# Factor-based SEM building on consistent PLS: An information systems illustration

**Ned Kock**

June 2017

***ScriptWarp Systems™***
Laredo, Texas
USA

# Factor-based SEM building on consistent PLS: An information systems illustration

**Ned Kock**

Full reference:

Kock, N. (2017). *Factor-based SEM building on consistent PLS: An information systems illustration*. Laredo, TX: ScriptWarp Systems.

## Abstract

*Partial least squares (PLS) methods have desirable characteristics that have led to their extensive use in the field of information systems for path analyses with latent variables. Such variables are typically conceptualized as factors in structural equation modeling (SEM). In spite of their desirable characteristics, PLS methods suffer from a fundamental problem: unlike the classic covariance-based SEM, they do not deal with factors, but with composites, and as such do not account for measurement error. This leads to biased parameters, even as sample sizes grow to infinity. We discuss a method that builds on the consistent PLS technique and that deals with factors, fully accounting for measurement error. We provide evidence that this new method shares the properly of statistical consistency with covariance-based SEM, but like traditional PLS methods has greater statistical power. Moreover, our method provides correlation-preserving estimates of the factors, which can be used in a variety of other tests. Our discussion builds on an illustrative model developed based on an information systems theory and related empirical work.*

**KEYWORDS**: Partial Least Squares; Structural Equation Modeling; Measurement Error; Path Bias; Variation Sharing; Monte Carlo Simulation

## Introduction

The field of information systems (IS) is closely associated with the development, implementation, assessment, and use of the partial least squares method (PLS) method (Chin, 1998; Chin et al., 2003; Kock, 2010; Dijkstra & Henseler, 2015). This method, developed by Wold (1980), has been extensively used in IS studies to investigate path models with latent variables (Goodhue et al., 2012; Dijkstra & Henseler, 2015). More often than not latent variables are quantifications of mental constructs, for which multiple imprecise direct measures (indicators) are obtained via questionnaires. In this context, PLS has often been compared with the classic covariance-based approach to structural equation modeling (SEM).

Such comparisons have led to a continuing and often antagonistic debate among proponents and detractors of PLS (Henseler et al., 2014; Rönkkö, M., & Evermann, 2013). While this debate has addressed numerous issues, it has often gravitated around one main problem: PLS and related methods do not deal with factors, which are aggregations of indicators and measurement errors, but with composites. Composites aggregate indicators but do not incorporate measurement error, and thus can only be seen as approximations of factors. In large part because of their focus on composites, PLS methods yield biased estimates of various parameters even as sample sizes grow to infinity. Among these asymptotically biased parameters are path coefficients, indicator weights, and indicator loadings.

Despite this problem PLS methods have some clear advantages over covariance-based SEM, which have led to their growing use. Notably, they virtually always converge to solutions, even with very small sample sizes. This is useful in cases where IS researchers want to investigate small populations (e.g., $N < 50$) to which they have full access, although a combination of weak effects and small sample sizes may lead to problems such as capitalization on error (see, e.g., Goodhue et al., 2007). Also, PLS methods do not normally have identification problems, allowing for the development of fairly complex models and their test with a limited number of indicators.

We make here what we believe to be an important contribution to this debate surrounding PLS methods. We discuss a method that combines elements of current PLS methods and covariance-based SEM, and that provides estimates of the true composites and correlation-preserving factors in a path model. In our method, the factors are estimated so as to preserve their true correlations (see, e.g., DiStefano et al., 2009), which addresses the well known correlation attenuation problem (Nunnaly, 1978; Nunnally & Bernstein, 1994). This problem is characterized by path coefficient estimates that asymptotically converge to values that underestimate the true values (Goodhue et al., 2012).

Our method builds on the consistent PLS technique (Dijkstra & Henseler, 2015), which is a parameter correction technique that is often referred to by the PLSc acronym (the “c” standing for consistent). Nevertheless, our method, which we refer to as PLSF (where the “F” is a reference to its focus on factor estimation), is not a parameter correction technique. Generally speaking, PLS-based parameter correction techniques adjust parameters estimated via PLS methods to correct for bias (Goodhue et al., 2012; Dijkstra & Schermelleh-Engel, 2014; Rönkkö, 2014). Our method estimates prototypical elements, such as factors, which are then used in the production of parameters. As such, no corrections are needed. The consistent PLS technique is used in the estimation of a few coefficients in the early stages of our method; notably the true reliabilities, which are nevertheless critical elements.

Dijkstra & Henseler (2015, p. 17) noted that: “Not only does [IS] research make ample use of PLS as a method of analysis, but also many extensions and advances of PLS can be credited to [IS] researchers.” We agree with this statement, and hope that the PLSF method will be seen as a contribution to this tradition. Our PLSF method is the culmination of several years of research on the basic elements that make it up. Particularly important among those elements is a function that fits a matrix of correlations among true composites to a matrix of correlations among true factors, which will be discussed later. This function relies on reliability measures. Previous attempts have led to approaches that were less accurate under certain conditions, due to relying on biased reliability estimates; or less computationally efficient, due to the need for nested iterations to converge to more accurate reliability estimates. The method discussed here is so far the one with the broadest range of application, and the greatest computational efficiency.

Throughout this discussion we refer to estimates of the corresponding true values, using terms such as “estimates of the true composites”. By this we mean that these are estimates of the corresponding true population values, not approximations of them. As such, these estimates are expected to asymptotically converge to the true values; i.e., progressively approach the true values as sample sizes grow to infinity. In the technical jargon of mathematical statistics, this property is labeled “consistency”. This is a highly desirable property that is not shared by classic PLS methods, unless an infinite number of indicators is used (consistency at large).

