Practical Aspects of Dynamic Structural Equation Models

Tihomir Asparouhov and Bengt Muthén

Version 4

December 2, 2024

1 Introduction

In this note we address some practical aspects of dynamic structural equation models. In particular, we focus on what can be done when the model estimation fails, i.e., it doesn't converge. Here we provide a check-list of ideas that may be helpful in resolving the convergence problems and may help with further understanding the analysis. We also address some other DSEM specific issues that are often considered in practical applications such as including overnight auto-correlation issues, three-level DSEM, and significance of standardized random slopes.

2 The source of the convergence problem

Generally, there are two types of common error messages produced by Mplus in failed DSEM analysis. The first one is that the convergence criterion was not satisfied within the maximum number of iterations. By default the maximum number of iterations is 50000. It is possible to increase that number with the BITER option, however, that generally would be the last resort attempt to resolve the issue. When this problem occurs, the estimation is mixing very slowly. The proper way to fix the problem is to identify the causes for the slow mixing and to modify the model or data accordingly. The second common error message is that the between level variance covariance matrix is singular or that it could not be updated due to a singular posterior distribution. This situation could be more complex to diagnose but generally the issue is most likely related to the random effects in the model.

It many situations, the source of the problem is not easily identifiable and there are several causes combined leading to convergence problems. Below we list some common scenarios that occur in real data applications.

2.1 Unidentified or poorly identified model

If the model is unidentified or is poorly identified, the outcome of the estimation very likely will be non-convergence within the maximum number of iterations. For example, latent variables measured by one or two indicators are often poorly identified and may be the source of the problem. The solution in this case is to use a better identified model by introducing model parameter restrictions, such as loadings fixed to 1 in the case of the

two-indicator factor model.

2.2 Random effects with small variances

If the variance of the random effect is small, the variance of the conditional distribution used for updating the random effect will likely be small as well. That means that the random effect might not change much across the MCMC iterations. This in turn will cause the parameters related to that random effect, such as the mean and the regression parameters of that random effect on other variables, to be mixing very slowly and possibly result in non-convergence. Random effects with small variances can be converted to non-random effects. In some situations, a small variance random effect can remain in the model if we simply do not correlate the random effect with other random effects. By default, Mplus will set all random effects on the between level to be uncorrelated with each other. When the random effects are uncorrelated, the model is more likely to converge. In addition, the model with uncorrelated random effects can be used to identify those small variance random effects that are not needed in the model, before proceeding with a more advanced model that correlates all random effects, see Asparouhov and Muthén (2024a). Small variance random effects tend to cause convergence problems when cluster sizes are larger and when the random effects are correlated with other variables.

With the Bayesian estimation, the variance of the random effects will always be significant in terms of having a positive credibility interval. This is somewhat of an artifact, however. The posterior distribution of a variance parameter can take only positive values and therefore the credibility interval will always imply that the parameter is significant. This is not how the credibility interval should be interpreted. In the case of variance parameters the credibility interval can not be used to infer significance. Here we provide a very simple suggestion for how to decide if the variance of a random effect is significant. Bayesian estimation and ML estimation are asymptotically equivalent. Asymptotically here means a large number of clusters. In most applications however, the number of clusters is not very large and probably when the number of clusters is smaller than 1000 this asymptotic equivalence is fairly approximate. Nevertheless, using an ML based method to test the significance of the variance of the random effect is a reasonable approach. Unfortunately, however, testing for significant variance of a random effect models is a challenging problem even with the ML estimation,

see Self and Liang (1987), because the parameter is at the border of the admissible space where both asymptotic variance covariance inference and likelihood ratio tests break down. We have however a very simple rule of thumb that can be utilized. Consider the Z score for the variance parameter: Z=estimate/standard error. If $Z > 3$, we can make some implications. First, the parameter can be considered to be away from the border. Second, the asymptotic variance covariance would be reasonable. Third, because $Z > 3$, the variance is significant. This same approach can be used with Bayes. We generally recommend variance components with $Z < 3$ to be of questionable significance and if such random effects are also causing convergence problems, the decision to remove the random effect from the model appears to be reasonable.

It should be noted here that there is a difference between small residual variance and small total variance of the random effect. If a random effect has a small residual variance but is regressed on several covariates with large and significant coefficients, clearly the total variance of the random effect would not be small and would also be significant. In such a situation, converting the random effect to a non-random effect may be less desirable. We would want to keep the significant regression coefficients in the model. That can generally be achieved by introducing interactions, however, if one of the two variables in the interaction is latent, the modeling framework will not be allowed in Mplus. In such a situation a reasonable approach is to keep the random effect in the model but remove any residual correlations that the small residual variance is involved in.

We illustrate the small variances convergence problems with a simulation study. Figure 1 contains the Mplus montecarlo setup for a bivariate two-level DSEM model. The data generating DSEM model has random intercepts and non-random auto-regressive (AR) coefficients, which include the cross-lagged coefficients. The estimated DSEM model has random intercepts and random AR coefficients. The variances of the AR random effects are all zero because they are not random in the data generating model. We can expect that the estimated values will also be near zero and therefore the simulation study illustrates small variance random effects. In the estimated between level model in Figure 1, all random effects are correlated. In this study, 3 out of 5 replications did not converge. If we replace the fully correlated between level model with the model where only the random intercepts are correlated and the small variance AR random effects are uncorrelated among each other and the random intercepts, the estimation converges in all 5 replications. Such a

result shows that small variance random effects cause problems mainly when they are correlated with other variables.

Small variance random effects can also cause an estimation problem when they are used as predictors for other variables. Because the variance of the effect is small, the random effect can be viewed approximately as a constant. Therefore the regression coefficient that multiplies that random effect will be poorly identified and with a large posterior distribution. Usually such parameters are slow to estimate and would require many iterations.

Figure 1: Simulation study illustrating small variances convergence problems

```
MONTECARLO:  NAMES ARE y z;
NOBS = 720;  NREP = 5;
NCSIZES = 2; CSIZES = 30(20) 20(6);
lagged = y(1) z(1);
missing=all;

model missing: [y*0 z*0];

ANALYSIS:   TYPE = TWOLEVEL RANDOM;
estimator=bayes;  proc=2; biter=(30000);

MODEL POPULATION:
  %WITHIN%
  y on y&1*0.7;
  z on z&1*0.7;
  z on y&1*0.2;
  y on z&1*0.2;
  y*0.5 z*0.5;
  y with z*-0.25;

  %BETWEEN%
  y*1; z*1;
  y with z*0.8;

MODEL:
  %WITHIN%
  s1 | y on y&1;
  s2 | z on z&1;
  s3 | z on y&1;
  s4 | y on z&1;

  %BETWEEN%
  y z s1 s2 s3 s4 with y z s1 s2 s3 s4;
```

2.3 Non-stationarity

A model is stationary when the eigenvalues of the AR (auto-regressive) matrix are all less than 1. In the univariate case this translates to the auto-regressive coefficient being less than 1. In the multivariate case, it is not easy to determine if the AR matrix has all eigenvalues less than 1 simply because the eigenvalues are not easy to compute by hand. In the bivariate case, however, we can provide a little guidance to quickly determine if the AR matrix is stationary. Suppose that the AR matrix is

$$R = \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix}.$$

The auto-regressive coefficients are r_{11} and r_{22} and the cross-lagged coefficients are r_{12} and r_{21}. In the most common scenario, where the AR coefficients r_{11} and r_{22} are positive and less than 1, and the cross-lagged coefficients are smaller than the auto-regressive coefficients, $r_{12} < r_{11}$ and $r_{21} < r_{22}$, we can simply check that the sum of the coefficients in each equation/row of the AR matrix are less than 1. That is, we can show that if $r_{11} + r_{12} < 1$ and $r_{22} + r_{21} < 1$, the model is stationary. The eigenvalues x_1 and x_2 satisfy

$$x_1 + x_2 = r_{11} + r_{22}$$

$$x_1 x_2 = r_{11} r_{22} - r_{12} r_{21}$$

Under the above restrictions of the autoregressive parameters, we can see that $r_{12} r_{21} < (1 - r_{11})(1 - r_{22})$. Because of the above equations, the last inequality is equivalent to $(1 - x_1)(1 - x_2) > 0$, which in turn leads to the conclusion that both eigenvalues must be between 0 and 1 and that the model is stationary.

