma - R_j \Sigma R_j^T. \tag{9}$$

In the above formulation the drift parameter ρ is replaced by a square matrix D referred to as the drift matrix. In equation (8), the model relies on exponentiation of a matrix. The formal definition of matrix exponentiation is as follows

$$Exp(A) = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + ... + \frac{A^n}{n!} + ... \tag{10}$$

Here I is the identity matrix. The above infinite time series always converges, just like in the univariate case, but it is typically not suitable to use for computational purposes. A more efficient computational method is given in Sidje (1998).

As in the univariate case, the diagonal entries of D are typically negative numbers. The autoregressive matrix $R(t) = Exp(Dt)$ changes as a function of time but it has the same asymptotic properties as in the univariate case. As t converges to 0, $R(t)$ converges to the identity matrix and as t converges to infinity $R(t)$ converges to the zero matrix. In Section 8 below, we provide more information on the Drift matrix and its origin and interpretation.

3 The CT-RDSEM model

Let Y_{ij} be the vector of observed variables for individual i at time t_{ij}. That is, Y_{ij} is the $j-$th observation obtained for individual i and the time the

observation was obtained is t_{ij}. As in the RDSEM model the variables are decomposed as within and between

$$Y_{ij} = Y_{wij} + Y_{bi}. \tag{11}$$

Here Y_{bi} is a latent variable that represents the time invariant portion of Y_{ij} and can generally be interpreted as the random intercept of Y_{ij}. The variable Y_{wij} is the time specific deviation from Y_{bi}.

Next, separate independent models are given for the within and between components. Suppose that X_{ij} is a vector of within level (time-specific) covariates and X_i is a vector of between level (time-invariant) covariates. The within level model is given by the contemporaneous relationships

$$Y_{wij} = B_w Y_{wij} + \Gamma_w X_{ij} + \varepsilon_{ij}. \tag{12}$$

In this equation B_w is a regression matrix that allows us to include in the model regressions among the endogenous variables and Γ_w are the regression coefficients for exogenous variables. For each i, the model for the residual process ε_{i1}, ε_{i2}, ε_{i3}, ... is given by equations (6-9), i.e., the process is AR(1) time series taking into account the continuous times of observations t_{ij}. In the Mplus framework ε is referred to as $Y\widehat{\ }$.

The within-level parameters are thus B_w, Γ_w, the drift matrix D, and Σ the total variance covariance matrix for ε_{ij}. Each of these parameters can be population specific or can be individually specific. If a parameter in B_w, Γ_w, and D is individually specific, a between level random effect s_i is created which will be a variable modeled on the between level. If a parameter in Σ is individually specific, a transformation is applied before it is modeled with a between level random effect. For diagonal elements σ_{pp} of Σ, i.e., variance parameters, $log(\sigma_{pp})$ is a between level random effect. Off diagonal elements of Σ can be made individually specific by applying the Fisher-z transformation for the corresponding correlation, i.e., $\tanh^{-1}(\sigma_{p_1p_2}/\sqrt{\sigma_{p_1p_1}\sigma_{p_2p_2}})$ is modeled as a between level random effect η_i, see Asparouhov and Muthén (2023). Thus

$$\sigma_{p_1p_2} = \frac{Exp(2\eta_i) - 1}{Exp(2\eta_i) + 1}\sqrt{\sigma_{p_1p_1}\sigma_{p_2p_2}}. \tag{13}$$

We can now describe the between level model. Let V_{bi} be the vector containing Y_{bi} and all individually specific random effects used in the within level model instead of non-random parameters. The between level model is then

$$V_{bi} = \nu_b + B_b V_{bi} + \Gamma_b X_i + \zeta_i \tag{14}$$

$$\zeta_i \sim N(0, \Sigma_B). \tag{15}$$

The parameters in the between model are ν_b, B_b, Γ_b and Σ_B.

The above model is quite flexible in terms of the variety of path analysis models that can be implemented. It includes latent centering in the sense of Asparouhov and Muthén (2019), i.e. the predictors on the within level Y_{wij} are centered $Y_{wij} = Y_{ij} - Y_{bi}$. If there are no within level covariates in the model and B_w is zero, then $Y_{wij} = \varepsilon_{ij}$ and the model becomes equivalent to the vector autoregressive (VAR) model, where all autoregressive and cross-lag parameters are adjusted to reflect the distance between the observations and are modeled via the drift matrix D. The RDSEM model in Mplus has two different contemporaneous regression possibilities on the within level. One is between the full within level components and one is between the residuals. In CT-DSEM only the full components can be used. Contemporaneous relationship between the residuals is not available. Latent factors and categorical dependent variables are currently not implemented in the CT-RDSEM framework.

Missing data can be accommodated in the above framework for the dependent variables. Missing values are imputed using all other observed data and the model. Unlike RDSEM, CT-RDSEM likelihood is not affected by observations where all dependent variables are missing, i.e., if on a particular occasion no data is collected, the observation does not need to be included in the analysis. This observation corresponds to the fact that missing data is inserted in RDSEM for periods without observations, while CT-RDSEM which is based on continuous times does not have periods at all. The CT-RDSEM model and the RDSEM model are technically equivalent for the case where $t_{ij} = j$, i.e., if the observations are indeed obtained at regular/equal intervals the two models should yield equivalent results. This however has two caveats. The two models use different parameterizations. The autoregressive matrix R in RDSEM and the drift matrix D in CT-RDSEM are of course connected via the relationship $R = Exp(D)$. In addition to that, the CT-RDSEM uses $\Sigma = Var(\varepsilon_{ij})$ as model parameters while RDSEM uses $Var(\xi_{ij})$ and the reparameterization between the two is $Var(\xi_{ij}) = \Sigma - R\Sigma R^T$.

Note also that the CT-RDSEM model parameters depend on the scale of the time variables. If the time is multiplied by 2, the drift matrix must be divided by 2 to preserve the model. In order to avoid numerical issues, where the drift matrix is on a very large scale or a very small scale, it is recommended that the time variable is scaled so that the average $\Delta t_{ij} =$

$t_{ij} - t_{i,j-1}$ is approximately 1, or close to that. This means that the time variable should be scaled so that on average the consecutive observations are 1 unit apart. This is not important but would set the drift matrix on a reasonable scale and would avoid D values that are too large or too small.

As described in the introduction, the RDSEM model with the TINTERVAL option provides an approximation to the CT-RDSEM model. If the TINTERVAL is set to δ, the interval length of one period is δ. Essentially, we have rescaled the time so the time span of δ is 1 in RDSEM. To compare the RDSEM to the CT-RDSEM model it can be useful to divide the time values t_{ij} by δ. This way a time unit of one would mean the same in both analyses.

The estimation of the CT-RDSEM model is based on Bayesian MCMC methodology. For the most part, the model is estimated as in the RDSEM model, see Asparouhov et al. (2018) and Asparouhov and Muthén (2023), i.e., using conjugate priors and explicit conditional distributions. The only parameter updating that must be clarified is the updating of D and Σ. Both do not have explicit conditional distribution and we use MH (Metropolis-Hastings) with a random walk jumping rule. The MH method is somewhat crude and often does not provide fast convergence but is robust and simple to implement. Several variations and alternative methods have been implemented in Mplus and are described in the Appendix.

4 Matrix Exponentiation

In this section we provide some further information for matrix exponentiation. The Drift matrix has a central role in continuous time modeling and the transition from Drift to autoregressive parameters is based on matrix exponentiation. Thus, for CT-RDSEM modeling, it is essential to understand matrix exponentiation to some extent. In certain cases matrix exponentiation can be derived explicitly but in most cases it cannot. The two simplest examples are

$$e^{\begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix}} = \begin{pmatrix} e^a & 0 \\ 0 & e^b \end{pmatrix} \tag{16}$$

$$e^{\begin{pmatrix} a & c \\ 0 & b \end{pmatrix}} = \begin{pmatrix} e^a & c(e^a - e^b)/(a-b) \\ 0 & e^b \end{pmatrix} \tag{17}$$

The above formula requires a and b to be different. When a and b are the same we get

$$e^{\begin{pmatrix} a & c \\ 0 & a \end{pmatrix}} = \begin{pmatrix} e^a & ce^a \\ 0 & e^a \end{pmatrix} \tag{18}$$

Formula (16) generalizes to diagonal matrices of any size. Formula (17) generalizes to sparse matrices where the $p-$th column and row combined contain no more than 1 off diagonal value. A general formula for a 2x2 matrix exponentiation is available in Bernstein and So (1993), but that formulation is much less interpretable than the upper triangular form given in (17) and (18). A more general method for matrix exponentiation explicit formula derivation is available in Mouçouf and Said (2020), but that method probably is also not practical for statistical model applications and interpretations.

The above explicit formulas point out that the diagonal entries in the drift matrix are almost exclusively going to be negative numbers because the autoregressive coefficients e^a is less than 1 and thus a must be negative. The general model requires that the residual variance covariance $\Sigma - R_j \Sigma R_j^T$ is always positive definite. If the drift matrix and the auto-regressive matrix are diagonal, the autoregressive parameters must be less than 1 or the residual variance covariance will have negative variance numbers. In that regard we see here that CT-RDSEM is different from RDSEM and DSEM where the autoregressive parameters distribution may exceed 1. In the CT-RDSEM case this cannot happen at least for the diagonal case. Similarly, RDSEM and DSEM can estimate autoregressive coefficients that are negative or at least a part of the posterior distribution is negative. In CT-RDSEM that cannot happen at least in the diagonal case.

The upper triangular form also shows that the cross-lag drift has the same statistical significance as the cross-lag autoregressive parameters since $(e^a - e^b)/(a - b)$ is always positive, and that significance is independent of the time between the observations. The cross-lag drift and the cross-lag autoregressive parameter have the same sign in the upper triangular 2x2 case. In that case we can also determine when the cross-lag reaches its maximum. If

$$D = \begin{pmatrix} a & c \\ 0 & b \end{pmatrix}, \tag{19}$$

the off diagonal entry of e^{Dt} is maximized at

$$t^* = \frac{log(-a) - log(-b)}{(-a) - (-b)}. \tag{20}$$

As discussed earlier, this quantity is often of substantive interest. If a and b are close, t^* is approximately the log derivative $1/(-a)$. Thus, t^* can be a small number if $(-a)$ is large, i.e. the autoregressive coefficient is small, or it can be a large number if $(-a)$ is small, i.e., the autoregressive coefficient is large. This intuitively can be translated as follows. The larger the autoregressive coefficients are, the more delayed the maximization of the cross-lagged effect will be. It is interesting to note here that the cross-lag drift parameter c has no impact on where the maximum cross-lag autoregressive coefficient is. The upper diagonal 2x2 situation can also be used as an approximation in many cases for the full 2x2 VAR model where often at least one of the cross-lagged effects is not statistically significant (and can be fixed to 0).