We provide evidence that our method shares the property of consistency with covariance-based SEM, but like classic PLS has greater statistical power. Moreover, our method provides estimates of factor scores, which can be used in a variety of other tests. Among such tests are two that have been developed in the field of IS and have been widely used in a variety of fields since their publication: full collinearity tests, which concurrently assess both lateral and vertical collinearity among factors (Kock & Lynn, 2012); and factor nonlinearity tests, where best-fitting nonlinear functions are estimated for each pair of linked factors, and subsequently used in the estimation of nonlinear path coefficients (Guo et al., 2011; Kock, 2010; Moqbel et al., 2013).

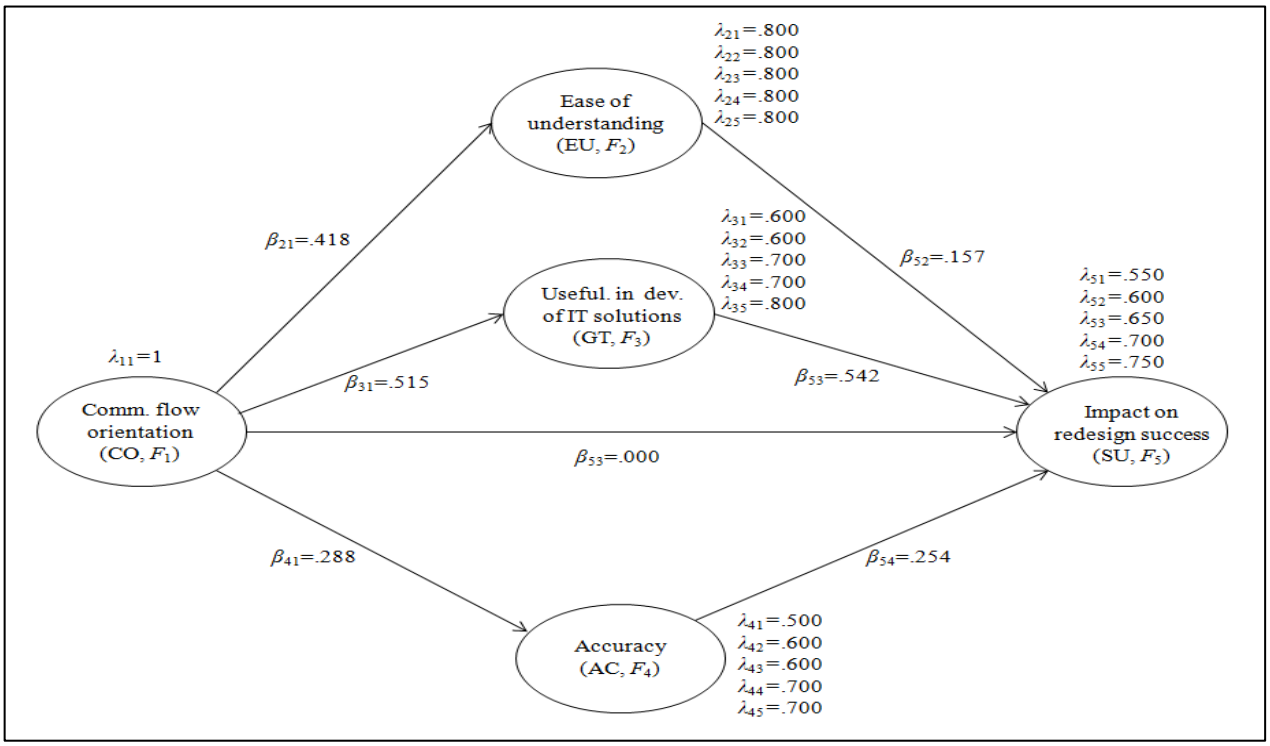
Our discussion is organized as follows. We start by describing an illustrative model, based on an IS theory and related empirical work, which we use as a basis for discussion and to generate data for analyses as a true population model. Next we discuss composites and factors, and how we can go from composites to factors, which is in part what the PLSF method does. A discussion of the PLSF method follows, where it is presented as a set of four stages, each characterized by a function. We proceed with an assessment of the method’s performance against three other methods, including covariance-based SEM; this is done through the juxtaposition of results from analyses of a finite population and from a Monte Carlo experiment. We conclude with a discussion of our findings and its implications. For simplicity, and without any impact on the generality of our discussion, we assume that all variables are standardized – i.e., scaled to have a mean of zero and standard deviation of one.

## Illustrative model

Figure 1 shows an illustrative model that we will use in the discussion that follows. The model is also used later as a basis to generate data for analyses, as a true population model. It is based on data from field studies and controlled experiments (Kock, 2003; 2007; Kock & Murphy, 2001; Kock et al., 2008; 2009), notably a field study involving 156 individuals participating in business process redesign projects employing information technology (IT) solutions to process problems (Kock et al., 2009), and a controlled experiment involving 210 graduate business students majoring in IS (Kock et al., 2008).

The model contains five factors, associated with the following constructs: communication flow orientation (CO, $F_1$), ease of understanding (EU, $F_2$), usefulness in the development of IT solutions (GT, $F_3$), accuracy (AC, $F_4$), and impact on redesign success (SU, $F_5$). This model is based on communication flow optimization theory (Kock, 2003; 2007; Kock & Murphy, 2001; Kock et al., 2008; 2009), a theory that has been developed and validated within the field of IS.