In principle, stationarity is not required in DSEM. However, when stationarity is not satisfied in the entirety of the posterior distribution of the auto-regressive matrix, convergence problems can occur. In the univariate case, the MCMC updating involves dividing by $1-r$. If r is close to 1, numerically this is divided by zero and the computation will be contaminated with a large numerical error. During the MCMC iterations, the entire posterior distribution must be visited. If the posterior distribution is wide enough to cross over into non-stationarity, given a sufficient number of MCMC iterations, the posterior draw of r will eventually be close enough to 1 and destabilize the computation. This exact thing can happen in the multivariate case. The

matrix I minus the AR matrix would become singular and the posterior distribution of the intercept would be undefined. Furthermore, when the AR matrix is cluster specific, the AR matrix must remain stationary in every cluster in the entire posterior distribution, i.e., in every MCMC draw. When a particular cluster has very limited information, due to a poorly identified model or a limited number of observed values, the posterior distribution of the AR matrix will be quite wide and quite likely to lead to non-stationary values. This in turn may cause convergence problems.

The one key feature that appears to consistently separate montecarlo studies and real data applications is the size of the variances of the random auto-regressive coefficients. In montecarlo studies, the variances of the random AR effects are usually set between 0.01 and 0.1. Higher variances typically will lead to non-stationarity, at least in some of the clusters. In real data applications, it is fairly common, however, to see the variances of the random AR effects above 0.1. Typically, this would imply that the range of the AR coefficient across clusters is of length at least 0.6 and that most likely, the point estimates in some of the clusters are not stationary. Note that this is an observation regarding just the point estimates. The range of the entire posterior distribution would be even larger. In such situations, the conclusion is that it is a matter of getting through enough MCMC iterations and chains to run into a set of parameters that will destabilize the computation through dividing by a near-zero number. To summarize, variances of random AR effects above 0.1 should be inspected carefully in each cluster.

We do not view the issues discussed above as stationarity problems. In these situations, we would not recommend modeling trends or striving for stationary models. We view the problem instead as a limited information problem. Weak information leading to wide posterior distributions for a few of the clusters that can cause computational problems for the entire estimation.

We illustrate the non-stationarity convergence problems with a simulation study. Figure 2 contains the Mplus montecarlo setup for a bivariate two-level DSEM model with random intercept and random AR matrix. With the average random effect parameters, the model is stationary but close to the stationarity borderline. In this study, 5 out of 10 replications did not converge. For the 5 iterations that converged, the average random intercept variance estimates are very large and don't match the true values at all. The means of the diagonal AR random effects in Figure 2 are set at 0.7. If we change these means to 0.4 in the data generation model, the model is moved

away from the stationarity border. In that case, the estimation converges in 9 out of the 10 replications and the parameter estimates are close to the true values.

Figure 2: Simulation study illustrating non-stationarity convergence problems

```
MONTECARLO:  NAMES ARE y z;
NOBS = 1000; NREP = 10;
NCSIZES = 1; CSIZES = 100(10);
lagged = y(1) z(1);
missing=all;

model missing: [y*0 z*0];

ANALYSIS:   TYPE = TWOLEVEL RANDOM;
estimator=bayes; proc=4; biter=(30000); chain=4;

MODEL POPULATION:
  %WITHIN%
  s1 | y on y&1;
  s2 | z on z&1;
  s3 | z on y&1;
  s4 | y on z&1;
  y*0.5 z*0.5; y with z*0.25;

  %BETWEEN%
  y*1; z*1; y with z*0.8;
  s1-s4*0.01; [s1-s2*0.7 s3-s4*0.25];

MODEL:
  %WITHIN%
  s1 | y on y&1;
  s2 | z on z&1;
  s3 | z on y&1;
  s4 | y on z&1;
  y*0.5 z*0.5; y with z*0.25;
```



```
  %BETWEEN%
  y*1; z*1; y with z*0.8;
  s1-s4*0.01; [s1-s2*0.7 s3-s4*0.25];
```

2.4 Large number of random effects compared to the size of the clusters

Large number of random effects compared to the size of the clusters typically will produce a wide posterior distribution. Such large posterior distributions may lead to a non-stationary AR matrix that can cause convergence problems. The most reasonable approach in this case is to reduce the number of random effects. Ideally, the number of random effects, which includes all random intercepts and slopes, should be smaller than the size of the clusters. If the auto-regressive values are small, however, it is possible to estimate models with a larger number of random effects.

We can further illustrate this point with a simple regression model where a single dependent variable Y is regressed on Q covariates $X_1, ..., X_Q$. It is well known that if we try to estimate this model with Q observations or less using the ML estimator, the model is unidentified and the estimation will fail. There are an infinite number of parameters that fit the model equally well. The Fisher information matrix will be singular and the standard errors will be infinite. If we now transfer this situation to multilevel modeling in the Bayes settings, where all the regression slopes are now cluster specific, the same limitations will apply. If the clusters are of size Q or smaller, the estimation of the random effects within each cluster will yield infinitely large posterior distributions for all of the random effects, which subsequently will lead to unidentified model and non-convergence.

2.5 Large amount of missing data

Large amounts of missing data combined with small amounts of observed data can also lead to large posterior distributions and non-stationary AR matrices. In this case as well, the most reasonable solution is to convert some of the cluster-specific random effects to cluster-invariant non-random effects. If only a few clusters have an excessive amount of missing data, such clusters can be removed from the sample. These clusters contain very small information and removing them is unlikely to affect the estimation in a meaningful way. Simulation studies show that removing clusters with very limited information may increase MSE for some parameters but will also decrease MSE for other parameters.

2.6 Clusters that have very few observed values

Clusters that have very few observed values, i.e., short time-series data, will naturally yield wide posterior distribution for the random effects. This in turn may yield non-stationary convergence problems. The preferred solution in this case is to remove small clusters from the sample. As in the previous case, the overall performance of the model estimation will not change dramatically but it can resolve convergence problems.

We illustrate the small clusters convergence problems with a simulation study. Figure 3 contains the Mplus montecarlo setup for a bivariate two-level DSEM model with random intercept, random AR matrix, and random variance covariance matrix. 20 small clusters are included in this montecarlo study. In this study, 3 out of 5 replications did not converge. If we remove the small clusters from the simulation study, the estimation converges in all replications.

Figure 3: Simulation study illustrating small clusters convergence problems

```
MONTECARLO:  NAMES ARE y z;
NOBS = 1040; NREP = 5;
NCSIZES = 2; CSIZES = 50(20) 20(2);
lagged = y(1) z(1);
missing=all;

model missing: [y*0 z*0];

ANALYSIS:   TYPE = TWOLEVEL RANDOM;
estimator=bayes; proc=4; chain=4; biter=(30000);

MODEL POPULATION:
  %WITHIN%
  s1 | y on y&1;
  s2 | z on z&1;
  s3 | z on y&1;
  s4 | y on z&1;
  v1 | y; v2 | z;
  f by y@1 z@1;
  v3 | f;

  %BETWEEN%
  y*1; z*1; y with z*0.8;
  s1-s4*0.005; v1-v3*0.005;
  [v1-v3*-1]; [s1-s2*0.4 s3-s4*0.2];

MODEL:
  %WITHIN%
  s1 | y on y&1;
  s2 | z on z&1;
  s3 | z on y&1;
  s4 | y on z&1;
  v1 | y; v2 | z;
  f by y@1 z@1;
  v3 | f;

  %BETWEEN%
  y z s1-s4 v1-v3 with y z s1-s4 v1-v3;
  [v1-v3*-1]; [s1-s2*0.4 s3-s4*0.2];
```

2.7 Random residual covariance with varying signs

Hamaker et al. (2018) illustrates how random covariance can be estimated in Mplus by using the random variance for a common factor. This approach, however, is mostly applicable when the random covariances have the same sign across the clusters. If the covariances are positive in some clusters and negative in other clusters, the construction may fail. There is an easy way now to resolve this problem because subsequent Mplus development includes also direct random covariance modeling, see Asparouhov and Muthén (2023), Section 9.1.