Another useful fact for matrix exponentiation is the fact that

$$e^{A+B} \neq e^A e^B. \tag{21}$$

Equality holds only for commutative matrices where $AB = BA$. This to a large extent is the reason why a simple computation for matrix exponentiation does not exist, i.e., a matrix cannot be simply decomposed as its diagonal and the various off-diagonal entries.

Additional information for the Drift matrix and its connection to the autoregressive matrix is given in Section 8.

5 Some general comments

The following section is intended to provide some practical advice for estimating CT-RDSEM models in Mplus.

5.1 Large negative drift

If the autoregressive parameters are small the drift parameters become large negative numbers. That can create a problem in terms of identification and convergence. A drift of -100 is typically the same as a drift of -1000 in terms of model and data fit. The parameters in the different MCMC chains can

converge to any number between -100 and -1000 and thus produce unreasonable PSR and convergence problems. If the problematic drift parameter is subject-invariant non-random parameter, this can typically be resolved by providing a weakly informative priors to keep the drift parameter from exploding, for example $N(-1, 1)$ prior for the diagonals, but occasionally a non-diagonal parameter need a weakly informative prior and in those cases the correct value is N(0,1). The FBITER option can be used with say 2000 iterations and the traceplot curves can be inspected to locate convergence problems such as large negative drifts. If a large random drift occurs in the estimation, weakly informative priors will not help. In this case, the solution is to ignore the PSR convergence criterion altogether and instead use the traceplot curves and inspect one at a time after a long enough MCMC sequence is estimated. Clearly some traceplots will be indicating non-convergence but such can be ignored with the explicit understanding that these diverging parameters are essentially yielding the same model. Clearly this requires comprehending fully the intricacies of CT-RDSEM. Model simplifications, such as replacing random drift with non-random drift or removing individual clusters from the analysis can also be used as a solution in this case.

5.2 Larger models

Because the Metropolis-Hastings (MH) algorithm is used in this estimation currently, larger models are likely to be prone to very slow convergence. Currently all drift parameters are updated in one MH step together in a multivariate fashion. The same applies to the total residual variance covariance parameters Σ. It is possible to update one parameter at a time but that is much more computationally expensive, and it has not been found useful. When the model is large, random walk MH steps are likely to end up being small yielding slow convergence. The solution in this case is to gradually build a model and only retain drift parameters minimally when they come significant. Univariate models should be the start.

5.3 Drift parameter trace plots

In the current Mplus algorithm the weak/slow mixing occurs primarily for the drift parameters. Although the residual variance covariance parameters are also estimated via the Metropolis-Hastings algorithm they appear to be mixing much better. All other model parameters, such as intercepts,

regression parameters or between level parameters will have explicit posterior distribution and are thus expected to mix flawlessly. This points out that Drift parameters trace plots should be monitored in addition to the standard PSR convergence criterion.

5.4 Random drift with non-random variance

Simple regression models are robust to heteroskedasticity, meaning that parameter estimates remain unbiased even when the residual variance is not cluster invariant. To a fairly large extent this carries over to DSEM and RDSEM models, although not completely, see Jongerling et al. (2015). The CT-RDSEM model, however, is quite different. Consider for example CT-RDSEM model where observations are equally spaced, do not vary across time or individual and the autoregressive coefficient is subject specific but invariant across time since the observations are equally spaced. In that case the model becomes

$$Y_{it} = \alpha_i + r_i(Y_{i,t-1} - \alpha_i) + \varepsilon_{it}. \tag{22}$$

If we now assume that the variance of $Y_{w,it} = Y_{it} - \alpha_i$ is not subject specific, i.e., it is a parameter v then

$$Var(\varepsilon_{it}) = (1 - r_i^2)v. \tag{23}$$

This model constraint is quite artificial and essentially results in heavy weighting in the likelihood for observations with higher autoregressive coefficients. If the constraint doesn't hold, model parameters will be heavily weighted to fit the data with higher autocorrelations instead of equal weighting. It is therefore recommended that models with random drift, also be estimated with random variances so that the above artificial constraint is not imposed on the model. Using a variety of real data examples, we have noticed the following phenomenon. CT-RDSEM models with random drift often do not match the estimates of the corresponding RDSEM-TINTERVAL model. When random variance is added to the model, the model estimates align and are also much more similar to the RDSEM-TINTERVAL without the random variance, and are not similar to the CT-RDSEM model without the random variance. We therefore conclude from this experimentation that the random variance feature is needed much more in CT-RDSEM models than in RDSEM models. This is an unusual but important finding.

5.5 Data with abrupt changes

Consider now the situation when two consecutive observations for individual i are taken closely in time but are substantially different. One such example that we have found is the case of smoking urge where data is collected at random prompts but also by self-initiated reporting method in an event of a spike in smoking urge. If such data is analyzed as one time series and a random prompt occurs just before a spike in urge, we would have two observations in proximity of time with potentially large outcome differences. Such observations typically will become outliers and with a very heavy influence on the model estimates, which typically will also force the autocorrelation to converge to zero (diagonal drift parameter converging to minus infinity). The explanation is as follows. If the two observations are $Y_{i,j-1}$ and $Y_{i,j}$ observed at times $t_{i,j-1}$ and t_{ij}, and $t_{ij} - t_{i,j-1}$ converges to zero, according to the CT-RDSEM model, the observations $Y_{ij} - Y_{i,j-1}$ must also converge to zero and the variance of $Y_{ij} - Y_{i,j-1}$ must converge to zero as well. If that doesn't happen, when for example abrupt change in outcome happens in proximity of time, the conditional distribution $[Y_{ij}|Y_{i,j-1}]$ will be many times the small residual standard deviation, i.e., an outlier. This outlier will heavily impact the log-likelihood and the only resolution will be to have the autoregressive coefficient be pushed down towards zero or generally small numbers. This issue can be alleviated somewhat if covariates predicting the outcome can be found. Differences in the covariates might be able to explain the abrupt change in the outcome to some extent. In the smoking urge example, one predictive covariate can be the type of prompt used for the observation: random or self-initiated. Another alternative to alleviate this problem is to change time stamps so that all times of observations are apart at least above a minimal value. Many automated data collection methods already contain such minimal value but if multiple data collection methods are used the minimal value may have to be enforced manually at the analyst's discretion. The manual time shift is not a clean and well-defined process because if there are 3 observations or more collected at almost the same time, some observations must be shifted multiple times. At that point the manual shift is engaging in a procedure that is like the TINTERVAL algorithm which by default also sets up the data to have a minimum distance apart (that would be the value in the TINTERVAL command). Another alternative is just to average observations taken in proximity or to simply delete all but one of the approximately repeated observations. The current

Mplus algorithm will not warn if observations are too close in the timeline. It will do so only if the distance between the observations is precisely zero. However, the issue can easily be recognized by comparing results from CT-RDSEM and RDSEM-TINTERVAL. If the autocorrelation of CT-RDSEM is zero and the corresponding RDSEM-TINTERVAL autocorrelation is not, then most likely the data contains observations taken in proximity of time but with very different outcome values.

5.6 The TINTERVAL comparison

There is an easy way to compare the CT-RDSEM model with the dynamic models introduced in Asparouhov et al. (2018). The direct comparison is between CT-RDSEM and the RDSEM model. The TINTERVAL algorithm, described in Asparouhov et al. (2018), discretizes continuous time stamps so that the continuous time dynamic model can be approximated by a discrete time dynamic model. This approximation makes the two models equivalent in interpretation as well as model parameter estimates. Readers familiar with RDSEM can thus easily transition into the CT-RDSEM framework by the existence of this equivalence.

The easiest way to connect the two models is as follows. If the RDSEM model is based on *TINTERVAL=t(δ)*, to obtain an equivalent CT-RDSEM model we replace that specification with *ctime=t; define: t=t/δ;*. With these settings the autoregressive parameters of RDSEM can be found in the autoregressive curves reported by Mplus for time $t = 1$. The effect of δ is substantial on the RDSEM model but has no effect on the CT-RDSEM model. In RDSEM, δ is typically set to be a value nearing the lower end of the distribution of the time distance between consecutive observations. If the value of δ is sufficiently small the RDSEM approximation of CT-RDSEM will be very good. However, when the value of δ is small enough, the analyzed data set becomes large as missing values populate the unobserved slots in the discretized time frame. This causes slow and long computations. In addition, the amount of missing data that is inserted by TINTERVAL may become so large that the MCMC algorithm which imputes all such values may yield slow and protracted mixing and ultimately can get destabilized in the attempt to populate an extraordinarily high dimensional posterior of missing values and parameters. On the other hand, values of δ that are too high provide only a crude approximation to the CT-RDSEM model. This can be recognized easily if the results of δ and 2δ yield autoregressive parameter in univariate models

that are not the square of each other. That is a property of the CT-RDSEM model that never fails. Doubling the distance between observations always results in the autoregressive parameter in univariate models being squared. In some complex RDSEM application a proper value of δ might not be found at all. That is, sufficiently small values that can yield good approximation do not converge due to large amount of missing data.

In some cases the data has integer time stamps, such as daily observations. In such circumstances, if the model is estimated with *TINTERVAL=t($\delta = 1$)*, the time stamp for the RDSEM and CT-RDSEM model are the same. This implies that the models are the same and typically the estimates match very closely, apart from minor variations due to the random nature of the Bayes estimation. Occasionally, however, differences in the estimation can be attributed to the different parameterization and the fact that CT-RDSEM does not accommodate autoregressive values such as negative values or values bigger than 1. Differences between the two methods can occur also because of different treatment of the initial period when the time-series data is short.

It should be noted here again that CT-RDSEM model and the RDSEM model also differ in the residual variance covariance, but that is not because the models are different but because the parameterization is different. Residual variance in the CT-RDSEM model is the total variance Σ of the residual which includes the autoregressive part. The residual variance parameter in the RDSEM model is the variance of the residual excluding the autoregressive part. In the CT-RDSEM model that is not a model parameter but a quantity that varies with the continuous time and can become zero when the time between observations converges to zero.

Finally, the equivalence between RDSEM and CT-RDSEM can be used in those situations where CT-RDSEM convergence is too slow. The CT-RDSEM models can be used to fine tune the proper TINTERVAL with simpler univariate models but if a full complex CT-RDSEM model does not converge, the RDSEM model can be used instead. This synergy between the two methodologies needs further exploration and development.

5.7 The advantage of the lack of initial condition

The RDSEM and DSEM models, being invariant over time, model the starting equation in the same way as the equations at any other time point using a predictor from the previous time point. This implies that at time point 0 a missing observation (latent variable) is used as a predictor. That ini-

tial value is imputed by the MCMC algorithm and for models with 10 or more observations per person has little to no impact on the overall estimation. For shorter time series models, however, the assumptions regarding the initial value do impact the estimation. The precise assumption about that initial value is different in the RDSEM / DSEM frameworks and even when equivalent models are estimated with short (< 10) observations per person a noticeable difference in results can be seen between the two frameworks. The assumptions also cause the model estimates to be different than the ML estimates obtained for the model when the estimation is conducted in a wide format similarly to the RI-CLPM.