**Figure 1.** Illustrative model




Business processes are sets of interrelated activities (Kock, 2007; Mendling et al., 2012), by which virtually any good or service is produced in organizations. For example, the set of interrelated activities involved in assembling a car, carried out by an automaker, is a business process. Communication flow optimization theory's main domain of application are efforts whereby business processes are analyzed, redesigned, and implemented with IT. One of the theory's main predictions is that the extent to which the business process representations used in these efforts focus on how communication takes place in organizations positively affects redesign success. Among other things, the theory highlights the importance of understanding how information and knowledge flows in organizations in order to successfully redesign the organizations' processes with the help of IT.

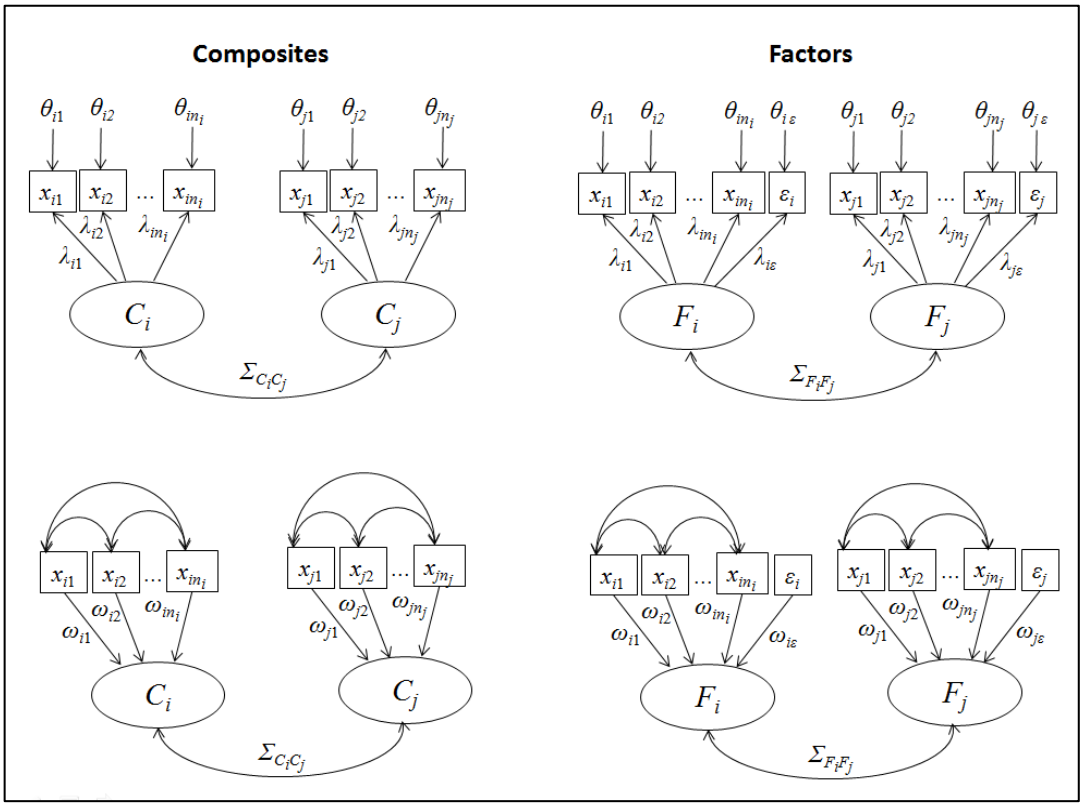
According to the theory the overall effect of communication flow orientation on business process redesign success, with respect to the business process representations used, is fully mediated by a few elements that relate to the representations. Chiefly among these elements are a representation's ease of understanding (CO > EU > SU), usefulness in the development of IT solutions (CO > GT > SU), and accuracy (CO > AC > SU). The full mediation is expressed in the model through a path coefficient of magnitude zero for the direct link CO > SU. That is, the overall effect of CO on SU is fundamentally an indirect effect.

While neither the theory nor the empirical studies that led to it or validated it are the foci of this paper, the fact that our illustrative model is based on a carefully developed and tested theory lends credence to the model's viability. This is important, because the model is reasonably complex. Moreover, using a model base on an IS theory makes our contribution more meaningful to an IS audience.

## From composites to factors

The basic difference between a composite and its corresponding factor is that the composite does not incorporate measurement error, while the factor does (Adelman & Lohmoller, 1994; Kline, 2010; Lohmöller, 1989; McIntosh et al., 2014). This is illustrated in Figure 2, where composites are shown on the left and corresponding factors on the right. This illustration assumes reflective measurement. That is, it assumes that the indicators $x_{i1}$, $x_{i2}$ … $x_{in_i}$ of a factor $F_i$ are derived as imprecise measures of that factor, with this imprecision being characterized by loadings $\lambda_{i1}$, $\lambda_{i2}$ … $\lambda_{in_i}$ lower than 1 and the existence of indicator errors $\theta_{i1}$, $\theta_{i2}$ … $\theta_{in_i}$.

**Figure 2.** Composites and factors




Each measurement error associated with a factor, and stored as a column vector in $\varepsilon_i$, accounts for the variance in the factor that is not explained by the indicators. That is, a factor's measurement error is in fact a residual whose existence is caused by the measurement imprecision of the factors' indicators (Nunnaly, 1978; Nunnally & Bernstein, 1994). As such, each measurement error is, by definition, uncorrelated with its adjacent indicators and correlated

with its factor. Thus a measurement error's weight $\omega_{i\varepsilon}$ equals its loading $\lambda_{i\varepsilon}$. For the indicators, however, typically weights $\omega_{i1}$, $\omega_{i2}$ … $\omega_{in_i}$ and loadings differ in value. In reflective models where data is collected with properly designed questionnaires (i.e., whose answers yield measures with high loadings), normally weights are expected to be lower than loadings (Kock & Mayfield, 2015; Lohmöller, 1989).