We illustrate the issue with a simulation study. Figure 4 contains the Mplus montecarlo setup for a two-level model with random covariance using Hamaker et al. (2018) estimation method. In the data generation model, the random covariance is generated as the difference between the random variances of the two factors f_1 and f_2. The random covariance is equally likely to be positive or negative. The estimated model, based on Hamaker et al. (2018), assumes that the random covariance is positive. In this study, 5 out of 10 replications did not converge. Figure 5 contains the setup for directly estimating random covariance. With this setup all replications converge.

Figure 4: Simulation study illustrating the random residual covariance with varying signs convergence problems using Hamaker et al. (2018) setup

```
MONTECARLO:  NAMES ARE y z;
             NOBS = 1000;
             NREP = 10;
             NCSIZES = 1;
             CSIZES = 50(20);

ANALYSIS:   TYPE = TWOLEVEL RANDOM;
                estimator=bayes;
            proc=2;
            biter=(30000);

MODEL POPULATION:

  %WITHIN%
  y*1 z*1;
  f1 by y@1 z@1;
  f2 by y@1 z@-1;
  v1 | f1; v2 | f2;

  %BETWEEN%
  y*1; z*1;
  y with z*0.8;
  v1*0.1 v2*0.1;

MODEL:

  %WITHIN%
  v1 | y; v2 | z;
  f by y@1 z@1;
  v3 | f;

  %BETWEEN%
```

Figure 5: Simulation study illustrating the random residual covariance with varying signs using direct setup for random covariance

```
MONTECARLO: NAMES ARE y z;
 NOBS = 1000;
 NREP = 10;
 NCSIZES = 1;
 CSIZES = 50(20);

ANALYSIS: TYPE = TWOLEVEL RANDOM;
 estimator=bayes;
 proc=2;
 biter=(30000);

MODEL POPULATION:

 %WITHIN%
 y*1 z*1;
 f1 by y@1 z@1;
 f2 by y@1 z@-1;
 v1 | f1; v2 | f2;

 %BETWEEN%
 y*1; z*1;
 y with z*0.8;
 v1*0.1 v2*0.1;

MODEL:

 %WITHIN%
 v1 | y; v2 | z;
  c | y with z;

 %BETWEEN%
```

2.8 Exploding parameter estimates

In some situations, convergence failure results in final estimates that show exploding parameter values, that is, very large parameter values. This can occur when the model is unidentified or poorly identified. The exploding parameters could be analyzed to identify the weakness of the model. Exploding parameters can also occur when there is a singular variance covariance matrix. In that case, the instability can spread in the entire model and analyzing the exploding parameters is unlikely to provide useful information.

2.9 Variance fixed to zero or to a small value

In some modeling situations it is necessary to fix a residual variance to zero or a small value. This may cause very slow mixing in the Bayes estimation. The problem can be resolved by increasing the small value the variance is fixed to.

2.10 The random walk option

The Mplus default analysis options usually yield the most efficient model estimation. If estimation options are specified that are different from the Mplus defaults, these options might be the source of the problem. One example is the option ALGO=GIBBS(RW). This option allows Mplus to estimate non-block diagonal variance covariance matrices, however, it can produce very slow mixing estimation. We recommenced using models with block diagonal variance covariance matrices and removing that option. The random walk option works well only when the size of the variance covariance matrix is small.

Consider the bivariate DSEM simulation study given in Figure 6. In this simulation study, the intercepts, the auto-regressive parameters, and the residual variance covariance matrices are all random effects. A between level covariate X is also included in the model. The covariate is used to predict the dependent variable Y, using the Y on X statement. The estimated DSEM model also includes all covariances between the random effects as well as all the covariances between the random effects and the covariate X, with the exception of the covariance between Y and X which would be unidentified given the regression model between the two variables. Because the estimated variance covariance matrix on the between level includes all

the parameters except one, it is not a block diagonal matrix and the option ALGO=GIBBS(RW) must be specified. Figure 7 shows the tech1 output for this model and illustrates why the variance covariance matrix is not block-diagonal. The covariance between X and Y is fixed to 0 and this violates the block-diagonal structure. A variance covariance matrix is block-diagonal if it satisfies the following property: if A is correlated with B and C then B and C must also be correlated. In this model, this property is not satisfied for the variables X, Y and Z.

In this simulation study given in Figure 6, all 5 replications fail to converge. The solution to this problem is to replace the covariances **s1-s4 v1-v3 z with x;** with the regressions **s1-s4 v1-v3 z on x;**. Such a model has a block-diagonal variance covariance matrix and the option ALGO=GIBBS(RW) can be removed. All 5 replications converge with this modification. The two DSEM models are equivalent but only the second model can be estimated efficiently.

Figure 6: Simulation study illustrating the random walk/GIBBS(RW) convergence problems

```
MONTECARLO:  NAMES ARE y z x;
NOBS = 1000; NREP = 5;
NCSIZES = 1; CSIZES = 50(20);
lagged = y(1) z(1);
between=x;

ANALYSIS:   TYPE = TWOLEVEL random;
estimator=bayes; proc=2; ALGORITHM=GIBBS(RW); biter=5000;

MODEL POPULATION:
  %WITHIN%
  s1 | y on y&1;
  s2 | z on z&1;
  s3 | z on y&1;
  s4 | y on z&1;
  f by y@1 z@1;
  v1 | y; v2 | z; v3 | f;

  %BETWEEN%
  y*1 z*1 x*1 s1-s4*0.01 v1-v3*0.01;
  y on x*0.4; z with y*0.3; z with x*0.2;
  [v1-v3*-1 s1-s2*0.4 s3-s4*0.2];

MODEL:

  %WITHIN%
  s1 | y on y&1;
  s2 | z on z&1;
  s3 | z on y&1;
  s4 | y on z&1;
  f by y@1 z@1;
  v1 | y; v2 | z; v3 | f;

  %BETWEEN%
  y on x;
  s1-s4 v1-v3 y z with s1-s4 v1-v3 y z;
  s1-s4 v1-v3 z with x;
```

Figure 7: Tech1 with non-block diagonal variance covariance

PSI

	V2	V3	X	Y	Z
V2	32				
V3	38	39			
X	45	46	47		
Y	53	54	0	55	
Z	61	62	63	64	65

2.11 Excessive number of iterations

The posterior distribution of the random effects and the model parameters is generally concentrated around their point estimates. Theoretically, however, the posterior distributions encompass all values, i.e., from minus infinity to plus infinity. For most of these values, the density function is near zero and they are extremely unlikely to occur. If we run an extreme amount of iterations, however, the thin tails of the posterior distributions will also be populated to some extent. Some of these unlikely values may result in convergence problems. For example, a variance covariance matrix in Mplus is considered singular when the condition number of the matrix is less than 10^{-10}. With a very small probability that can occur even for a perfectly good model. If we run an excessive amount of iterations, it is possible to reach these unlikely points in the posterior distribution which will cause the computation to stop. We do not recognize this as a problem with the model or the estimation but with the setting requiring the excessive number of iterations. Instead of using an excessive number of iterations, it is more appropriate to use the Mplus default convergence criterion and the automated determination for the number of MCMC iterations. Before accepting the model as final, we also recommend doubling the number of iterations. If the PSR convergence criterion continues to be satisfied in all of the additional iterations, we conclude that the model has converged and the number of iterations is sufficient to populate the main part of the posterior distribution.

Generally, when the number of iterations is above 50000, it could be considered excessive. Such number of iterations should be used only if it is necessary to establish convergence, i.e., if the PSR criterion continues to exceed 1.1. Also, an excessive number of iterations can occur via the use of the THIN option in Mplus. The THIN command simply hides the intermittent draws, but can equally well lead to the perils of the extreme tails of the posterior distribution.