The CT-RDSEM model has an advantage in that regard. The model for each observation is time specific and no time invariance is assumed or used in the estimation. The first observation simply does not have an autoregressive predictor but is simply given as in equation (6). This avoids the initial concept altogether. The result of this is that CT-RDSEM yields identical log-likelihood as ML wide format. In turn, the CT-RDSEM model performs well even with a very small number of observations per person and is expected to outperform RDSEM and DSEM.

The wide ML estimation for the autoregressive model with random intercept and continuous time of observations requires setting up autoregressive structure via model constraint and it can be pieced together from User's Guide (Muthén and Muthén (1998-2017)) example 6.17, using the Tscore command from example 6.12, and the subject specific constraints described in example 5.23. As such, the setup of this model with ML is much more complex than the CT-RDSEM model and the input would increase substantially when the number of observations per person increases substantially. The data would also need to be reformatted so that it comes in a wide format and missing values must be added at the end of the time series so that all individuals have an equal number of observations.

Small number of observations per cluster can also hinder comparisons between CT-RDSEM and RDSEM-TINTERVAL as the initial conditions assumption will become influential. In most such cases CT-RDSEM should be preferred.

5.8 The bounded posterior for autoregressive parameters

Comparisons between CT-RDSEM and RDSEM-TINTERVAL can also be invalidated by the fact that the posterior distribution of the autoregressive parameters in CT-RDSEM are bounded between 0 and 1. In RDSEM-TINTERVAL there are no such constraints. This can be seen typically when the sample is small or when the autoregressive parameters are random and the number of observations per person is not large. In both of these situations the posterior distribution of the autoregressive parameters is wider and can cross outside of the (0,1) interval in the RDSEM framework.

5.9 Time and variable scale

The scale of the times of observations is not important theoretically but from a numerical perspective it would be preferable for the time scale to be somewhat standardized so that the average distance between consecutive observations is approximately 1. This will enable us to keep the Drift parameters which are directly multiplied by the time variable from exploding to large values and potentially appearing as non-converging. Although Mplus does consider the uncertainty in the scale of the time, it can potentially become an issue in certain situations. Standardizing the time scale would also improve the readability of the autoregressive curves.

The scale of the variables Y_{ij} is also typically not important. However, the MH algorithm has a starting value of the size of the random jumps, which is 0.1. The MH algorithm quickly adjusts the size of the jumps to balance the acceptance / rejection probabilities. However, if the variables are on too large or too small of a scale, this process unnecessarily hinders the estimation. Thus, it is advisable for the scale of the variables to be approximately on a standardized scale. The drift parameters have their own scale, which is only connected to the time scale. Therefore standardizing the variables is unlikely to help with drift parameter estimation. It can help, however, with residual variance covariance estimation which is also estimated with the MH algorithm.

5.10 Small data convergence issues

Small data typically yields wide posterior distribution for the parameters. The MH algorithm therefore will take longer to populate the posterior. In addition, the fringe parts of the posterior distribution are sometimes hard to escape and convergence can become painfully slow. To avoid such issues the complexity of the model should match the amount of data. Random drift and residual variances should be used with larger samples only.

5.11 Starting values

The CT-RDSEM model uses starting values of -1 for all diagonal drift parameters and zero for the off-diagonal parameters. Changing the diagonal starting values to 0 or positive values will quickly cause convergence issues. It is important to understand that the drift parameters scale is different from that of the auto-regressive parameter scale and not introduce inappropriate starting values.

5.12 The burnin phase

The burning phase in the MH algorithm is the phase of the estimation where the jumps in each MCMC step are adjusted to yield optimal acceptance rejection probabilities. The default of the algorithm is 1000 iterations. Simpler examples can be estimated with shorter bunrin phase such as 100 iterations. More complex examples might benefit from a longer than 1000 burnin phase. There is a limit to how much a burnin phase can help the estimation. Regardless of how long the burnin phase is set to be, the size of the jumps are eventually adjusted to an optimal size. Even if the burnin phase is large, the size of the jumps will stop changing. Mplus uses as a minimal number of iterations twice the burnin phase. Therefore, increasing the burnin phase to extreme values will only protract the convergence.

5.13 Some limitations

The CT-RDSEM model allows only lag-1 models. This technically is not a precise statement. If the time between consecutive observations is 1, then Y_t is regressed on Y_{t-1}. If the time between the consecutive observations is 2, then Y_t is regressed on Y_{t-2}, which then resembles the RDSEM model

with lag 2. The CT-RDSEM model is a lag-1 model in the sense that only one previous observation is used as a predictor and that there is just one autoregressive parameter. The regression parameter of Y_t on Y_{t-2} is not an independent parameter from the regression parameter of Y_t on Y_{t-1} but is the square of that.

Various other modeling features are not developed yet in Mplus, such as factor models and models with non-continuous outcomes. Some of these have been developed already elsewhere, see Driver et al. (2017), Boker et al. (2004) etc.

5.14 Missing data

In the CT-RDSEM framework, observations with missing values on all dependent variables are automatically removed from the data. Such observations do not contribute to the model likelihood and if they are included the convergence will be slowed down because the additional missing values would need to be imputed. This is in stark contrast with the RDSEM-TINTERVAL framework where observations with missing values for all dependent variables are inserted in the analyzed data.

In the CT-RDSEM framework, observations are included if some of the outcomes are missing, and some are present. The missing outcomes are imputed during the estimation.

6 Simulation Studies

We illustrate the CT-RDSEM methodology with some key examples. First we consider the bivariate VAR (vector autoregressive) model. Then we consider the VAR model with random drift parameters. We also describe the CT-RDSEM path analysis model, where contemporaneous regression is estimated and both the endogenous and the exogenous variables are autoregressive. This model also features latent centering for the covariate, i.e., both variables have random intercepts. We also consider a latent growth model in the CT-RDSEM framework. In this model, the continuous times of observations, which are also individually specific, affect not just the latent growth trajectory but also the residual autocorrelation structure. This is a key model that is difficult to implement in the maximum-likelihood framework in Mplus. We also include simulation studies designed to directly compare CT-RDSEM

and RDSEM-TINTERVAL methodologies.

6.1 The CT-RDSEM VAR model

In this section we consider the bivariate CT-RDSEM VAR model. Let Y_{1ij} and Y_{2ij} be the two outcomes for individual i at times t_{ij}. The model is given as follows. For $p = 1, 2$

$$Y_{pij} = Y_{wpij} + Y_{bpi} \tag{24}$$

$$\begin{pmatrix} Y_{b1i} \\ Y_{b2i} \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \sigma_{b11} & \sigma_{b12} \\ \sigma_{b12} & \sigma_{b22} \end{pmatrix} \right) \tag{25}$$

$$\begin{pmatrix} Y_{w1ij} \\ Y_{w2ij} \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{w11} & \sigma_{w12} \\ \sigma_{w12} & \sigma_{w22} \end{pmatrix} \right) \tag{26}$$

$$Y_{w1ij} = r_{ij11} Y_{w1i,j-1} + r_{ij12} Y_{w2i,j-1} + \xi_{1ij} \tag{27}$$

$$Y_{w2ij} = r_{ij21} Y_{w1i,j-1} + r_{ij22} Y_{w2i,j-1} + \xi_{2ij} \tag{28}$$

$$R_{ij} = \begin{pmatrix} r_{ij11} & r_{ij12} \\ r_{ij21} & r_{ij22} \end{pmatrix} = Exp((t_{ij} - t_{i,j-1})D) \tag{29}$$

$$D = \begin{pmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{pmatrix}. \tag{30}$$

The above equations show the model for $j > 1$. For $j = 1$, we can assume that $t_{i0} = -\infty$ and thus $R_{i1} = 0$ and the predictor from period 0 is multiplied by 0 and not a part of the model.

The above model relates to the general CT-RDSEM model as follows. In equation (12), when both regression matrices B_w and Γ_w are zero, then $Y_{wij} = \varepsilon_{ij}$. In the general model these are autoregressed as in equation (7), which is precisely what equations (27-28) represent.

For this simulation study we generate data using $N = 50$ individuals with $T = 30$ observations per person. Random times of observations are generated as follows $t_{i1} = 0$ and $t_{ij} = t_{ij-1} + \zeta_{ij}$ where $\zeta_{ij} \sim IG(3, 5)$. Mplus uses the inverse gamma distribution to generate the distances between consecutive observations. In this simulation the average distance between consecutive observations is thus 2.5. The mean of the $IG(a, b)$ distribution is $b/(a-1)$, while the mode is $b/(a+1)$. The distribution is generally skewed to the right.

We use 100 replications in this simulation. Each replication takes about 5 seconds to estimate the VAR model. The results of the simulation are

Table 1: CT-RDSEM Var model

Parameter	True Value	Abs. Bias	Coverage
σ_{w11}	1.1	.01	.92
σ_{w12}	0.5	.01	.90
σ_{w22}	1.3	.01	.93
σ_{b11}	1.2	.20	.89
σ_{b12}	0.4	.05	.96
σ_{b22}	0.5	.05	.93
μ_1	3	.01	.93
μ_2	.3	.01	.97
d_{11}	-0.5	.01	.98
d_{12}	.2	.00	.93
d_{21}	.1	.00	.88
d_{22}	-0.6	.00	.89

reported in Table 1. The estimates have minimal bias, and the coverage is near the nominal levels. The only parameters that show larger biases are the variance covariance on the between level, which is related to the fact that there are only 50 individuals in the data. If we increase the sample to 100 individuals, the bias is decreased by half and the coverage is above 93%. The Mplus input file for this simulation is given in Figure 1. The input is the same as for the RDSEM model, with the only addition here being the *CTIME* option which specifies how the time increments are generated.

6.2 The CT-RDSEM VAR model with random drift

In this section we conduct a simulation study similar to the one in the previous section but we now introduce subject specific random drift parameters, in addition to the subject specific random intercepts. The drift parameters for individual i are between-level random effects

$$d_{ip_1p_2} \sim N(d_{p_1p_2}, v_{p_1p_2}). \tag{31}$$

In this simulation study, we did not include covariance between the random drift parameters and between the random drift parameters and the random

Figure 1: CT-RDSEM VAR simulation study

```
MONTECARLO:  NAMES ARE y1-y2;
             NOBS = 1500;
             NREP = 100;
             NCSIZES = 1;
             CSIZES = 50(30);
             lagged=y1-y2(1);
             CTIME=IG(3,5);

ANALYSIS:   TYPE IS TWOLEVEL;
            estimator=bayes; proc=2;
            burnin=100;

MODEL MONTECARLO:

  %WITHIN%
  y1^ on y1^1*-0.5;
  y2^ on y2^1*-0.6;
  y1^ on y2^1*0.2;
  y2^ on y1^1*0.1;
  y1*1.1; y2*1.3;
  y1 with y2*0.5;
  %BETWEEN%
  y1*1.2 y2*0.5;
  Y1 WITH Y2*0.4;
  [Y1*3 Y2*0.3];

MODEL:

  %WITHIN%
  y1^ on y1^1*-0.5;
  y2^ on y2^1*-0.6;
  y1^ on y2^1*0.2;
  y2^ on y1^1*0.1;
  y1*1.1; y2*1.3;
  y1 with y2*0.5;
  %BETWEEN%
  y1*1.2 y2*0.5;
  Y1 WITH Y2*0.4;
  [Y1*3 Y2*0.3];
```

intercepts, for both the data generating model and the model estimation. In real data analysis such covariances may need to be explored, however, not just for the purpose of correctly specifying the model but also for the inference that may be found for the various random effects. A detailed discussion on the impact of random effect covariance is available in Asparouhov and Muthén (2024). It is shown there that for larger cluster sizes (such as time-series models), incorrectly specifying the random effect distribution has minimal impact on the model estimation.