Each factor is an aggregation of its indicators and its measurement error, where the latter can be seen as a "residual indicator" that is uncorrelated with the factor's actual indicators. This does not mean that a factor is caused by its indicators; in fact, at the population level in reflective models causality flows in the other direction.

Each factor's corresponding composite, which we refer to here as the factor's "true composite", is an aggregation of the factor's actual indicators. Therefore, each factor is also an aggregation of its true composite and its measurement error. Mathematically this can be expressed as follows, where: $x_i$ is the matrix storing the indicators associated with factor $F_i$ (itself stored as a column vector), $\omega_i$ is a column vector storing indicator weights, $C_i$ is a column vector storing the associated composite, and $\omega_{iC}$ is the composite weight.

$$F_i = \sum_{j=1}^{n_i} x_{ij}\omega_{ij} + \varepsilon_i\omega_{i\varepsilon} \rightarrow$$

$$F_i = x_i\omega_i + \varepsilon_i\omega_{i\varepsilon} \rightarrow$$

$$F_i = C_i\omega_{iC} + \varepsilon_i\omega_{i\varepsilon}.$$

The true reliability associated with each factor $F_i$, which we shall refer to as $\rho_i$, is the amount of variance explained by the true composite (Nunnally & Bernstein, 1994). This quantity can also be referred to as the true composite reliability (Dillon & Goldstein, 1984; Peterson & Yeolib, 2013). Since the true composite and corresponding measurement error are uncorrelated, it follows that the weights associated with each of these entities can be obtained directly from the true reliability. Their relationship can be expressed mathematically as follows, for each given factor.

$$\omega_{iC} = \sqrt{\rho_i},$$

$$\omega_{i\varepsilon} = \sqrt{1 - \rho_i}.$$

In the presence of measurement error we know that, for each pair of true composites $C_i$ and $C_j$, and corresponding factors $F_i$ and $F_j$, the correlation between the composites $\Sigma_{C_iC_j}$ has a lower absolute magnitude than the correlation between the factors $\Sigma_{F_iF_j}$. That is, the fact that the true composites are essentially their corresponding factors with measurement error removed leads to a correlation attenuation (Nunnaly, 1978; Nunnally & Bernstein, 1994). The magnitude of this attenuation is given by:

$$\Sigma_{F_i F_j} = \frac{\Sigma_{C_i C_j}}{\sqrt{\rho_i \rho_j}}.$$

Because of this attenuation, path modeling methods that rely on composites tend to yield path coefficient estimates that asymptotically converge to values that underestimate the true values. It should be noted that this is not always the case, since the attenuation refers to the correlations and not to the path coefficients – e.g., in some cases, correlations and corresponding path coefficients even have different signs (Kock, 2015; Kock & Mayfield, 2015). But why would the *absence* of measurement error lead to a correlation attenuation? Generally speaking, it is the *existence* of error, not its absence, that causes decreases in the correlations among variables.

The reason why the absence of measurement error leads to a correlation attenuation in models with factors is that, as a residual, each measurement error shares variation with its corresponding factor, and thus is correlated with it. In a model with linked factors, this means that a factor's measurement error will also be correlated with other factors. In analyses relying only on composites, these correlations are not captured, leading to the correlation attenuation phenomenon just described.

As we can see from this discussion, estimates of the true composites and factors are obtainable, and researchers do not have to rely on composites in analyses of path models with latent variables. True composite estimates should be very close to exact "replicas" of the true composites. Factor estimates, on the other hand, will not incorporate exactly the same patterns of randomness found in the original factors. Those are unique and unrecoverable (Mueller, 1996). However, while post-estimation random patterns will be unique, they will be reduced to uncorrelated error that will have no effect on any parameter estimation (Bentler & Huang, 2014).

Estimates of the true reliabilities are critical ingredients in the subsequent estimation of true composites and factors, and they are needed early in the estimation process. As demonstrated by Dijkstra & Henseler (2015), the consistent PLS technique provides estimates of the true reliabilities. Moreover, consistent PLS also provides estimates of the true loadings, and approximations of the weights. The latter, while not estimates of the true weights, are closer to the true values than unity weights, and thus add to the computational efficiency of our PLSF method.

## The PLSF method

The PLSF method generates estimates of the true composites and factors, with these serving as the foundation for the production of asymptotically unbiased estimates of various model parameters. The method can be seen as being comprised of four main functions: $\mathcal{F}_1$, the consistent PLS function; $\mathcal{F}_2$, the true composite estimation function; $\mathcal{F}_3$, the true factor estimation function; and $\mathcal{F}_4$, the full parameter estimation function. The execution of each function refers to a PLSF stage, for a total of four stages.