2.12 Factors with one or two indicators

Factors with one or two indicators are generally difficult to identify and may cause convergence problems even without any auto-regressive structures. In principle, such models can be estimated in DSEM, however, such estimation often requires very rich data sets that are rare in real applications. The solution here is to limit the flexibility of the model. For example, a two-

indicator factor model would be well identified if both loadings are fixed to 1.

2.13 Lack of variation in the data

If variables do not change within a cluster it is difficult to see how one can construct a meaningful time-series model for such variables. Mplus will list all variables that do not change within clusters as warning messages. The clusters can be removed from the analysis using the VARIABLE command option USEOBSERVATIONS = Y EQ _CLUSVAR.

In principle, information from the entire population can be used by the estimation to make inference for a cluster with no variation/information, but such an approach is limited in scope. It depends on how many of these clusters are in the population. The more non-informative data occurs in the population the more likely it is that the estimation will fail. Holding auto-regressive parameters invariant across clusters, i.e., using non-random auto-regressive parameters instead of random, can be used to resolve such convergence problems. Alternatively, removing clusters with such a low level of information from the analysis all together can improve the model estimation. Because the amount of information contained in such clusters is very small, removing these data points from the population is unlikely to affect the overall analysis dramatically.

2.14 Categorical variables that are predominately constant

Time-series model estimation is informed by how variables change across time. If the variables do not change much across time, it will be very difficult to build time-series models. If a binary variable remains mostly zero across time, it will be difficult to build a well-identified subject-specific time-series model, unless the time series data is very long. For categorical data that is predominately constant, we recommend using population level auto-regressive parameters, i.e., non-random auto-regressive parameters.

2.15 Using priors to resolve convergence problems

In some situations, convergence problems can be resolved by specifying weakly informative or proper priors instead of using the Mplus default uninforma-

tive and improper priors. Priors can help the estimation when the number of clusters is small. The most common scenario when priors can resolve convergence problems is when the variance covariance matrix on the between level is singular. The default prior for a variance covariance matrix in Mplus is $IW(0, -p-1)$ where p is the size of the variance covariance matrix. One alternative prior specification is $IW(I, p+1)$ where I is the identity matrix. The advantage of this prior is that it is a proper prior and it provides an uninformative/uniform marginal prior for the correlation parameters, see Asparouhov and Muthén (2010) and Asparouhov and Muthén (2021). Using such a proper prior for the between level variance covariance matrix can resolve convergence problems in some situations. Consider the bivariate DSEM example given in Figure 8. Using the Mplus default priors, with the model prior statement commented out, all 5 replications fail to converge. Using the prior $IW(I, p+1)$, with the model prior statement, only 1 of the replications fails to converge.

2.16 Scales of variables

In some situations convergence problems can be caused by the scale of the variables. Variables with variances over 1000 or under 0.001 can generally cause numerical problems. It is recommended to have all variables in any model on somewhat of a standardized metric. One can generally accomplish this using the DEFINE: command in Mplus and by dividing or multiplying by a power of 10 or by subtracting a number or both. Even if re-scaling of the variables doesn't resolve the convergence problem, it will likely be very helpful with diagnosing the remaining problem.

2.17 Using exogenous variables to keep the model parsimonious

Convergence problems can arise simply because the model has too many parameters. Generally, we want to keep any model as parsimonious as possible. This would reduce the likelihood that a convergence problem will occur and it will preserve the power of the model to detect significant connections between the variables.

The simplest way to reduce the number of parameters in a model is to treat exogenous variables as such, i.e., they should only be regressed on.

Figure 8: Using priors to resolve convergence problems

```
MONTECARLO:  NAMES ARE y z;
NOBS = 800; NREP = 5;
NCSIZES = 2; CSIZES = 30(20) 20(10);
lagged = y(1) z(1);
missing=all;

model missing: [y*0 z*0];

ANALYSIS:   TYPE = TWOLEVEL RANDOM;
estimator=bayes;  proc=2; biter=(30000);

MODEL POPULATION:

  %WITHIN%
  y on y&1*0.5;
  z on z&1*0.5;
  z on y&1*0.2;
  y on z&1*0.2;
  y*0.5 z*0.5; y with z*-0.25;

  %BETWEEN%
  y*1; z*1; y with z*0.6;

MODEL:

  %WITHIN%
  y on y&1*0.5;
  z on z&1*0.5;
  z on y&1*0.2;
  y on z&1*0.2;
  y*0.5 z*0.5; y with z*-0.25;

  %BETWEEN%
  y*1 (v1); z*1 (v2); y with z*0.6 (c);

!model prior:
!v1~IW(1,3);
!v2~IW(1,3);
!c~IW(0,3);
```

Model statements such as "exogenous WITH exogenous" are not needed and should be removed. Model statements such as "endogenous WITH exogenous" should be converted to "endogenous ON exogenous".

2.18 Multiple wave sampling

Some special issues arise in DSEM when data is collected in multiple waves. Consider as an example the case where data is collected during the first week (wave) of the month for several months. The waves are separated by 23 days where no observations are taken. The DSEM analysis will insert missing values in those 23 days. If the autocorrelation from one day to the next is r, then the autocorrelation across 23 days is r^{23} which in most situations will be practically zero. Thus, 23 missing rows are added to the data, to account for an autocorrelation that does not exist. Autocorrelation between consecutive waves that are long stretches of time apart is unrealistic. If the waves are further apart, a potentially huge amount of missing data will need to be added in the DSEM estimation. To make the estimation more efficient, it is recommended that only a limited number of missing data rows are added between waves. This would usually amount to altering the time variable.

If we have a preliminary or an approximate estimate of the autocorrelation r, we recommend that no more than $L = \log(0.01)/\log(r)$ missing data rows are inserted between waves. This will ensure that the amount of missing data is long enough to practically reduce the autocorrelation between the waves to 0 (< 0.01). If the preliminary estimate for r is 0.5, then $L \approx 7$. Inserting 7 rows of missing data will yield the same result as inserting 23 rows of missing data. Reducing the number of missing rows between the waves to L is expected to yield a much faster estimation.

Multiple wave sampling may have additional complications. The subject specific component of the variable may change over the waves. Wave and subject specific random effects may be needed and those may be autocorrelated. Some ideas for addressing these issues are given in Section 4 below, such as 3-level modeling, etc. A different approach to multi-wave data, based on multivariate modeling, is described in Hamaker et al (2021).

3 Building up the model

A two-level DSEM model should always be constructed in stages. Start with the simplest model to the most advanced. Using this strategy will reduce the likelihood of an unidentifiable convergence problem. We recommend using the following estimation sequence.

1. **Two-level model**
   This model can simply be the unrestricted/unstructured two-level model. In this model all variables are correlated on the within and the between levels, but there are no lagged variables involved in the model at all. Alternatively, the model can be the structured two-level model that has the structure of the DSEM model without the auto-regressive parameters.

2. **Two-level DSEM model with random intercepts only**
   All other parameters should be non-random. In Asparouhov et al. (2018) and Asparouhov and Muthén (2020) it is shown that the RDSEM model is the more natural extension(transition) from the two-level model than the DSEM model. Note that the RDSEM and the DSEM models differ only if there are covariates in the model. Even if the DSEM model is the ultimate goal, it might be useful at this stage of the analysis to explore the RDSEM model as well. This is easy to do in Mplus and involves replacing the & symbol (used for lagged regressions) with the ^ symbol.

3. **Two-level DSEM model with uncorrelated random intercepts and slopes**
   Random slopes should be added one at a time. Random slopes should be uncorrelated and unstructured on the between level. Random effects that have non-significant variance (Z score less than 3) should be converted back to non-random. Uncorrelated random effects are discussed in Asparouhov and Muthén (2024a)

4. **Two-level DSEM model with correlated random intercepts and slopes**
   Random effects can be structured or correlated on the between level at this stage.