In this simulation, the sample has 100 individuals with 100 observations each. The time increments are generated from IG(3,3) with a mean value of 1.5 and 95% of the values between 0.4 and 4.5. We generate and analyze 100 samples and each analysis takes approximately 2 minutes to converge. The results of the simulation study are given in Table 2. The bias is minimal and the coverage is near the nominal level. Figure 2 shows the input file for this simulation.

Note here that because the variance of the random drifts is small, the distribution of the diagonal drift parameters remains negative across the data generation part of the simulation. This will generally guarantee that for each individual, the drift matrix yields an admissible solution. If we increase the variance of the random effects, however, in some instances the generated drift matrix will be inadmissible and will be discarded. This creates a situation where the model generating distribution is cut to only a part of the desired distribution. In such cases, the estimated distribution for the drift parameters is expected to match the cut distribution mean and variance and not those specified in the data generation model. This may appear as a bias in the final results but it is not. The difference between the estimated values and the generating values would be attributed to the difference between the desired generating distribution and the portion of that distribution that yields admissible drift matrices.

6.3 The CT-RDSEM path analysis model

In this section we consider the bivariate CT-RDSEM path analysis model. This model is a simple regression between two variables Y_{1ij} and Y_{2ij} where times of observations are subject specific and continuous. Both variables have autocorrelation across time. The two variables are decomposed as the sum of the time invariant component (random intercept, i.e., the between part of the variable) and the remaining time specific residual (within part of the

Table 2: CT-RDSEM Var model with random drift

Parameter	True Value	Abs. Bias	Coverage
σ_{w11}	1.1	.00	.94
σ_{w12}	0.6	.00	.94
σ_{w22}	1.3	.00	.96
σ_{b11}	1.2	.04	.97
σ_{b12}	0.4	.03	.94
σ_{b22}	1.0	.05	.96
μ_1	3	.00	.91
μ_2	2	.00	.93
d_{11}	-1.0	.00	.95
d_{12}	-.2	.00	.95
d_{21}	0	.00	.89
d_{22}	-0.9	.01	.88
v_{11}	.2	.02	.88
v_{12}	.2	.02	.88
v_{21}	.2	.02	.91
v_{22}	.2	.02	.97

Figure 2: CT-RDSEM VAR with random drift simulation study

```
MONTECARLO:  NAMES ARE y1-y2;
             NOBS = 10000;
             NREP = 100;
             NCSIZES = 1;
             CSIZES = 100(100);
             lagged=y1-y2(1);
             CTIME=IG(3,3);

ANALYSIS:   TYPE IS TWOLEVEL random;
            estimator=bayes; proc=2;

MODEL MONTECARLO:

  %WITHIN%
  s1 | y1^ on y1^1;
  s2 | y2^ on y2^1;
  s3 | y1^ on y2^1;
  s4 | y2^ on y1^1;
  y1*1.1; y2*1.3;
  y1 with y2*0.6;
  %BETWEEN%
  [y2*2]; y2*1;
  [Y1*3]; y1*1.2;
  y1 with y2*0.4;
  s1-s4*0.02;
  [s1*-1 s2*-0.9 s3*-0.2 s4*0];

MODEL:

  %WITHIN%
  s1 | y1^ on y1^1;
  s2 | y2^ on y2^1;
  s3 | y1^ on y2^1;
  s4 | y2^ on y1^1;
  y1*1.1; y2*1.3;
  y1 with y2*0.6;
  %BETWEEN%
  [y2*2]; y2*1;
  [Y1*3]; y1*1.2;
  y1 with y2*0.4;
  s1-s4*0.02;
  [s1*-1 s2*-0.9 s3*-0.2 s4*0];
```

variable). The regression/path analysis consists of a within level regression and a between level regression, i.e., two separate regression equations. The regression between the time invariant components (random intercepts) is the between level regression. The time specific components of the variables are also regressed on each other (within level regression). The model essentially features latent centering for the predictors. Latent centering is an important concept in multilevel modeling, see Asparouhov and Muthén (2006), Ludke et al. (2008), Asparouhov and Muthén (2019), and in DSEM and RDSEM modeling Asparouhov et al. (2018), and Asparouhov and Muthén (2020). All the components of this multilevel model are important to obtain unbiased estimates for the regression parameters: separating the within and the between level regressions and allowing for autocorrelation in both the predictor and the predicted variables which properly accommodates the subject specific times of observations.

The model is given as follows. For $p = 1, 2$

$$Y_{pij} = Y_{wpij} + Y_{bpi} \tag{32}$$

$$Y_{b2i} = \mu_2 + \beta_b Y_{b1i} + \varepsilon_{b2i} \tag{33}$$

$$Y_{b1i} \sim N(\mu_1, \sigma_{b1}), \varepsilon_{b2i} \sim N(0, \sigma_{b2}) \tag{34}$$

$$Y_{w2ij} = \beta_w Y_{w1ij} + \varepsilon_{w2ij} \tag{35}$$

$$Y_{w1ij} \sim N(0, \sigma_{w1}), \varepsilon_{w2ij} \sim N(0, \sigma_{w1}) \tag{36}$$

$$Y_{w1ij} = Exp(d_1(t_{ij} - t_{i,j-1}))Y_{w1i,j-1} + \xi_{w1ij} \tag{37}$$

$$\varepsilon_{w2ij} = Exp(d_2(t_{ij} - t_{i,j-1}))\varepsilon_{w2i,j-1} + \xi_{w2ij} \tag{38}$$

We conduct a simulation study using 100 replications with a sample size of 100 individuals and 10 observations per person. The distance between consecutive observations is generated with IG(3,3), which yields a mean value of 1.5. The results are given in Table 3. The bias is minimal, and the coverage is near the nominal levels. The input file for this simulation study is given in Figure 3. At the mean of the data time increments the autocorrelation here for the first variable is 0.64 and for the second variable it is 0.41.

Figure 3: CT-RDSEM path analysis simulation study

```
MONTECARLO:  NAMES ARE y1-y2;
             NOBS = 1000;
             NREP = 100;
             NCSIZES = 1;
             CSIZES = 100(10);
             lagged=y1-y2(1);
             CTIME=IG(3,3);

ANALYSIS:   TYPE IS TWOLEVEL;
            estimator=bayes; proc=2;
            burnin=100;

MODEL MONTECARLO:

  %WITHIN%
  y1^ on y1^1*-0.3;
  y2^ on y2^1*-0.6;
  y2 on y1*1;
  y1*1.1; y2*1.3;
  %BETWEEN%
  y2 on y1*-0.5;
  y1*1.2 y2*0.5;
  [Y1*1 Y2*2];

MODEL:

  %WITHIN%
  y1^ on y1^1*-0.3;
  y2^ on y2^1*-0.6;
  y2 on y1*1;
  y1*1.1; y2*1.3;
  %BETWEEN%
  y2 on y1*-0.5;
  y1*1.2 y2*0.5;
  [Y1*1 Y2*2];
```

Table 3: The CT-RDSEM path analysis model

Parameter	True Value	Abs. Bias	Coverage
σ_{w1}	1.1	.03	.94
σ_{w2}	1.3	.00	.95
β_w	1	.01	.97
σ_{b1}	1.2	.07	.97
σ_{b2}	0.5	.11	.95
β_b	-0.5	.03	.98
μ_1	1	.01	.97
μ_2	2	.02	.97
d_1	-0.3	.02	.95
d_2	-0.6	.01	.91

6.4 Linear Growth Model

Linear growth model is a standard latent variable model, but it can have two levels of complications with continuous times of observations. First, the growth factor is multiplied by a subject specific time score, which means that it is no longer a loading parameter that can be fixed to the time of observations. In Mplus this can be resolved using type=random which steps outside the realm of SEM as loadings are subject specific, or it can be resolved by estimating the model as a two-level model where the growth factor is a random slope for the covariate of time. Both approaches use the EM-algorithm with MLE. The second complication arises from the inclusion of residual autocorrelation which must also account for the individually specific times of observation. The model can be described as follows

$$Y_{ij} = \alpha_i + \beta_i t_{ij} + \varepsilon_{ij} \tag{39}$$

$$Cov(\varepsilon_{ij_1}, \varepsilon_{ij_2}) = \sigma Exp(d|t_{ij_1} - t_{ij_2}|) \tag{40}$$

$$\begin{pmatrix} \alpha_i \\ \beta_i \end{pmatrix} \sim N\left(\begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{12} & \phi_{22} \end{pmatrix} \right) \tag{41}$$

In Mplus currently it is not possible to do a direct simulation study on this model because the time variable cannot directly be used in the model as

a covariate. It is therefore necessary to conduct a two-step simulation study. The first step is to generate the data and the second step is to analyze it. This is accomplished with the external montecarlo feature which allows us to analyze multiple data sets with the same model. In this simulation, we use samples with 100 individuals and 10 observations for every individual. The random times of observation are obtained using time increments generated from IG(10,3) with a mean of 0.33. The first step in the simulation study is given in Figure 4 and the second step is in Figure 5. In Step 1 we generate the Y variable excluding the growth component. The variable s which will be used in the second step to augment the data is generated here as an observed between level variable. All the generated data is saved in this step, meaning that all 100 data sets are stored. In the second step the variable Y is adjusted in the defined statement by adding $s \cdot t$, which is the growth factor. Here s is the variable generated in the first step intended to be used as the growth factor. Also, in the defined statement we make a copy of the time variable which will be used in the model. In the model, the growth factor s here is a latent unobserved variable. The results of the simulation study are given in Table 4. The bias in all parameter estimates is minimal and the coverage is near the nominal level.

It should be noted here that the parameter settings in this simulation study are not quite selected completely at random. The autocorrelation at the average increment of time is 0.52. There are three different modeling components that are competing to fit the correlation among the observed variables: the random intercept, the random slope and the autocorrelation. These modeling components are well differentiated if the number of observations per individual is infinitely large but with smaller numbers of observations, they overlap quite a bit. For example, autocorrelation that nears 1 when the sample is small can be modeled only with the random intercept. In that regard, for smaller time-series situations, the likelihood of the model becomes somewhat flat and the model itself becomes poorly identified. Thus, not all models of this kind will yield unbiased estimates at $T = 10$. Some parameter values will need many observations per person.