### Function $\mathcal{F}_1$: The consistent PLS function

This function, expressed in equation form below, takes as inputs the matrix $x$ of all indicators, and the matrix $\mathcal{S}$ containing the model specification. The matrix $x$ has $N$ rows, where $N$ is the sample size; and one column for each of the indicators in the model. The matrix $\mathcal{S}$ is made up of two sub-matrices: one specifying factor-factor associations, and the other specifying indicator-

factor associations – i.e., specifying the structural and measurement model links respectively. The outputs of function $\mathcal{F}_1$ include a column vector $\hat{\rho}$ containing estimates of the true reliabilities associated with all of the factors in the model, and a matrix $\hat{\lambda}$ of estimates of the true loadings for all factors. This function also produces initial estimates of the matrices $\hat{C}$ and $\widehat{\omega}$ of composites and indicator weights, based on the basic design of PLS Mode A, which will be used as starting values in the next stage.

$$\left[\hat{\rho}, \hat{\lambda}, \hat{C}, \widehat{\omega}\right] = \mathcal{F}_1(x, \mathcal{S}).$$

The consistent PLS technique is discussed in detail by Dijkstra & Schermelleh-Engel (2014), and Dijkstra & Henseler (2015). The corresponding function $\mathcal{F}_1$ produces its outputs by first estimating composite weights via the basic design of PLS Mode A (Lohmöller, 1989, p. 29), also known as PLS Mode A employing the centroid scheme. Then estimates of the true reliabilities and loadings are generated.

### Function $\mathcal{F}_2$: The true composite estimation function

This function takes as inputs $x$, $\hat{\rho}$, $\hat{\lambda}$, $\hat{C}$ and $\widehat{\omega}$. The composites in the matrix $\hat{C}$ and the indicator weights in the matrix $\widehat{\omega}$ are used as initial values, whereas the reliabilities in $\hat{\rho}$ and loadings in $\hat{\lambda}$ are fixed across the iterations carried out within $\mathcal{F}_2$. As expressed in equation form below, the outputs of this function comprise the following model-wide estimates of the corresponding true values: a matrix $\hat{C}$ of composites, a matrix $\widehat{\omega}$ of weights, vectors $\widehat{\omega}_C$ and $\widehat{\omega}_\varepsilon$ of composite and measurement error weights respectively, and a matrix $\hat{\varepsilon}$ of measurement errors.

$$\left[\hat{C}, \widehat{\omega}, \widehat{\omega}_C, \hat{\varepsilon}, \widehat{\omega}_\varepsilon\right] = \mathcal{F}_2\left(x, \hat{\rho}, \hat{\lambda}, \hat{C}, \widehat{\omega}\right).$$

It is clear from our previous discussion on composites and factors that each true composite is completely determined by its indicators, aggregated based on appropriate weights. The indicators are uncorrelated with the corresponding measurement error. Therefore, the matrix $\hat{\varepsilon}$ produced and initially used internally by $\mathcal{F}_2$ is at first a matrix of random uncorrelated "noise", which at the conclusion of $\mathcal{F}_2$ stores measurement errors that are correlated only with their corresponding factors. In this stochastic approach to estimation, the measurement errors are necessary for the proper estimation of the true composites in $\mathcal{F}_2$, through iterations of three key equations until successive estimates of each of the elements in the weight vectors $\widehat{\omega}_i$ that make up $\widehat{\omega}$ change by less than a small fraction:

$$\hat{F}_i = Stdz\left(\hat{C}_i \widehat{\omega}_{iC} + \hat{\varepsilon}_i \widehat{\omega}_{i\varepsilon}\right),$$

$$\hat{\theta}_i = x_i - \hat{F}_i \hat{\lambda}_i{}',$$

$$\widehat{\omega}_i = \Sigma_{x_i x_i}{}^{-1} \left(\Sigma_{x_i x_i} - diag\left(\Sigma_{x_i \hat{\theta}_i}\right)\right) \hat{\lambda}_i{}'^{+},$$

where for each composite $\hat{C}_i$ we have: $\hat{F}_i$ as its corresponding factor, $\hat{\theta}_i$ as the matrix of estimated indicator errors, $\Sigma_{x_i x_i}$ as the covariance matrix of the indicators associated with the factor, and $\Sigma_{x_i \hat{\theta}_i}$ as the matrix of estimated covariances among indicators and their errors. The function $Stdz(\cdot)$ denotes the standardization function, and $diag(\cdot)$ returns the diagonal of a matrix, the superscript $'$ denotes the transpose operation, the superscript $-1$ the classic matrix inversion, and the superscript $+$ the Moore–Penrose pseudoinverse transformation. See Appendix A for the derivation of these equations.

### Function $\mathcal{F}_3$: The true factor estimation function

This function takes as inputs $\hat{\rho}$, $\hat{C}$, $\hat{\omega}_C$, $\hat{\varepsilon}$ and $\hat{\omega}_\varepsilon$. As indicated below, the outputs of this function are the final estimates of the matrix of factors $\hat{F}$ and the matrix of measurement errors $\hat{\varepsilon}$. These final estimates will contain all of the model-implied variation that is reflected in the model's key "signature" employed by the PLSF method. This model "signature" is $\hat{\Sigma}_{FF}$, the estimated matrix of correlations among factors, calculated within $\mathcal{F}_3$ based on the matrix of correlations among estimated composites $\Sigma_{\hat{C}\hat{C}}$ and the vector of reliabilities $\hat{\rho}$.

$$\left[\hat{F}, \hat{\varepsilon}\right] = \mathcal{F}_3\left(\hat{\rho}, \hat{C}, \hat{\omega}_C, \hat{\varepsilon}, \hat{\omega}_\varepsilon\right).$$