5. **Two-level DSEM model with random intercepts, slopes and variances**

6. **Two-level DSEM model with random intercepts, slopes, variances and covariances**

The simpler the model is, the easier it is to correctly identify the source of a convergence problem or estimation instability. Gradually building up the model will allow us to quickly identify the model component that caused the problem. That would be the last component that was added to the model.

We recommend that variables are combined gradually. First, a univariate auto-regressive model should be estimated for all the variables of interest. These univariate models can then be combined into bivariate and multivariate auto-regressive models. Hamaker et al. (2021) illustrates in detail such a build up sequence of models. In addition, Hamaker et al. (2021) recommends analyzing the time-series data one person at a time as a preliminary exploration step.

The more complex models are likely to need many more MCMC iterations to converge and are more likely to experience slow or incomplete convergence. For such models, it is important to monitor convergence not only with the PSR convergence criterion used by Mplus by default, but also with the traceplots. Posterior distribution effective sample size of at least 400 is also recommended, see Vehtari et al. (2020).

4 How to account for overnight time

One issue that is often considered in practical applications is how to account for the overnight time with the DSEM model. Suppose that a number of observations are taken each day and that data from multiple days in a row have been recorded. To some extent, this situation is also similar to multiple wave sampling, where the single wave data takes the role of the within day sampling, while the time between waves takes the role of the overnight time. In what follows, we primarily focus on the day and overnight case but the ideas can be used for multiple wave sampling as well.

The most common approach to the overnight issue is to ignore the sleep time completely and treat the timeline as homogeneous. That is, time is recorded in a particular scale of hours or portions of days. We then treat an interval of 8 hours of sleep as if these 8 hours are awake time, or any other

time segment on the timeline that is 8 hours long. Clearly such an approach is problematic as awake time and sleep time are different. Note that this issue is the same whether or not we treat the time as continuous as in CTRDSEM, see Asparouhov and Muthén(2024b), we use DSEM/RDSEM with hours as the time record, or we use TINTERVAL with some sort of interval of time that is different from hours. Neither of the frameworks attempts to differentiate sleep and awake time. Also note that the effect of overnight time is likely to be variable specific. We can not expect that sleep time will affect different variables the same way.

One approach that is sometimes considered appealing is to treat the data as 3-level data where "day" becomes the middle level of nesting for the time-series. This approach can be used in Mplus with 3-level models but not with DSEM where autocorrelation is taken into account. A 3-level DSEM construction is described in Asparouhov (2017), pages 48-53, where the day's data is modeled with a wide approach. This method, however, has some limitations and would not be applicable in all situations. We shall return to this type of modeling later on.

Here we focus on the situation where the timeline is treated as a continuum across days. We describe a method which can be used to determine whether or not there is a qualitative difference between sleep and awake time. Tinterval generally fills in the sleep time with missing values, but it does not take into account the possible qualitative difference between sleep and awake time. This then poses the question, how many missing values should represent the sleep time. If a variable is completely reset from the moment we fall asleep and there is no autocorrelation that carries over from one day to the next, we then may want to insert more missing data than Tinterval provides. If a variable is not completely reset from the moment we fall asleep, but is largely unchanged during the sleep time, then inserting less missing values than Tinterval provides would be beneficial. If the 1-hour autocorrelation is ρ, then AR(1) model predicts that the 8-hour sleep autocorrelation is ρ^8 and missing data is inserted to reflect that expectation. If, however, sleep time is qualitatively different and the overnight correlation does not match that expectation, we would want to adjust the timeline to provide a better treatment for the overnight timeline.

As an illustration, we consider the PA (positive affect) variable discussed in Muthén et al. (2024). We aim to determine if night time autocorrelation is different from autocorrelation across the entire time-line. We will use the CTRDSEM framework in Mplus for this analysis i.e., continuous times of

observations. We compare the autocorrelation for two variables. The first variable is PA as originally recorded in the data. The second variable is the PAON (PA overnight) variable which contains only neighboring observations that include an overnight segment, i.e., we construct a new variable, identical to PA, where we replace the observed values with missing, unless the observations are the last observation from one day and the following observations is first observation in the next day. If sleep and awake time are qualitatively the same, the autocorrelations for the two variables would be equal. Because most of the PAON are missing, we expect that the autocorrelations for PAON would have a much bigger confidence interval. As a preliminary step, we estimate the analysis only with PA to determine the between part of PA. Subsequently, that between part is subtracted from the data so we treat both PA and PAON as within level variables.

Figure 9 contains the input for this analysis and Figure 10 contains the results. While there is no statistically significant difference in the results (as the confidence intervals of the autocorrelation parameters overlap), we see that the point estimates are quite different. The drift parameter for the full data is -0.677, while for the overnight only data it is -1.154. The day time autocorrelation for two observations that are one hour apart is Exp(-0.677)=0.51, while the corresponding value for the night time is estimated at Exp(-1.154)=0.32. The ratio between the drift parameters is $1.7 = (-1.154)/(-0.677)$. This suggests that in terms of "process memory" a night time hour is worth 1.7 hours of day time and the full night time 8 hours of sleep is thus worth 1.7*8=13.6 hours of daytime. The autocorrelation between the hypothetical last possible observation of the day and the first possible observation of the next day (assuming 8 hours of sleep) would match the autocorrelation for two day time observations that are 13.6 hours apart.

In practical terms, we can use this information as follows. Suppose that we are estimating the model using Tinterval of 3 hours. This will split the daytime in 5 periods and the night time in 3 periods for a total of 8 periods in the day. The 3 night time periods are expected to have missing values. The above analysis, however, would suggest that if we are to analyze the PA variable as a continuum, we would be better off if an one additional period (12 hours) is inserted for the night time to reflect the fact that greater memory loss occurs during the night time for the positive affect variable.

In this particular example, however, the additional missing row is probably not necessary. The day time one hour autocorrelation is estimated at 0.5. The autocorrelation in an 8 hour period is then $0.5^8 = 0.004$ while in a 13.6

hour period, it is $0.5^{13.6} = 0.0001$. Both of these are very small and both are essentially indistinguishable from zero. Both imply that the overnight memory of the positive affect variable is essentially completely lost. Thus, inserting an additional row of missing data for the night time is not essential in this case. In fact, one can argue that if the one hour autocorrelation is less than 0.7, the overnight autocorrelation is then less than $0.06 = 0.7^8$. This value is quite small for practical purposes. As long as the variable has a larger memory loss during the night, it is not necessary to make any adjustments to the Tinterval data structure. Both Tinterval and the adjusted data structure would imply nearly a complete overnight reset. Note, however, that if the memory loss overnight is smaller than the day time, the situation would be different and an adjusted data structure might be beneficial.

Figure 9: Overnight time comparison

```
DATA:   FILE = 2.dat;

VARIABLE:
        NAMES = PA PAON hrs day id grp;
        MISSING = ALL (999); ! Missing Values
        USEVAR  = pa paon;
        cluster = id;
        ctime = hrs;
        lagged = pa(1) paon(1);
        within=pa paon;

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

Model:
        %within%
        pa^ on pa^1;
        paon^ on paon^1;
        pa with paon@0;

        %between%
```

Figure 10: Overnight time comparison results

```
                                    Posterior  One-Tailed          95% C.I.
                    Estimate         S.D.      P-Value   Lower 2.5%  Upper 2.5%
Significance
 Drift
  PA^        ON
      PA^1            -0.677        0.029      0.000      -0.734      -0.620      *

  PAON^      ON
      PAON^1          -1.154        0.414      0.000      -2.197      -0.616      *


    AUTOREGRESSIVE CURVES

      PA^ ON PA^1
         T          Estimate     CINT 2.5%    CINT 97.5%
       1.000         0.508        0.480        0.538

      PAON^ ON PAON^1
         T          Estimate     CINT 2.5%    CINT 97.5%
       1.000         0.315        0.111        0.540
```