Figure 4: Linear growth model simulation study: Step 1

```
MONTECARLO:  NAMES ARE y s;
             NOBS = 2000;
             NREP = 100;
             NCSIZES = 1;
             CSIZES = 100(10);
             lagged=y(1);
             CTIME=IG(10,3);
             between=s;
             repsave=all;
             save=a*.dat;


ANALYSIS:   TYPE IS TWOLEVEL;
            estimator=bayes; proc=2;
            burnin=100;

MODEL MONTECARLO:

  %WITHIN%
  y^ on y^1*-2;
  y*1.1;
  %BETWEEN%
  y*1.2; s*0.1;
  [Y*1 s*0.2];
  y with s*-0.2;

MODEL:

  %WITHIN%
  y^ on y^1*-2;
  y*1.1;
  %BETWEEN%
  y*1.2; s*0.1;
  [Y*1 s*0.2];
  y with s*-0.2;
```

Figure 5: Linear growth model simulation study: Step 2

```
data: file is alist.dat;
      type=montecarlo;

variable:    NAMES ARE ss y id t;
             lagged=y(1);
             cluster=id;
             CTIME=T;
             usevar=y t2;
             within=t2;

define: y=y+ss*t;
        t2=t;

ANALYSIS:   TYPE IS TWOLEVEL random;
            estimator=bayes; proc=2;
            burnin=100;

MODEL:

  %WITHIN%
  y^ on y^1*-2;
  s | y on t2;
  y*1.1;

  %BETWEEN%
  y*1.2; s*0.1;
  [Y*1 s*0.2];
  y with s*-0.2;
```

Table 4: Linear growth model

Parameter	True Value	Abs. Bias	Coverage
α	1	.00	.95
β	0.2	.00	.96
d	-2	.03	.92
ϕ_{11}	1.2	.12	.93
ϕ_{12}	-0.2	.02	.91
ϕ_{22}	0.1	.01	.95
σ	1.1	.00	.90

6.5 Comparison between CT-RDSEM and RDSEM using RDSEM data

In this section we compare the estimate of the CT-RDSEM and RDSEM methods in the situation when the data is generated as RDSEM, i.e., the times of observations are integer. We consider the simple RDSEM model

$$Y_{ij} = \alpha_i + \beta X_{ij} + \varepsilon_{ij} \tag{42}$$

$$\varepsilon_{ij} = r\varepsilon_{i,j-1} + \xi_{ij} \tag{43}$$

$$\xi_{ij} \sim N(0, \sigma), \alpha_i \sim N(\alpha, v). \tag{44}$$

The data is generated with this RDSEM model and analyzed with both the RDSEM and CT-RDSEM models. The samples have 100 individuals with 30 observations. The covariate is standard normal. The results of the simulation are presented in Table 5. Both methods perform equally well, and the differences are minimal. The RDSEM estimation setup is given in Figure 6 and the CT-RDSEM estimation is given in Table 7. In the CT-RDSEM estimation, model constraints are used to perform the reparameterization from drift to autoregressive coefficient and from the total residual variance to the residual variance.

Figure 6: RDSEM estimation on RDSEM data

```
MONTECARLO:  NAMES ARE y x;
             NOBS = 3000;
             NREP = 100;
             NCSIZES = 1;
             CSIZES = 100(30);
             lagged=y(1);
             repsave=all;
             within=x;
             save=a*.dat;


ANALYSIS:   TYPE IS TWOLEVEL;
            estimator=bayes; proc=2;
            burnin=100;

MODEL MONTECARLO:

  %WITHIN%
  y^ on y^1*0.5;
  y*1.1;
  y on x*1;
  x*1;
  %BETWEEN%
  y*1.2;
  [y*1];

MODEL:

  %WITHIN%
  y^ on y^1*0.5;
  y*1.1;
  y on x*1;
  %BETWEEN%
  y*1.2;
  [y*1];
```

Figure 7: CT-RDSEM estimation on RDSEM data

```
data: file is alist.dat;
      type=montecarlo;

variable:    NAMES ARE y x id t;
             lagged=y(1);
             cluster=id;
             CTIME=T;
             within=x;

ANALYSIS: TYPE IS TWOLEVEL random;
          estimator=bayes; proc=2;
          burnin=100;

MODEL:

  %WITHIN%
  y^ on y^1*-0.693 (d);
  y*1.466 (v);
  y on x*1;
  %BETWEEN%
  y*1.2;
  [y*1];

model constraint:
  new(r*0.5); r=exp(d);
  new(sig*1.1); sig=(1-r^2)*v;
```

Table 5: Comparison between CT-RDSEM and RDSEM using RDSEM data: Abs. Bias(Coverage)

Parameter	True Value	RDSEM	CT-RDSEM
r	0.5	.00(.96)	.00(.97)
β	1	.00(.94)	.00(.97)
σ	1.1	.00(.96)	.00(.96)
α	1	.03(.94)	.03(.97)
v	1.2	.01(.95)	.04(.95)

6.6 Comparison between CT-RDSEM and RDSEM using CT-RDSEM data

In this section we compare the estimate of the CT-RDSEM and RDSEM-TINTERVAL methods in the situation when the data is generated as CT-RDSEM, i.e., the times of observations are continuous. We consider the simple CT-RDSEM model

$$Y_{ij} = \alpha_i + \beta X_{ij} + \varepsilon_{ij} \tag{45}$$

$$\varepsilon_{ij} = Exp(d(t_{ij} - t_{i,j-1}))\varepsilon_{i,j-1} + \xi_{ij} \tag{46}$$

$$\varepsilon_{ij} \sim N(0, \sigma), \alpha_i \sim N(\alpha, v). \tag{47}$$

The data is generated with this CT-RDSEM model and analyzed with both the RDSEM-TINTERVAL and CT-RDSEM models. The samples have 100 individuals with 30 observations. The time increments are generated from the $IG(3, 5)$ distribution with a mean of 2.5. The covariate is standard normal. For the TINTERVAL parameter we use two values 1 and 0.5. The results of the simulation are presented in Table 6. The three methods perform equally well and the differences are minimal except for the drift parameter. The RDSEM estimation with TINTERVAL of 1 yields biased estimate and lower coverage for the drift. When the TINTERVAL is lowered to 0.5 the model approximation is more accurate, and the drift estimate is nearly unbiased and the coverage recovers almost to nominal levels. This emphasizes the importance of using a low enough TINTERVAL value with RDSEM when the data comes with continuous times of observations.

Table 6: Comparison between CT-RDSEM and RDSEM using CT-RDSEM data: Abs. Bias(Coverage)

Parameter	True Value	CT-RDSEM	RDSEM(1)	RDSEM(0.5)
d	-0.7	.00(.91)	.10(.30)	.03(.85)
β	1	.00(.90)	.00(.92)	.00(.91)
σ	1.1	.00(.92)	.00(.92)	.01(.94)
α	1	.02(.92)	.02(.93)	.02(.93)
v	1.2	.02(.98)	.03(.97)	.02(.97)

The CT-RDSEM estimation setup is given in Figure 8, the RDSEM estimation with TINTERVAL of 1 is given in Figure 9, and the RDSEM estimation with TINTERVAL of 0.5 is given in Figure 10. In the RDSEM estimations model constraints are used to perform the reparameterization from autoregressive coefficient to drift parameter and from the residual variance to the total residual variance.

7 The autoregressive curves

In this section we illustrate the auto-regressive curves utility implemented in Mplus. We use data generated from the VAR model discussed earlier except that one of the cross-lagged effects is changed from 0.1 to 0. The input file for obtaining the autoregressive curves is given in Figure 11. The command to request the curves is PLOT: TYPE = DRIFT (0, 5, 0.1). The curves will be computed for all time values from 0 to 5 with increments of 0.1, i.e., the autoregressive coefficients are computed for $t = 0, 0.1, 0.2, ..., 4.9, 5$ using the usual formula $AR = Exp(tD)$ where D is the estimated drift matrix. Even without that command, the autoregressive curves are computed and reported in the output and if the generic PLOT: TYPE IS PLOT3 command is used the curves are available in the graphics utility. However, the DRIFT option can be used to specify the entire region we want the curves to be plotted in. If the DRIFT option is not used, the autoregressive curves will be computed and plotted in the region that occurs in the data. To be more specific, Mplus computes the 5-th and 95-th percentiles of the distribution of the time increments and uses that as the region where the curves are

Figure 8: CT-RDSEM estimation on CT-RDSEM data

```
MONTECARLO:  NAMES ARE y x;
             NOBS = 3000;
             NREP = 100;
             NCSIZES = 1;
             CSIZES = 100(30);
             lagged=y(1);
             CTIME=IG(3,5);
             repsave=all;
             within=x;
             save=a*.dat;


ANALYSIS:   TYPE IS TWOLEVEL;
            estimator=bayes; proc=2;
            burnin=100;

MODEL MONTECARLO:

  %WITHIN%
  y^ on y^1*-0.7;
  y*1.1;
  y on x*1;
  x*1;
  %BETWEEN%
  y*1.2;
  [y*1];

MODEL:

  %WITHIN%
  y^ on y^1*-0.7;
  y*1.1;
  y on x*1;
  %BETWEEN%
  y*1.2;
  [y*1];
```

Figure 9: RDSEM estimation on CT-RDSEM data using TINTERVAL of 1

```
data: file is alist.dat;
      type=montecarlo;

variable:    NAMES ARE y x id t;
             lagged=y(1);
             cluster=id;
             Tinterval=T(1);
             within=x;

ANALYSIS:   TYPE IS TWOLEVEL random;
            estimator=bayes; proc=2;

MODEL:

  %WITHIN%
  y^ on y^1*0.497 (r);
  y*0.825 (sig);
  y on x*1;
  %BETWEEN%
  y*1.2;
  [y*1];

model constraint:
  new(d*-0.7); d=log(r);
  new(v*1.1); v=sig/(1-r^2);
```

Figure 10: RDSEM estimation on CT-RDSEM data using TINTERVAL of 0.5

```
data: file is alist.dat;
      type=montecarlo;

variable:    NAMES ARE y x id t;
             lagged=y(1);
             cluster=id;
             Tinterval=T(0.5);
             within=x;

ANALYSIS:   TYPE IS TWOLEVEL random;
            estimator=bayes; proc=2;

MODEL:

  %WITHIN%
  y^ on y^1*0.705 (r);
  y*0.554 (sig);
  y on x*1;
  %BETWEEN%
  y*1.2;
  [y*1];

model constraint:
  new(d*-0.7); d=log(r)/0.5;
  new(v*1.1); v=sig/(1-r^2);
```

plotted. In that regard, this region is also important as it reflects the actual autoregressive effects that occur in the data that was analyzed.