The final estimates of $\hat{F}$ and $\hat{\varepsilon}$ are generated within $\mathcal{F}_3$ through iterations of the three main equations below, whereby the matrix of correlations among estimated factors $\Sigma_{\hat{F}\hat{F}}$ is fitted to the estimated matrix of correlations among factors $\hat{\Sigma}_{FF}$. While the former (i.e., $\Sigma_{\hat{F}\hat{F}}$) varies across iterations, the latter (i.e., $\hat{\Sigma}_{FF}$) is calculated early in $\mathcal{F}_3$ and kept unchanged thereafter within $\mathcal{F}_3$. The iterations continue until the sum of the absolute differences $\hat{\Sigma}_{F_i F_j} - \Sigma_{\hat{F}_i \hat{F}_j}$ falls below a small fraction, or until the sum of the absolute differences between successive estimates of $\Sigma_{\hat{F}_i \hat{F}_j}$ changes by less than a small fraction.

$$\hat{\varepsilon}_i = Stdz\left(\hat{\varepsilon}_i + \left(\hat{\Sigma}_{F_i F_j} - \Sigma_{\hat{F}_i \hat{F}_j}\right)\frac{\hat{\Sigma}_{F_i F_j}}{\hat{\omega}_{i\varepsilon}}\left(\hat{C}_j \hat{\omega}_{jC} + \hat{\varepsilon}_j \hat{\omega}_{j\varepsilon}\right)\right),$$

$$\hat{F}_i = Stdz\left(\hat{F}_i + \left(\hat{\omega}_{iC} - \Sigma_{\hat{F}_i \hat{C}_i}\right)\hat{C}_i \hat{\omega}_{iC}\right),$$

$$\hat{\varepsilon}_i = Stdz\left(\hat{\varepsilon}_i - \Sigma_{\hat{C}_i \hat{\varepsilon}_i}\hat{C}_i \hat{\omega}_{iC} + \left(\hat{\omega}_{i\varepsilon} - \Sigma_{\hat{F}_i \hat{\varepsilon}_i}\right)\hat{F}_i \hat{\omega}_{i\varepsilon}\right).$$

The above are labeled "variation sharing" equations. Through them successive estimates of factors $\hat{F}_i$ and measurement errors $\hat{\varepsilon}_i$ acquire or lose variation from correlated factors, composites, and measurement errors (denoted as $\hat{F}_j$, $\hat{C}_j$ and $\hat{\varepsilon}_j$); in such a way that the following constraints are enforced: $\hat{\Sigma}_{F_i F_j} = \Sigma_{\hat{F}_i \hat{F}_j}$, $\Sigma_{\hat{F}_i \hat{C}_i} = \hat{\omega}_{iC}$, $\Sigma_{\hat{F}_i \hat{\varepsilon}_i} = \hat{\omega}_{i\varepsilon}$, and $\Sigma_{\hat{C}_i \hat{\varepsilon}_i} = 0$. The first constraint, namely $\hat{\Sigma}_{F_i F_j} = \Sigma_{\hat{F}_i \hat{F}_j}$, drives the iterative convergence process. See Appendix A for the derivation of these equations.

### Function $\mathcal{F}_4$: The full parameter estimation function

This function, expressed in equation form below, marks the final stage of the PLSF method. It ensures that all estimates produced are internally consistent, by taking as inputs $x$, $\hat{F}$, $\widehat{\omega}_C$, $\hat{\varepsilon}$ and $\widehat{\omega}_\varepsilon$. Based on these inputs, notably $\hat{F}$ and $\hat{\varepsilon}$, it re-estimates $\hat{C}$, $\widehat{\omega}$ and $\hat{\lambda}$.

$$\left[\hat{C}, \widehat{\omega}, \hat{\lambda}, \hat{\beta}, \hat{\theta}, \hat{\zeta}\right] = \mathcal{F}_4\left(x, \hat{F}, \widehat{\omega}_C, \hat{\varepsilon}, \widehat{\omega}_\varepsilon\right).$$

Additionally, function $\mathcal{F}_4$ produces a matrix of estimates of the path coefficients $\hat{\beta}$, indicator residuals $\hat{\theta}$, and endogenous factor residuals $\hat{\zeta}$. These estimates are obtained by solving the equation below for each endogenous factor $\hat{F}_i$, where $N_i$ is the number of factors $\hat{F}_j$ ($j = 1 \ldots N_i$) pointing at $\hat{F}_i$ in the model. The instrumental variables $\hat{I}_i$ implement a two-stage least squares estimation, and exist for all endogenous factors in the model that contain variation from other factors but are not directly linked with those factors (including factors involved in reciprocal relationships, in non-recursive models). These instrumental variables control for in-model endogeneity, and their corresponding path coefficients $\hat{\beta}_i$ allow for endogeneity significance tests. The indicator residuals in $\hat{\theta}$ and the residuals in $\hat{\zeta}$ are subsequently obtained directly based on these factor estimates.

$$\hat{F}_i = \sum_{j=1}^{N_i} \hat{\beta}_{ij}\, \hat{F}_j + \hat{\beta}_i \hat{I}_i + \hat{\zeta}_i.$$

At the end of the four stages that make up the PLSF method we have estimates of the true values of various parameters stored in the following: $\hat{F}$, $\hat{C}$, $\hat{\varepsilon}$, $\hat{\zeta}$, $\widehat{\omega}$, $\widehat{\omega}_C$, $\widehat{\omega}_\varepsilon$, $\hat{\lambda}$, $\hat{\beta}$ and $\hat{\theta}$. In Appendix B we provide all of the steps and equations that make up the PLSF method, for each of the four functions, as well as the algorithmic sequence of their execution and explanatory notes. This should facilitate the implementation of the method in numeric computing environments such as R and GNU Octave.