4.1 The three-level DSEM model

In this section we provide an alternative and possibly simpler treatment for the overnight time modeling. The approach is based on the three-level DSEM model discussed in Asparouhov (2017), pages 48. Suppose that Y_{idt} is the observed value for individual i on day d at time t. The model is given by the following equations

$$Y_{idt} = \mu + Y_i + E_{it} + F_{id} + G_{idt} \tag{1}$$

$$G_{idt} = \rho_1 G_{id,t-1} + \varepsilon_{1,idt} \tag{2}$$

$$F_{id} = \rho_2 F_{i,d-1} + \varepsilon_{2,id}. \tag{3}$$

The model is a three level model because it uses 3-level nesting: observations are nested within days and then nested within individuals. The observed Y_{idt} is separated into 3 latent components: Y_i represents the time invariant value for individual i (random intercept at level 3), F_{id} represents the day specific value for an individual (level 2 random intercept), and G_{idt} represents the deviation from those two values, i.e., the residual not explained by the two random intercepts. The above equation actually has a fourth competent: E_{it}. This component is a time specific deviation that is specific to the individual and the time in the day. For example, an early bird person might have consistently higher values in the early part of the day and a night owl person might have higher values at the later part of the day. This fourth component is not generally necessary can be removed from the model. We expect E_{it} to be small in most situations. However, the estimation setup we use can benefit from having that component as it provides better mixing even when the component is negligible.

The above model has two separate autoregressive parameters: within day ρ_1 and across days ρ_2. The model has just 7 parameters: the variances of the four components in equation (1), the mean parameter μ, and the two autoregressive parameters. The model automatically resolves the overnight modeling as any kind of correlation from one day to the next is explicitly and separately modeled as the autocorrelation between the day's effects F_{id}.

As an illustration, we shall use the positive affect example discussed earlier. The estimation approach discussed in Asparouhov (2017) uses a 2-level DSEM model where the days data is setup in wide format. Since the times of observations in the data are continuous, but the above model uses integer times of observations, we will need to use Tinterval approximation to the

times of observation. In Asparouhov and Muthén (2024b) it is suggested that comparison between CT-DSEM (continuous time) model and DSEM-Tinterval model can be used to fine-tune the approximation. We shall apply this strategy as a first step. We consider DSEM with Tintervals of 1 hour, 2 hours and 3 hours segments. Using a two-level model with a subject specific random intercept we obtain the following values for the one hour autocorrelation. Using continuous times AR=0.55. With Tinterval of 1 hour, the one hour autocorrelation is AR=0.54. With Tinterval of 2 hour, the one hour autocorrelation is AR=0.64 (this is the square root of the two-hour AR estimate). With Tinterval of 3 hour, the one hour autocorrelation is AR=0.72 (the third root of the 3-hour AR estimate). We conclude that only Tinterval of 1 hour can provide a good approximation for the continuous times of observations and cruder values can not. This means that the approach in Asparouhov (2017), would require a multivariate analysis with 24 variables and the day data must be organized as rows with 24 records in each row.

Because the process of setting up the data correctly for this analysis is somewhat complex, we provide the necessary Mplus input files. The estimation is organized in four steps and the Mplus input files are given in Figure 11-14. Step 1 of the estimation is simply running the two-level model with Tinterval of 1 hour to execute the continuous time to integer hours approximation. Step 2 is just a data transformation. Here we compute the variable DAY. In this sample DAY is from 1 to 7. We also compute the variable T which is the hour within the day. This is now an integer from 1 to 24. We also compute an identifier IDDAY for each subject and day. This is simply computed as 10*ID+DAY. Because day is from 1 to 7, such definition of the identifier IDDAY will yield unique values for each individual day record. Step 3 uses the LONGTOWIDE command which orders the days data as rows. Step 4 estimates model (1-3). With the USEOBS command, we exclude individuals that have observed value in only one day.

The Step 4 estimate for ρ_2 is 0.95 and the confidence interval for that parameter includes 1. When the autocorrelation is 1, F_{id} essentially plays the same role as the random intercept Y_i. The covariance structure implied by the time-series on F_{id} is the same as the covariance structure implied by the random intercept Y_i. We conclude that the data is not sufficient to be able to estimate this three level model with separate autocorrelation structures for within day and between days. If the data contained more than 7 days of observations, we would have likely been able to identify the model well, including the two separate autocorrelations. We can nevertheless

estimate a separate model that includes only random intercept for the day's data, essentially fixing ρ_2 to 0. This analysis is given in Figure 15 and the results are given in Figure 16. The one hour autocorrelation is estimated at 0.43, i.e., including a random intercept at the day level reduced the one hour autocorrelation from 0.55 to 0.43.

Equation (1) variance decomposition, in terms of percentages of the total variance, are as follows. The individual effect Y_i accounts for 55% of the variance. The day effect F_{id} accounts for 8% of the variance. The individual time of the day effect E_{it} accounts for 2% of the variance. The remaining 35% of the variance is the time specific deviation G_{idt}. Of these 35%, 7% can be accounted for by the previous hour observations $G_{id,t-1}$ and only 28% are unexplained by the model and can be attributed to random shocks.

Figure 11: Three-level DSEM Step 1

```
DATA:   FILE = adjusted2.csv;

VARIABLE:
        NAMES = PA ID HRS
        MISSING = ALL (999);
        USEVAR  = pa;
        cluster = id;
        tinterval = hrs (1 time);
        lagged = pa(1);

Analysis:
        type = twolevel;
        estimator = bayes;

Model:
        %within%
        pa on pa&1;

        %between%
        [pa]; pa;

savedata: file is 1h.dat;
```

Figure 12: Three-level DSEM Step 2

```
DATA:   FILE = 1h.dat; variance=nocheck;

VARIABLE:
          NAMES = pa z1-z3 hrs id;
          usevar = pa hrs id day t idday;
          missing=*;

define:
if (hrs<=24) then day=1;
if (hrs>24 .and. hrs<=48) then day=2;
if (hrs>48 .and. hrs<=72) then day=3;
if (hrs>72 .and. hrs<=96) then day=4;
if (hrs>96 .and. hrs<=120) then day=5;
if (hrs>120 .and. hrs<=144) then day=6;
if (hrs>144 .and. hrs<=168) then day=7;
t=hrs-(day-1)*24;
idday=id*10+day;

Analysis: type = basic;

savedata: file is 1hs2.dat;
```

Figure 13: Three-level DSEM Step 3

```
data: file=1hs2.dat; variance=nocheck;

variable:
names=pa hr id day t idday;
USEVAR = pa1-pa24;
auxiliary = id day;
MISSING = *;

DATA LONGTOWIDE:
LONG = pa;
WIDE = pa1-pa24;
IDVARIABLE = idday;
repetition=t(1-24);

savedata: file is 1hs3.dat;
```

Figure 14: Three-level DSEM Step 4

```
data: file=1hs3.dat; variance=nocheck;

variable:
names=y1-y24 id day idday;
missing=*;
USEVAR = y1-y24;
cluster=id;
useobs = id ne 238 and id ne 196 and id ne 197 and id ne 549;
Tinterval=day(1);

ANALYSIS: TYPE = TWOLEVEL;
ESTIMATOR=BAYES;
PROCESSORS=2;
biter=(30000);

MODEL:
%WITHIN%
f BY y1-y24@1 (&1);

y1-y24@0.01;

f1 BY y1@1; f2 BY y2@1; f3 BY y3@1;
f4 BY y4@1; f5 BY y5@1; f6 BY y6@1;
f7 BY y7@1; f8 BY y8@1; f9 BY y9@1;
f10 BY y10@1;
f11 BY y11@1; f12 BY y12@1; f13 BY y13@1;
f14 BY y14@1; f15 BY y15@1; f16 BY y16@1;
f17 BY y17@1; f18 BY y18@1; f19 BY y19@1;
f20 BY y20@1;
f21 BY y21@1; f22 BY y22@1; f23 BY y23@1;
f24 BY y24@1;

f1-f24*1 (1);
f*0.5;
f ON f&1*0.3;
f2-f24 pon f1-f23*0.5 (2);
f WITH f24@0;

%BETWEEN%
fb BY y1-y24@1;
fb*0.4;
y1-y24*0.01 (4);
[y1-y24*0] (3)
```