The autoregressive curves are given in Figure 12. The diagonal entries start at 1 and gradually decrease towards 0. The off diagonal entries start at 0 and also converge towards zero but peak somewhere in the middle. The confidence intervals are also given for all the curves. It should be noted here that significance does not change across time. AR parameters that are significant at one point are significant at all time points. AR parameters that are not significant at one point are not significant at all time points. The significance of the off-diagonal AR parameters matches that of the corresponding drift parameters. The cross-lagged effect for the first variable appears to be decreasing slower than the diagonal autoregressive coefficient and at time T=5, the off-diagonal effect is bigger than the diagonal in this example. Because one of the effects is not significant and nearly 0 we can use formula (20) to compute the time t^* for which the cross-lagged effect is maximized. Using that formula and the two diagonal drift parameters estimates -0.5 and -0.548 we obtain that the significant cross-lagged effect is maximized at $t^* = 1.9$. That is confirmed also visually by Figure 12.

8 Interpretation of the drift matrix

The drift matrix as described so far is difficult to interpret without being converted to autoregressive coefficients of autoregressive curves. However, a proper and meaningful interpretation of the matrix does exist. It is based on differential equations and we purposefully have avoided differential equations so far. In this section we provide a brief description of the origin of the drift matrix.

Let $Y(t)$ represent the continuous process of the outcome variable in terms of the time variable t. Even though statistical data analysis is always based on a discrete sample from that curve, in reality the model does estimate an actual curve as a continuous function of time. The continuous time autoregressive process implies that

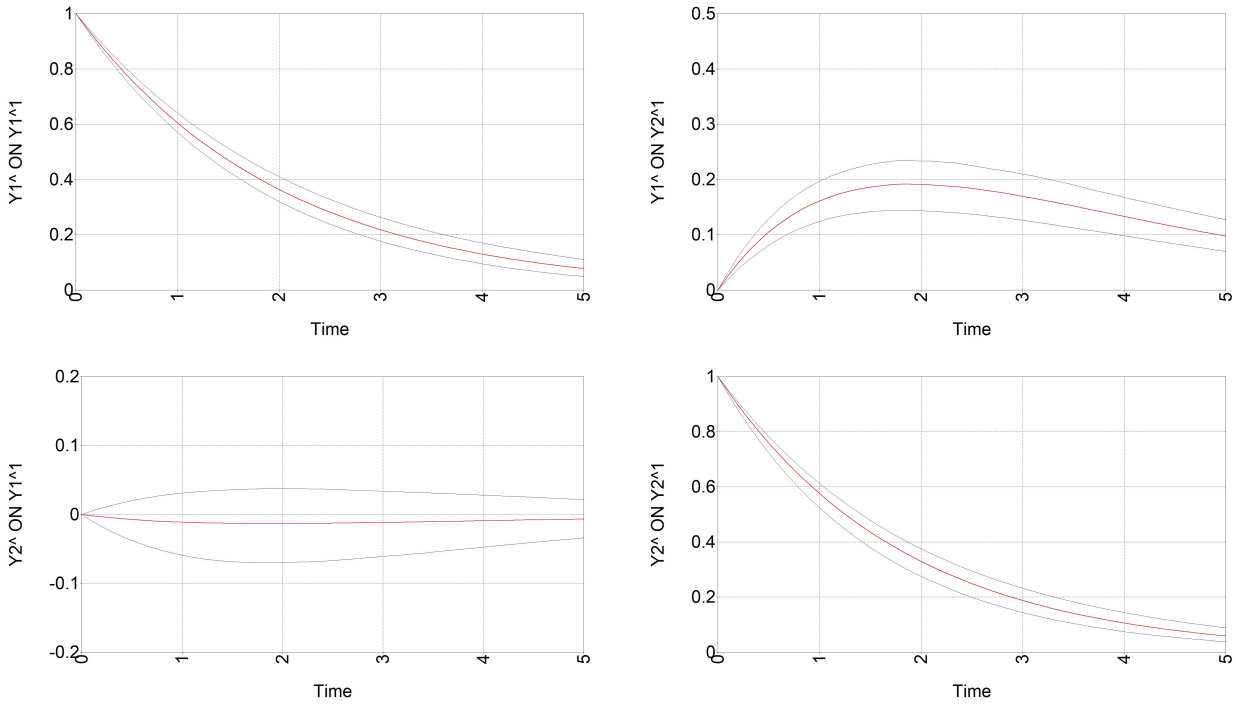
$$F(t) = E(Y(t)|Y(0)) = e^{Dt}Y(0). \tag{48}$$

That is, if the starting position is at Y(0) at time 0, at time t the process is expected to be at $F(t) = e^{Dt}Y(0)$.

Figure 11: Mplus input for autoregressive curves for the VAR model

```
variable:    NAMES ARE y1-y2 id t;
             lagged=y1-y2(1);
             cluster=id;
             CTIME=t;

DATA: file=1.dat;

ANALYSIS:   TYPE IS TWOLEVEL;
            estimator=bayes; proc=2;

MODEL:

  %WITHIN%
  y1^ y2^ on y1^1 y2^1;

  %BETWEEN%
  y1 with y2;

PLOT: TYPE = DRIFT (0, 5, 0.1);
```

Figure 12: Autoregressive curves for the VAR model



Taking a first derivative of the above equation we get that

$$F'(t) = DF(t). \tag{49}$$

Thus the drift matrix gives us a way to compute the rate of change (first derivative) for the expected process. At time 0

$$F'(0) = DF(0) = DY(0). \tag{50}$$

The first derivative also shows how the process changes in an infinitesimal amount of time t and that change is $tDY(0)$. In terms of a Taylor series expansion, using just the first derivative as an approximation

$$Y(t) \approx Y(0) + tDY(0). \tag{51}$$

We can expect that approximation to be very good for small values of t. The process is approximately linear in terms of infinitesimal amount of time and $DY(0)$ shows the direction of change for that process.

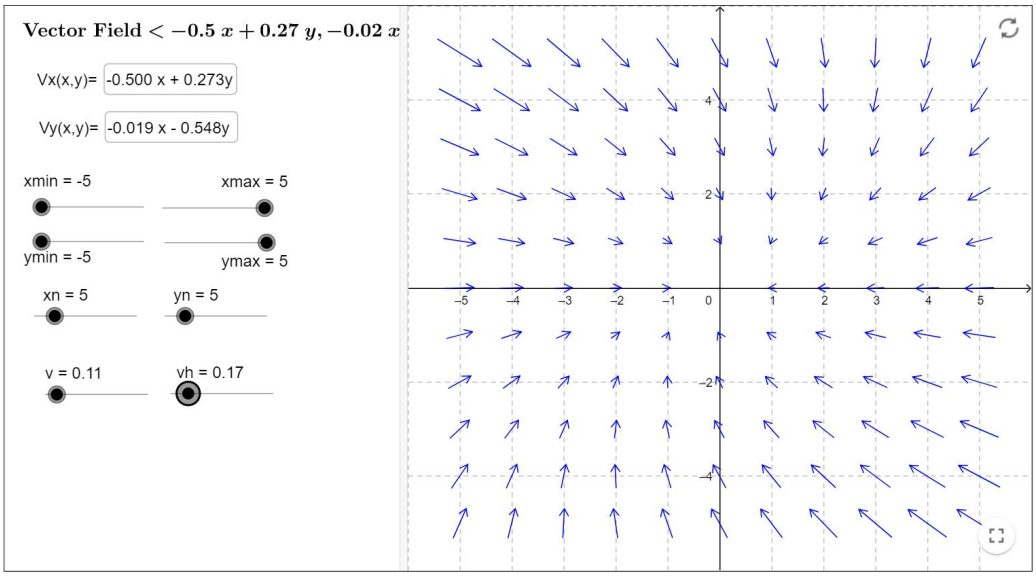
For two dimension processes, it is customary to plot arrows of the size of $DY(0)$ on a finite grid to illustrate the direction of change in the entire space, see Figure 2 in Ryan et al. (2018). Such a plot is called the vector field of the process and is currently not available in Mplus, but can be done for example in an online tool available at https://www.geogebra.org/m/QPE4PaDZ. Here we illustrate the vector field on the example used in the previous section, see Figure 13. This plot represents the vector field

$$< -0.500x + 0.273y, -0.019x - 0.548y > \tag{52}$$

where the coefficients are the drift parameter estimates. We should note here that in this entire section, the variables $Y(t)$ represent the simple autoregressive process defined in Section 2, i.e., we are focusing here on the autoregressive residuals across time. This is why all arrows converge to zero. The expected value of $F(t)$ for large t is always 0. If we are to include the between level part of the model or non-autoregressive within-model, then the arrows will converge to the expected/mean value of the process, which is often just the between part of the variables.

The curve $F(t)$ is the expected process for $Y(t)$ if you start at $Y(0)$. This curve can also be plotted in a vector field plot for a particular starting point Y(0) and it would show that the curve smoothly rides on along the vector field.

Figure 13: Vector field plot for simulated example



The above equations are sometimes also written as follows

$$Y(t + \Delta t) \approx Y(t) + \Delta t DY(t) \tag{53}$$

where t is any point in time and Δt represents an infinitesimal increment of time.

In the above differential equations, we used the derivative of the conditional expectation. However, underlying the continuous time modeling are actually stochastic differential equations that are utilized in Physics and Econometrics. In particular, the CT-RDSEM modeling is rooted in the Ornstein–Uhlenbeck (OU) stochastic process, Uhlenbeck and Ornstein (1930), which is defined by the following stochastic differential equation

$$dY(t) = DY(t)dt + \sigma dW_t, \tag{54}$$

where W_t is the Wiener process (Brownian motion), which is the integral, i.e., the totality of white noise. The above equation can be interpreted as: the change in $Y(t)$ consists of the drift part $DY(t)$ and white noise dW_t. The OU process conditional on the observation at minus infinity, i.e., unconditional, is a stationary process with a constant mean and variance. Furthermore, the Euler–Maruyama method, used for approximate numerical solutions to

stochastic differential equations, discretizes the OU process to a discrete-time AR(1) model. This is exactly how the RDSEM-Tinterval model approximates the CT-RDSEM model.

9 A real-data example

In this section we illustrate the CT-RDSEM model with data from a smoking cessation study (Shiffman et al., 1997, Muthén et al. 2024). Random prompts were issued by a handheld device on average five times a day for reporting on smoking urge as well as negative affect. In addition to the random prompts, several more event-oriented self reports were also made during temptations to smoke and brief smoking lapses. There are 235 individuals in the sample and the number of observations per person varies between 5 and 239, with an average number of observations per person near 100. On average 5 observations per day are recorded and the number of recorded days per person varies from one day to 4 weeks.