## Finite population illustration

A normal finite population ($N$=10,000) was created, based on the illustrative model described earlier, to demonstrate the performance of the PLSF method vis-à-vis other methods. A finite population of this size incorporates only a small amount of sampling error, and has the advantage of allowing us to calculate the values of various true model parameters that can be used in a preliminary assessment of the PLSF method's ability to generate estimates of the true factors (minus uncorrelated error). Among these parameters are path coefficients, full collinearity variance inflation factors (VIFs), loadings, and weights. The disadvantage of using a finite population is that it does not exactly replicate the properties of the infinite population from which it derives, which is why we also conducted a classic Monte Carlo experiment to assess the PLSF method.

Full collinearity VIFs were added to our analysis due to their importance in tests of empirical data, as they assess collinearity among all factors in a model (Kock & Lynn, 2012), and also due to the fact that their magnitude of variation and dependence on the estimates of all factor scores make them particularly sensitive to factor estimation problems. Full collinearity VIFs allow

researchers to identify both vertical and lateral collinearity in models. Vertical, or classic, collinearity reflects redundancy among predictors in a model with various factors. Lateral collinearity reflects redundancy among predictors and criteria.

**Table 1.** Path coefficients and full collinearity VIFs for finite population (*N*=10,000)

| Path coefficients | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | PLSF | | FIML | | OLS | | PLS | |
| | True | Est. | Diff. | Est. | Diff. | Est. | Diff. | Est. | Diff. |
| CO>EU | 0.4223 | 0.4208 | -0.0015 | 0.4188 | -0.0036 | 0.3971 | -0.0253 | 0.3971 | -0.0252 |
| CO>GT | 0.5074 | 0.5066 | -0.0008 | 0.5085 | 0.0012 | 0.4575 | -0.0499 | 0.4599 | -0.0474 |
| CO>AC | 0.2947 | 0.3021 | 0.0074 | 0.3041 | 0.0095 | 0.2661 | -0.0286 | 0.2673 | -0.0274 |
| CO>SU | 0.0146 | 0.0137 | -0.0009 | 0.0132 | -0.0014 | 0.0917 | 0.0771 | 0.0899 | 0.0753 |
| EU>SU | 0.1466 | 0.1479 | 0.0013 | 0.1477 | 0.0011 | 0.1206 | -0.0259 | 0.1219 | -0.0247 |
| GT>SU | 0.5356 | 0.5331 | -0.0025 | 0.5262 | -0.0095 | 0.3983 | -0.1373 | 0.4022 | -0.1334 |
| AC>SU | 0.2562 | 0.2565 | 0.0003 | 0.2664 | 0.0102 | 0.2025 | -0.0537 | 0.2040 | -0.0522 |
| RMSE | | | 0.0031 | | 0.0066 | | 0.0679 | | 0.0659 |
| Full collinearity VIFs | | | | | | | | | |
| | | PLSF | | FIML | | OLS | | PLS | |
| | True | Est. | Diff. | Est. | Diff. | Est. | Diff. | Est. | Diff. |
| CO | 1.6618 | 1.6752 | 0.0135 | 1.8265 | 0.1648 | 1.5451 | -0.1167 | 1.5489 | -0.1128 |
| EU | 1.2575 | 1.2541 | -0.0034 | 1.3119 | 0.0544 | 1.2084 | -0.0491 | 1.2091 | -0.0485 |
| GT | 1.8865 | 1.8921 | 0.0055 | 2.4263 | 0.5398 | 1.4966 | -0.3899 | 1.5062 | -0.3803 |
| AC | 1.2186 | 1.2181 | -0.0005 | 1.3803 | 0.1616 | 1.1364 | -0.0823 | 1.1384 | -0.0803 |
| SU | 1.8813 | 1.8892 | 0.0079 | 2.5014 | 0.6201 | 1.4590 | -0.4223 | 1.4687 | -0.4127 |
| RMSE | | | 0.0076 | | 0.3827 | | 0.2658 | | 0.2594 |

**Table 2.** Summarized loadings and weights for finite population (*N*=10,000)

| Loadings | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | PLSF | | FIML | | OLS | | PLS | |
| | True | Est. | Diff. | Est. | Diff. | Est. | Diff. | Est. | Diff. |
| AC1<AC | 0.4955 | 0.5007 | 0.0052 | 0.5108 | 0.0153 | 0.6529 | 0.1574 | 0.6157 | 0.1202 |
| AC2<AC | 0.5959 | 0.6005 | 0.0046 | 0.6036 | 0.0077 | 0.7050 | 0.1091 | 0.7059 | 0.1100 |
| AC3<AC | 0.5986 | 0.5969 | -0.0017 | 0.5964 | -0.0022 | 0.7000 | 0.1013 | 0.6988 | 0.1001 |
| AC4<AC | 0.7003 | 0.6999 | -0.0004 | 0.7002 | -0.0001 | 0.7531 | 0.0528 | 0.7721 | 0.0718 |
| AC5<AC | 0.7010 | 0.6981 | -0.0028 | 0.6945 | -0.0064 | 0.7513 | 0.0504 | 0.7647 | 0.0638 |
| RMSE | | | 0.0034 | | 0.0082 | | 0.1022 | | 0.0957 |
| Weights | | | | | | | | | |
| | | PLSF | | FIML | | OLS | | PLS | |
| | True | Est. | Diff. | Est. | Diff. | Est. | Diff. | Est. | Diff. |
| AC1<AC | 0.1385 | 0.1387 | 0.0003 | - | - | 0.2807 | 0.1423 | 0.2275 | 0.0890 |
| AC2<AC | 0.2077 | 0.2132 | 0.0055 | - | - | 0.2807 | 0.0730 | 0.2788 | 0.0711 |
| AC3<AC | 0.2174 | 0.2171 | -0.0003 | - | - | 0.2807 | 0.0634 | 0.2748 | 0.0574 |
| AC4<AC | 0.3168 | 0.3128 | -0.0040 | - | - | 0.2807 | -0.0361 | 0.3123 | -0.0045 |
| AC5<AC | 0.3197 | 0.3144 | -0.0053 | - | - | 0.2807 | -0.0390 | 0.3007 | -0.0190 |
| RMSE | | | 0.0038 | | - | | 0.0805 | | 0.0577 |