Figure 15: Three-level DSEM Step 5

```
data: file=1hs3.dat; variance=nocheck;

variable:
names=y1-y24 id day idday;
missing=*;
USEVAR = y1-y24;
cluster=id;
useobs = id ne 238 and id ne 196 and id ne 197 and id ne 549;

ANALYSIS: TYPE = TWOLEVEL;
ESTIMATOR=BAYES;
PROCESSORS=2;
biter=(30000);

MODEL:
%WITHIN%
f BY y1-y24@1;

y1-y24@0.01;

f1 BY y1@1; f2 BY y2@1; f3 BY y3@1;
f4 BY y4@1; f5 BY y5@1; f6 BY y6@1;
f7 BY y7@1; f8 BY y8@1; f9 BY y9@1;
f10 BY y10@1;
f11 BY y11@1; f12 BY y12@1; f13 BY y13@1;
f14 BY y14@1; f15 BY y15@1; f16 BY y16@1;
f17 BY y17@1; f18 BY y18@1; f19 BY y19@1;
f20 BY y20@1;
f21 BY y21@1; f22 BY y22@1; f23 BY y23@1;
f24 BY y24@1;

f1-f24*1 (1);
f*0.5;
f2-f24 pon f1-f23*0.5 (2);
f WITH f1-f24@0;

%BETWEEN%
fb BY y1-y24@1;
fb*0.4;
y1-y24*0.01 (4);
[y1-y24*0] (3)
```

Figure 16: Three-level DSEM Step 5 results

	Estimate	Posterior S.D.	One-Tailed P-Value	95% C.I. Lower 2.5%	Upper 2.5%	
Significance						
F2 ON						
F1	0.432	0.029	0.000	0.378	0.493	*
Variances						
F	0.108	0.012	0.000	0.083	0.131	*
F1	0.378	0.011	0.000	0.357	0.399	*
FB	0.738	0.078	0.000	0.604	0.911	*
Intercepts						
Y1	5.698	0.049	0.000	5.598	5.796	*
Residual Variances						
Y1	0.031	0.007	0.000	0.020	0.047	*

4.2 The three-level DSEM model with continuous times

In this section we will estimate a three-level DSEM model which uses the continuous times of observations. Note that the between day autocorrelation is not going to be based on continuous times because the day records are integers. Only the within day autocorrelation will be based on continuous times. The desired model can be formulated as follows. Let Y_{idj} is the j-th observed value for individual i on day d at time t_{idj}.

$$Y_{idj} = \mu + Y_i + F_{id} + G_{idj} \tag{4}$$

$$G_{idj} = Exp(a(t_{idj} - t_{id,j-1}))G_{id,j-1} + \varepsilon_{1,idj} \tag{5}$$

$$F_{id} = \rho_2 F_{i,d-1} + \varepsilon_{2,id}. \tag{6}$$

Here a is the drift parameter. Note that the random effect E_{it} is not included in this model.

We describe a multistage estimation for the above model. The steps are given in Figures 17-20. The first step is similar to the first step from the previous section given in Figures 11, however, we now use continuous time modeling. In this step we also estimate the random effect Y_i. The second step is also similar to the second step from the previous section given in Figures 12. This step is just a data transformation. Here we compute the DAY and IDDAY variables and subtract the effect Y_i from Y_{idj}, thus removing the highest level from the analysis. In the third step, we estimate the same model as in the first step but now the cluster variable is IDDAY. This allows us to estimate the effect F_{id}. We will use the estimates of F_{id} for a time series analysis in the final step. Because there are less than 5 observations per day, the estimates F_{id} are highly variable (larger posterior distributions). It is thus necessary to impute these variables. Instead of analyzing a single estimate of F_{id}, we obtain a sample from the distribution of F_{id} to form imputed data sets. We analyze the imputed data sets in the last step. In this example, we used 10 imputations. For the higher level effect Y_i, imputation is generally not as necessary because Y_i estimates are based on more observations and are less variable.

In the third step we also estimate the drift parameter, because here we account for both Y_i which was subtracted from the data and F_{id} which is explicitly modeled. Thus the estimate of the drift parameter is more reliable in this step then it is in the first step.

In the fourth step, we analyze the multiply imputed data for F_{id} with a two-level DSEM model to determine the value of ρ_2. One key issue here is that the multiply imputed data for F_{id} must be manually reordered in all 10 imputed data sets so that for each individual the day data comes in ascending order. This reordering of the data can be done with Excel or Google Spreadsheet so that the rows are ordered first by ID and then by DAY.

The final results must now be combined. Step 1 yields the estimates for μ and $Var(Y_i)$. Step 3 yields the estimate for a (and $\rho_1 = Exp(a)$) as well as $Var(G_{idj})$. Step 4 yields the estimates for ρ_2 and $Var(\varepsilon_{2,id})$. We present the results in Figure 21. For easier comparison we placed these as if they come from the model in Figure 14. There is direct correspondence between the parameter estimates in Figures 16 and 21 except for two things. The first difference is that in Figure 21, the variance of F1 is $Var(G_{idj})$ while in Figure 16 it is the residual variance $Var(\varepsilon_{1,idj})$. This is a feature of continuous time modeling where the total variance is the model parameter rather than the residual variance. The second difference is that ρ_2 is fixed to 0 in Figures 16.

The one hour autocorrelation of this three level analysis is $\rho_1 = 0.41$ which is close to the result we obtained in the previous section. The across day autocorrelation ρ_2 is not significant and is estimated at 0.04. In the previous section, the estimation of ρ_2 failed because the data was not sufficient to separate Y_i and F_{id}. The estimation in this section, however, prevents that problem because the two effects are never estimated simultaneously. They are estimated in separate stages of the multistage process. Multistage estimation is generally inferior to single stage estimation because it assumes that the estimates from the earlier stage are correct and are fixed to the correct value. In this case, however, the multistage estimation is helpful and we can complete our investigation.

The final variance decomposition for the continuous time three level DSEM is as follows. The individual effect Y_i accounts for 57% of the variance. The day effect F_{id} accounts for 4% of the variance. The remaining 39% of the variance is the time specific deviation G_{idt}. Of this 39%, 7% can be accounted for by the previous hour observations $G_{id,t-1}$ and only 32% are completely unexplained by the model.

All these results are similar to the results obtained in the previous sections. The small differences are likely due to normal variability in the Bayesian estimation, to the Tinterval approximation, to the imperfections of multistage analysis, and to the fact that the model in the previous section

had one additional random effect for the time of day. Nevertheless, the conclusion of the three different analyses we featured here is the same. There is no autocorrelation that occurs for the PA variable overnight. Including a day specific random effect, however, reduces the one hour autocorrelation estimate by 20%. Ultimately this implies that we can treat the timeline across days as a continuum and that three-level analysis is not necessary. The days effect can be modelled without three level analysis as well. Dummy day indicators can be included with random or non-random coefficients, see Muthén et al. (2024). Alternatively, in the case of 5 observations or less as is here, a random intercept can likely be replaced by a lag 2 modeling.