The random prompts are a minimum of 10 minutes apart, but the self reports are not separated in time from other observations. This creates about 5 to 10 occasions where a random prompt recording of no smoking urge is almost immediately followed by a self-report with maximum smoking urge. Such observations, as discussed in Section 5.5, become outliers when they are modeled in dynamic settings. The CT-RDSEM model assumes that observations taken at almost the exact same time would be very similar and thus the model can not account for such abrupt changes in outcome. If the data is analyzed in its original state, the smoking urge autocorrelation estimate is 0. This estimate however is driven only by those 10 observations which have outsized influence on the likelihood value. If those observations are removed from the data, the autocorrelation is not zero. One way to try to resolve the problem is to attempt to explain the abrupt change in the outcome with covariates. The most promising covariate available in the data is the binary indicator covariate which is 1 for self-reporting and 0 for random prompts. Self-reporting by definition is expected to be related to higher smoking urge. However, that covariate can explain only about 20% of the abrupt change and thus the problem is alleviated to some extent but not enough to prevent the outsized influence of the outliers. Clearly, the smoking urge variable is subject to random shocks that can not be accounted for without some additional latent variable modeling. In addition, it might be hypothesized

that a random prompt regarding smoking urge may itself become a trigger for the smoking urge. As a resolution to this issue and for the purpose of this illustration, we have chosen the following approach. A minimum of 10 minutes distance between the observations is set not just for the random prompts but also for the self-reported prompts. This amounts to altering the time stamp for 10-20 observations. It should be noted here that such an issue arises almost exclusively in the continuous time modeling framework. RDSEM and DSEM models utilizing TINTERVAL automatically space out observations with nearly identical timestamps. Also it should be noted here that this issue is not uncommon. About half of the real datasets we have explored, contained such observations and even observations that are with exactly identical timestamps. Typically, the number of such observations is less than 10 and these might be the result of software glitch or typos.

The two variables that we consider for this illustration are smoking urge and negative affect. The model we use is the CT-RDSEM VAR model discussed earlier but with covariates. Four within level covariates are available: arousal, two dummy covariates indicating that the observation is non-random and due to temptation or a lapse in smoking, and the current status of Abstinence/Lapsed. Three between level covariates are available: age, gender and the average number of cigarettes at baseline. We analyze the data with the CT-RDSEM model, and with the RDSEM-TINTERVAL model using three different settings for the discretization interval: 20 minutes, 30 minutes and 1 hour. The estimation time for the CT-RDSEM model is 10 minutes, for RDSEM-TINTERVAL(20 min) it is 56 minutes, for RDSEM-TINTERVAL(30 min) it is 9 minutes, and for for RDSEM-TINTERVAL(60 min) it is 1 minute. The RDSEM-TINTERVAL(20 min) analyzes a data set that is more than 10 times larger than the data set analyzed with CT-RDSEM due to the inserted missing observations.

The regression coefficients for the predictors are largely unaffected by the estimation method, all four methods yield nearly identical results and significance patterns. Figures 14 and 15 contain the regression parameter estimates for CT-RDSEM and RDSEM-TINTERVAL(20 min). The autoregressive parameter results are reported in Table 7. Here we compare the CT-RDSEM values with the three RDSEM-TINTERVAL estimations for the corresponding time interval. The diagonal autoregressive coefficient for smoking urge r_{11} is overestimated by RDSEM-TINTERVAL. Not accounting for the precise times of observations fails to reflect the more erratic nature of smoking urge. The underestimation is most pronounced for TINTERVAL of 60 minutes.

Table 7: Smoking urge, negative affect autoregressive comparison CT-RDSEM (RDSEM-TINTERVAL)

Minutes	r_{11}	r_{12}	r_{21}	r_{22}
20	.42(.53)	.12(.15)	-.01(.00)	.82(.83)
30	.27(.44)	.14(.16)	-.01(.00)	.75(.77)
60	.07(.36)	.14(.14)	-.01(.00)	.55(.65)

The CT-RDSEM values behave as expected across time and the cubic relationship that we expect (assuming upper-diagonal matrix given that r_{21} is approximately 0) when comparing r_{11} at 20 and 60 minutes holds: $.42^3 = .07$. On the other hand, the RDSEM-TINTERVAL values do not $.53^3 = .15 \neq .36$. Similar pattern can be observed for the negative affect diagonal autoregressive coefficient but to a much smaller extent. This perhaps is due to the negative affect being less erratic than smoking urge. Interestingly, the cross-lag effects are the most robust to the time-frame treatment. The two diagonal drift parameters are estimated at -0.862 and -0.195. Using formula (20) the cross-lagged effect from negative affect to smoking urge r_{12} is maximized at $t^* = 2.23$ which corresponds to a 45 minute interval where the value is .15. The cross-lagged effect dissipates slowly and for example at 3 hours the value is .05.

The Mplus input files for the CT-RDSEM and RDSEM-TINTERVAL(20 min) models are given in Figures 16 and 17. The only difference between the two is the specification of the time variable. The time variable in the data is in days, meaning that 1 represents 1 day. Dividing the time variable by 0.014, which represents 20 minutes, allows us to align the time frame between the CT-RDSEM run and RDSEM-TINTERVAL(20 min) so that 1 represents 20 minutes in both cases.

Figure 14: Smoking urge model regression parameters CT-RDSEM

	Estimate	Posterior S.D.	One-Tailed P-Value	95% C.I. Lower 2.5%	Upper 2.5%	Significance
Within Level						
URGE ON						
OBSTYPE1	0.610	0.037	0.000	0.537	0.679	*
OBSTYPE2	1.907	0.051	0.000	1.808	2.008	*
STATUS	-0.480	0.044	0.000	-0.569	-0.396	*
AROUSAL	0.090	0.016	0.000	0.058	0.121	*
NEGAFF ON						
OBSTYPE1	0.163	0.012	0.000	0.140	0.185	*
OBSTYPE2	0.302	0.016	0.000	0.271	0.332	*
STATUS	-0.012	0.022	0.281	-0.057	0.030	
AROUSAL	0.165	0.006	0.000	0.152	0.177	*
Between Level						
URGE ON						
AGE	0.024	0.014	0.045	-0.003	0.053	
GENDER	-0.481	0.298	0.052	-1.062	0.100	
AVECIGS	0.038	0.013	0.002	0.013	0.064	*
NEGAFF ON						
AGE	0.006	0.004	0.082	-0.002	0.014	
GENDER	-0.046	0.090	0.303	-0.227	0.130	
AVECIGS	0.008	0.004	0.023	0.000	0.016	*

Figure 15: Smoking urge model regression parameters RDSEM-TINTERVAL(20 min)

	Estimate	Posterior S.D.	One-Tailed P-Value	95% C.I. Lower 2.5%	Upper 2.5%	Significance
Within Level						
URGE ON						
OBSTYPE1	0.593	0.037	0.000	0.521	0.665	*
OBSTYPE2	1.890	0.049	0.000	1.795	1.992	*
STATUS	-0.469	0.045	0.000	-0.552	-0.375	*
AROUSAL	0.093	0.016	0.000	0.061	0.123	*
NEGAFF ON						
OBSTYPE1	0.153	0.011	0.000	0.131	0.175	*
OBSTYPE2	0.317	0.016	0.000	0.285	0.347	*
STATUS	-0.006	0.019	0.381	-0.040	0.032	
AROUSAL	0.165	0.006	0.000	0.152	0.176	*
Between Level						
URGE ON						
AGE	0.024	0.014	0.048	-0.004	0.052	
GENDER	-0.478	0.292	0.051	-1.052	0.096	
AVECIGS	0.038	0.013	0.003	0.012	0.063	*
NEGAFF ON						
AGE	0.006	0.004	0.081	-0.003	0.014	
GENDER	-0.045	0.089	0.305	-0.220	0.126	
AVECIGS	0.008	0.004	0.024	0.000	0.016	*

Figure 16: Smoking urge CT-RDSEM

```
Data: file = smok.dat;

Variable:

    names = subject timeqd6 ObsType1 ObsType2 Status
    AROUSAL URGE NEGAFF GENDER AGE AVECIGS;
    missing = all(999);
    cluster = subject;
    lagged = URGE-NEGAFF(1);
    ctime = timeqd6;
    within = ObsType1-AROUSAL;
    between = GENDER-AVECIGS;

define:
    timeqd6=timeqd6/0.014;

Analysis:
    type = twolevel;
    estimator = bayes;
    proc = 2;

Model:
    %Within%
    URGE^ NEGAFF^ on URGE^1 NEGAFF^1;
    URGE NEGAFF on ObsType1-AROUSAL;

    %Between%
    URGE NEGAFF on GENDER-AVECIGS;
    URGE with NEGAFF

Plot: type = plot3 drift(0.5,10,0.5);
```

Figure 17: Smoking urge RDSEM-TINTERVAL(20 min)

```
Data: file = smok.dat;

Variable:

    names = subject timeqd6 ObsType1 ObsType2 Status
    AROUSAL URGE NEGAFF GENDER AGE AVECIGS;
    missing = all(999);
    cluster = subject;
    lagged = URGE-NEGAFF(1);
    tinterval = timeqd6(0.014);
    within = ObsType1-AROUSAL;
    between = GENDER-AVECIGS;

Analysis:
    type = twolevel;
    estimator = bayes;
    proc = 2;

Model:
    %Within%
    URGE^ NEGAFF^ on URGE^1 NEGAFF^1;
    URGE NEGAFF on ObsType1-AROUSAL;

    %Between%
    URGE NEGAFF on GENDER-AVECIGS;
    URGE with NEGAFF
```

10 Appendix: Brief description of variations in the CT-RDSEM model estimation

The drift parameters D and the total residual variance parameters V in the CT-RDSEM model do not have an explicit conditional distribution for use in the MCMC estimation. The updating of these parameters is based on the Metropolis-Hastings algorithm. In some situations, the updating of these parameters is slow. The parameters mix poorly and the convergence is slow. An efficient mixing algorithm has two properties: the size of the jumps between iterations is relatively large and the autocorrelation is relatively small. In addition, the MH algorithm updates the parameters with a certain probability. That is with a certain probability the parameter changes in each iteration or it remains the same. When the parameter does not change, i.e., the proposed new value is rejected, the parameter remains the same and this increases the autocorrelation. An efficient MH algorithm aims for approximately 50% acceptance/rejection probability. This would make it approximately as half as efficient as an updating algorithm which has explicit conditional distribution. The most basic version of the MH algorithm uses a random walk jumping rule, that is, the proposed new value is obtained from the previous value by adding a random zero mean normally distributed jump variable. The size of the jump, i.e. the variance of the normally distributed jumps, is what controls the efficiency of the algorithm. The bigger the jumps the smaller the autocorrelation in the parameter. However, the bigger the jumps are the less likely it is that the proposed new value is accepted as being a suitable candidate of the desired posterior. In both of the extremes: very small jumps and very large jumps, the autocorrelation in the parameter will be very high and the mixing will be poor and convergence will be slower. When the jumps are very small the new values are almost identical to the old values and thus the autocorrelation is large. When the jumps are large, most of the proposed new values are rejected, the parameter does not change often and thus the autocorrelation is also very high. Within the burnin phase the size of the jumps is thus adjusted to stay away from the two extremes. The size of the jumps does not change after the burnin phase to preserve the propriety of the MCMC estimation and only iterations after the burnin phase are used to determine convergence and posterior distribution for the parameters. Unlike most Bayes estimation procedures in Mplus, the CT-RDSEM estimation for D and V needs more attention. The graphic utilities

can be used to monitor the quality of the mixing. Both the trace plots and the autocorrelation plots should be evaluated for these parameters to ensure that the mixing is good, i.e., simply relying on the PSR convergence criterion might not be enough. The trace plots should look at least to some extent like a proper caterpillar plots and the autocorrelation plots should die down to near zero within the lag-30 frame that Mplus plots. In addition, the Mplus tech8 output and back screen printing is expanded now to print the acceptance/rejection rates for the parameters (first new column) as well as the size of the jumps (second new column). If the acceptance rejection rates go down to zero, the MCMC is likely stuck to a point where it can't find a suitable candidate to move to and likely to lead to convergence issues. If the size of the jumps go down to zero, the trace plots will look like continuous curves instead of caterpillars and convergence also will not be achieved. In an attempt to provide a larger arsenal of utilities that can be used to address such issues we have implemented a few different estimating options and those are described below. There are four groups of parameters: D, V, random effect D and random effect V. The different mixing schemes use different approaches for the four groups.

10.1 Multivariate Random Walk MH

This is the default estimation in Mplus. The estimation is most straightforward and robust in a way that it can easily adjust acceptance/rejection rates to the desired level but the method can not guarantee that the jumps are sufficiently large to avoid poor mixing. The normal jumps are directly added to the drift parameters, while the jumps on the variance covariance are added after a re-parameterization: log transform for the variances and Fisher-z transform for the correlations. The four blocks are estimated separately but within-block the parameters are updated simultaneously. We refer to this scheme as MMH(Multivariate Metropolis Hastings).

10.2 Univariate Random Walk MH

This estimation is obtained with "algo=UniMH'. The scheme is similar to MMH except that each parameter is updated separately. This allows the size of the jumps to be parameter specific, however, the cost of this is that the parameters and the data likelihood must be computed for each parameter change rather than just the four multivariate blocks.

10.3 MH2

This estimation is obtained with "algo=MH2'. This estimation scheme features two new ideas. The first one is a new proposal distribution for the variance covariance parameters. If we assume that the autocorrelation of the variables is zero, the conditional distribution of the variance covariance is the inverse wishart distribution with degrees of freedom equal to the sample size. We use that as the proposal distribution and we vary the degrees of freedom (DF) of the distribution to make it wide or more narrow. Because the new sample points are independent of the previous parameter estimates, we have no way to ensure that the acceptance rejection rates of this scheme will remain within proper bounds. Both very large DF or very small DF will lead to 100% rejection and thus the burnin phase is used to adjust the proposal distribution. In some cases, however, apparently the inverse wishart distribution is a poor approximation to the correct conditional distribution and no matter what DF we use the rejection rates are very low. Examples where this happens are: small cluster sizes and situations where the data contains outliers. Because new parameter draws are conditionally independent from the current parameter set, this method usually has small autocorrelation in the parameters and so simply monitoring the acceptance / rejection rates is sufficient. If the acceptance/rejection rates however drops to near zero the method will automatically be converted back to the default MMH. This approach is referred to as the Hastings independence chain, see page 133 Carlin and Louis (2009). The quality of the approach is completely reliant on having a meaningful proposal distribution. In this particular case it appears that the inverse-wishart distribution is indeed such a distribution, however, it does not work in all cases.

The second idea featured in this estimation scheme is regarding the drift parameters. Here instead of completely random walk jumps we use correlated jumps. The first half of the burnin estimation is as in MMH. Then in the second half we use the previous iterations to estimate $\Sigma = Var(D)$ and instead of using independent jumps we use jumps proportional to Σ. This can be more efficient when the drift parameters are somewhat highly correlated which is often the case. The jumps in the second half of the burnin period zero mean normal jumps drawn from $s\Sigma$. We always start with $s = 2$ and then adjust s in the remaining part of the burnin to obtain a proper rejection acceptance rate. The Σ matrix is also updated in the second half of the burnin phase. This idea is described on page 135 in Carlin and Louis (2009).

With this scheme the V and random V parameters are updated using the inverse wishart, the drift parameters are updated using correlated random walk, while the random drift parameters use independent random walk as in MMH.

10.4 MH3

This estimation is obtained with "algo=MH3'. This method is the same as MH2 except that the variance covariance parameters are estimated as in MMH.

10.5 MH4

This estimation is obtained with "algo=MH4'. This method is based on the Langevian-Hastings (LH) algorithm described on page 137 in Carlin and Louis (2009). The method is also a special case of Hamiltonian Montecarlo (HMC) and amounts to a 1-step HMC. The method is simply a combination of random walk and steepest descent optimization algorithm where the jump consists of a random walk jump in a combination with a non-random jump in the direction of the gradient. Because the jumping rule is non-symmetric the MH acceptance rejection probability takes into account the non-symmetry. The approach proceeds as follows. In the first half of the burnin phase the MMH method is used and after that the LH method is used in the second half of the burnin phase. The size of the jumps are updated during the entire bunrin phase. The computation of the gradient makes the iterations substantially slower, particularly because the gradient can not be computed explicitly and is currently numerically approximated. With this scheme the four blocks are updated as follows. For D and V we use random walk followed by LH, while for random D and V we use only random walk. The four blocks are updated separately but within each block the parameters are updated together. In principle, this method is expected to converge faster than random walk alone but because the iterations take longer to compute it might end up being slower than random walk. Occasionally the estimation gets stuck and the acceptance rate becomes zero. In that case the estimation will continue with the default MMH.

10.6 MH5

This estimation is obtained with "algo=MH5'. This scheme is similar to MH4. The difference here is that D and V are updated simultaneously. This can be valuable when the posterior correlation between D and V is larger.

References

[1] Asparouhov, T., Hamaker, E. L., & Muthén, B. (2018). Dynamic structural equation modeling. Structural Equation Modeling: A Multidisciplinary Journal, 25 , 359-388.

[2] Asparouhov, T., & Muthén, B. (2006). Constructing covariates in multilevel regression. Mplus Web Notes No. 11. http:// www.statmodel.com

[3] Asparouhov, T., & Muthén, B., (2019). Latent variable centering of predictors and mediators in multilevel and timeseries models. Structural Equation Modeling: A Multidisciplinary Journal. 26, 119–42.

[4] Asparouhov, T., & Muthén, B. (2020). Comparison of models for the analysis of intensive longitudinal data. Structural Equation Modeling: A Multidisciplinary Journal. 27, 275–297.

[5] Asparouhov, T., & Muthén, B., (2023). Bayesian Analysis Using Mplus: Technical Implementation. Muthén & Muthén, Los Angeles, CA.

[6] Asparouhov, T. & Muthén, B. (2024). Covariance of Random Effects in Multilevel Modeling. Web Note No. 24.
https://www.statmodel.com/

[7] Bernstein, D. S. & So, W. (1993) Some explicit formulas for the matrix exponential. IEEE Trans. Autom. Control, vol. 38, no. 8, pp. 1228-1232, Aug. 1993.

[8] Boker, S. M. (2002). Consequences of continuity: The hunt for intrinsic properties within parameters of dynamics in psychological processes. Multivariate Behavioral Research, 37 (3), 405–422

[9] Boker, S. M., Neale, M., & Rausch, J. (2004). Latent differential equation modeling with multivariate multi-occasion indicators. In O. H. van Montfort K. & A. Satorra (Eds.), Recent developments on structural equation models (pp. 151–174). Dordrecht, the Netherlands: Kluwer Academic.

[10] Carlin BP, Louis TA (2009) Bayesian methods for data analysis, 3rd edn. Chapman and Hall/ CRC Press, Boca Raton, FL

[11] Chow, S., Ferrer, E., & Nesselroade, J. R. (2007). An unscented kalman filter approach to the estimation of nonlinear dynamical systems models. Multivariate Behavioral Research, 42 (2), 283–321

[12] Deboeck P.R. & Preacher K.J. (2016) No need to be discrete: a method for continuous time mediation analysis. Structural Equation Modeling: A Multidisciplinary Journal, 23 , 61-75.

[13] Dormann, C., & Griffin, M. A. (2015). Optimal time lags in panel studies. Psychological Methods, 20, 487-505.

[14] Driver, C. C., Oud, J. H. L., & Voelkle, M. C. (2017). Continuous time structural equation modeling with R package ctsem. Journal of Statistical Software, 77. 1–35

[15] Hecht, M. & Zitzmann S. (2021) Exploring the Unfolding of Dynamic Effects with Continuous-Time Models: Recommendations Concerning Statistical Power to Detect Peak Cross-Lagged Effects, Structural Equation Modeling: A Multidisciplinary Journal, 28:6, 894-902.

[16] Jongerling, J., Laurenceau, J. P., & Hamaker, E. (2015). A multilevel AR(1) model: Allowing for inter-individual differences in trait-scores, inertia, and innovation variance. Multivariate Behavioral Research, 50, 334–349.

[17] Ludke O, Marsh HW, Robitzsch A, Trautwein U, Asparouhov T, Muthén B (2008). The multilevel latent covariate model:a new, more reliable approach to group-level effects in contextual studies. Psychol. Methods. 203–229.

[18] Mouçouf M. & Said Z. (2020) Some Explicit Formulas for Matrix Exponential, Matrix Logarithm, the n-th Power of Matrices and their Drazin Inverses. arXiv preprint arXiv:2010.12551.

[19] Muthén, B., Asparouhov, T., Shiffman S. (2024). Dynamic Structural Equation Modeling with Floor Effects. https://www.statmodel.com/download/DSEMFloorEffects.pdf

[20] Muthén, L. K., & Muthén, B. O. (1998-2017). Mplus user's guide (8th ed.). Muthén & Muthén.

[21] Oravecz, Z., Tuerlinckx, F., & Vandekerckhove, J. (2011). A hierarchical latent stochastic differential equation model for affective dynamics. Psychological Methods, 16, 468-490.

[22] Shiffman, Engberg et al (1997). A day at a time: predicting smoking lapse from daily urge. Journal of Abnormal Psychology, 106, 104-116.

[23] Sidje, R. B. (1998) Expokit: a software package for computing matrix exponentials. ACM Trans Math Softw. 24, 130–156.

[24] Ryan O., Kuiper R.M., Hamaker E.L. (2018) A continuous time approach to intensive longitudinal data: what, why and how?, Chapter 2 in Continuous time modeling in the behavioral and related sciences, K.V. Montfort, J.H.L. Oud, M.C. Voelkle (Eds.), Springer, New York, NY, pp. 27-54.

[25] Uhlenbeck, G. E.; Ornstein, L. S. (1930). On the theory of Brownian Motion. Phys. Rev. 36 (5): 823–841.