Figure 17: Three-level continuous time DSEM Step 1

```
DATA: FILE = adjusted2.csv;

VARIABLE:
   NAMES = PA ID HRS
   MISSING = ALL (999);
   USEVAR = pa;
   cluster = id;
   ctime = hrs;
   lagged = pa(1);

Analysis:
   type = twolevel;
   estimator = bayes;

Model:

   %within%
   pa on pa&1;

   %between%
   [pa]; pa;

savedata: file is 1.dat; save=FS(200);
```

Figure 18: Three-level continuous time DSEM Step 2

```
DATA:   FILE = 1.dat; variance=nocheck;

VARIABLE:
          NAMES = PA z1 paB z2-z5 hrs id;
          MISSING =*;
          usevar=pa hrs id day idday;

define:
PA=PA-PAB;
if (hrs<=24) then day=1;
if (hrs>24 .and. hrs<=48) then day=2;
if (hrs>48 .and. hrs<=72) then day=3;
if (hrs>72 .and. hrs<=96) then day=4;
if (hrs>96 .and. hrs<=120) then day=5;
if (hrs>120 .and. hrs<=144) then day=6;
if (hrs>144 .and. hrs<=168) then day=7;
idday=id*10+day;


Analysis: type = basic;

savedata: file=2.dat;
```

Figure 19: Three-level continuous time DSEM Step 3

```
DATA:   FILE = 2.dat;

VARIABLE:
          NAMES = PA hrs id day idday;
          MISSING =*;
          USEVAR  = pa;
          cluster = idday;
          ctime = hrs;
          lagged = pa(1);
          auxiliary= id day;

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

Model:
          %within%
          pa^ on pa^1 (a);

          %between%
          pa;

model constraint: new(rho1); rho1=Exp(a);

savedata: file=a*.dat; save=FS(10);
```

Figure 20: Three-level continuous time DSEM Step 4

```
DATA:   FILE = alist.dat; type=imputation;

VARIABLE:
        NAMES = id day PADAY;
        USEVAR  = paDAY;
        cluster = id;
        tinterval = day(1);
        lagged = paday(1);
        within=paday;

Analysis:
        type = twolevel;
        estimator = bayes;
        proc=2; fbiter=300;

Model:
        %within%
        paday^ on paday^1;

        %between%
```

Figure 21: Three-level continuous time DSEM Results

		Estimate	Posterior S.D.	One-Tailed P-Value	95% C.I. Lower 2.5%	Upper 2.5%	
Significance							
F	ON						
F&1		0.036	0.085	0.212	-0.259	0.119	
F2	ON						
F1		0.407	0.024	0.000	0.357	0.457	*
Variances							
F		0.052	0.005	0.000	0.038	0.061	
F1		0.503	0.014	0.000	0.480	0.532	*
FB		0.745	0.077	0.000	0.613	0.921	*
Intercepts							
Y1		5.694	0.059	0.000	5.578	5.811	*

5 Comparing significance levels between standardized and unstandardized results

This issue applies to DSEM/RDSEM as well as standard two-level models for models that include random slopes, i.e., estimated with type=random. Consider the case where a random slope is estimated using the model specification **S | Y on X**. In these circumstances, with the Bayes estimator, Mplus computes cluster specific standardization, which is usually requested with the command **OUTPUT:STAND(CLUSTER)**. That is, when the model contains random slopes, the estimated variance is cluster specific, and thus, the standardized estimates are cluster specific. For convenience, Mplus also reports the average standardized values across all the clusters as well as the standard error for these averages. It is often tempting to then compare the significance level for the average standardized **S | Y on X** value against the mean of **S** in the model estimates. Such a comparison however can not be done. That is, the point estimates of these two quantities are expected to match, but their significance levels are not expected to match.

The easiest way to see the difference is as follows. Let's consider the case of samples with large (infinite) cluster sizes but with a small number of clusters. In these cases, the cluster specific random slope will be estimated precisely. The cluster specific random slope will remain constant across the MCMC iterations while it will vary across clusters. This, however, implies that the average of the random slopes will also not very across MCMC iterations and as a result the average standardized **S | Y on X** will have 0 standard error. On the other hand the mean of **S** will not be zero since the number of clusters is small. The standard error for the mean of **S** will be zero only if the number of clusters is infinite.

Thus, the significance level of these two quantities are not comparable, despite the fact that their point estimates are nearly identical. The average standardized **S | Y on X** should not be treated as a model parameter and to evaluate the significance of the effect. The significance level of this quantity is interpreted as being conditional on precisely these particular clusters being in the sample. It is not conditional on a set of clusters being selected from a superpopulation that is sampled at random, as is the significance level of the model parameter: the mean of **S**.

In conclusion, the significance of the random effect should be primarily evaluated using the model estimated mean of **S** and not with the average stan-

dardized value. In addition, the significance of the cluster specific estimate of **S** might be of practical interest as well. These cluster specific significance levels are driven by the estimates of the random effect as well as the size of the clusters. The Mplus histogram of the random effect distribution includes color coding for the three possible outcomes: cluster specific significantly positive, cluster specific significantly negative and no significance.

6 Conclusion

The DSEM modeling framework is fairly complex and a variety of phenomenons discussed in this article can complicate the analysis and result in estimation problems. Estimating a SEM model, a multilevel regression model, or a time-series model is much simpler than estimating a DSEM model which combines all of these modeling frameworks in one. Approaching the DSEM analysis as if it is a simple regression is going to be futile. The analysis requires plenty of preliminary analysis. If the final model that is desired is attempted as the very first model estimation, it is very likely that estimation problems will occur.

In addition to the estimation complexities of the DSEM model, the analysis is complicated by a variety of data related problems that are specific to intensive longitudinal data (ILD) encountered in practice. It is extremely unusual in multilevel modeling applications to have clusters with very few or even a single observation. ILD data on the other hand often includes such situations. It is extremely unusual in multilevel modeling applications to have all observations within a cluster be identical. For ILD data, this is a fairly common situation. It is quite unusual in multilevel modeling applications to have 80% missing data. For ILD data, this is a fairly common situation because often the times of observation are randomized and are different from person to person. Missing data results from the fact that observations are not recorded in every period.

These data related problems most likely can not be resolved entirely by the final analysis. It would be beneficial if the design of the data collection takes into account the model the data will be analyzed with. For example, the times of observations are often randomized to reduce expectation bias of the responders. It is often hypothesized that subjects respond differently if they know in advance when they need to record observations, i.e., it is hypothesized that the subject response can be inadvertently biased. If such data

are to be analyzed with a VAR model, however, where we study the effects of observations in one period on the observations in the next period, collecting data with clearly identified concept of period would definitely enhance the quality of the study. Thus, randomized times of observations should be limited to being randomized within that concept of period to avoid creating large amounts of missing data.

References

[1] Asparouhov, T. (2017) Dynamic Structural Equation Modeling of Intensive Longitudinal Data Using Mplus Version 8
https://www.statmodel.com/download/Part%206%20Asparouhov.pdf

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

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

[4] Asparouhov, T. & Muthén, B. (2021). Bayesian analysis of latent variable models using Mplus. Technical report. Los Angeles: Muthén & Muthén. http://www.statmodel.com/download/BayesAdvantages18.pdf

[5] Asparouhov, T. & Muthén, B. (2023). Bayesian analysis using Mplus: Technical implementation. Technical appendix. Los Angeles: Muthén & Muthén http://www.statmodel.com/download/Bayes2.pdf

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

[7] Asparouhov, T. & Muthén, B. (2024b). Continuous Time Dynamic Structural Equation Models.
https://www.statmodel.com/download/CTRDSEM.pdf

[8] Hamaker, E. L., Asparouhov, T., Brose, A., Schmiedek, F., & Muthén, B. (2018). At the frontiers of modeling intensive longitudinal data: Dynamic structural equation models for the affective measurements from the COGITO study. Multivariate Behavioral Research, 53, 820–841.

[9] Hamaker, E.L., Asparouhov, T, & Muthén, B. (2021). Dynamic structural equation modeling as a combination of time series modeling, multilevel modeling, and structural equation modeling. To be published as Chapter 31 in: The Handbook of Structural Equation Modeling (2nd edition); Rick H. Hoyle (Ed.); Publisher: Guilford Press.

[10] Muthén, B., Asparouhov, T, & Keijsers, L. (2024) Dynamic Structural Equation Modeling with Cycles. Structural Equation Modeling: A Multidisciplinary Journal. Accepted.

[11] Self S.G. & Liang KY (1987) Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under nonstandard conditions. Journal of the American Statistical Association. 82, 605–610.

[12] Vehtari, A., Gelman, A., Simpson, D., Carpenter, B. & Burkner, P.C. Rank-normalization, folding, and localization: an improved $\hat{R}$ for assessing convergence of MCMC. Bayesian Analysis, 16, 667-718.