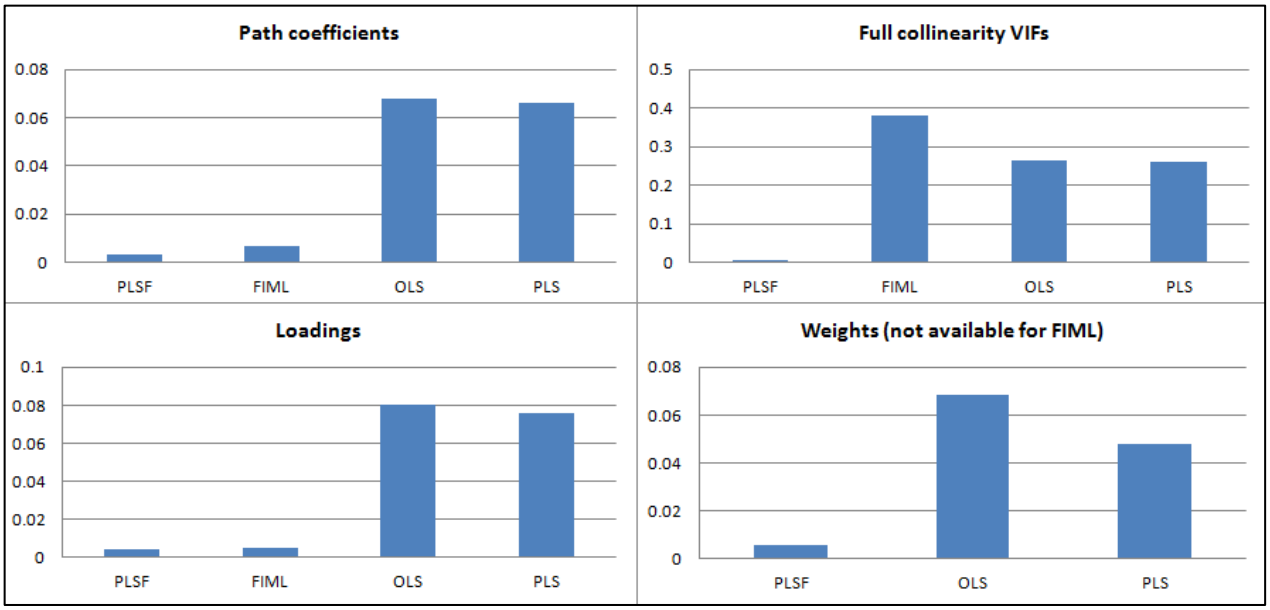
The methods against which PLSF is compared are: covariance-based SEM through full-information maximum likelihood (FIML); ordinary least squares regression with summed indicators (OLS); and PLS Mode A employing the path weighting scheme (PLS). The latter is the most widely used form of PLS path modeling approach employed in the field of IS (Goodhue

et al., 2012). We used pre-tested MATLAB 8.4 code from a widely used commercial software for the implementation of the OLS and PLS methods. We developed our own implementation of PLSF, also with MATLAB 8.4. For FIML, we used R 3.2.2 and the package lavaan 0.5-19 (Rosseel, 2012). We employed the same analysis settings as Dijkstra & Henseler (2015), who compared a similar set of methods.

Table 1 lists the path coefficients and full collinearity VIFs for the finite population. The FIML method does not estimate factor scores, which are needed to calculate the full collinearity VIFs. Several unrefined and refined methods exist to generate correlation-preserving approximations of factor scores based on FIML outputs (DiStefano et al., 2009). We employed two refined methods available in lavaan, the Thurstone and Bartlett methods (DiStefano et al., 2009; Bartlett, 1937; Hershberger, 2005; Thurstone, 1935). Only the Thurstone method yielded solutions for our model. The reason for this may be that the Bartlett method requires multiple matrix inversions, including nested inversions (DiStefano et al., 2009, p. 10), which make it inherently unstable. According to a seminal discussion by Bartholomew et al. (2009), both methods tend to yield very similar results; and the Thurstone method, also known as Thomson's method, has a more sound mathematical basis.

Table 2 lists a summarized set of loadings and weights for the finite population. To avoid crowding, and since the patterns observed here repeat themselves across latent variables and indicators, this summarized set focuses on AC and its respective indicators AC1, AC2 ... AC5. In our model AC has the lowest overall set of loadings, and thus potentially poses the most estimation challenges for the PLSF method. The FIML method does not generate estimates of weights, which is why they are not listed in the table.

**Figure 3.** Differences (RMSEs) with respect to true values




In each table the column labeled "True" lists the true values in our finite population of the path coefficients and full collinearity VIFs. The "Est." columns list the corresponding estimates employing each method. The "Diff." columns list the differences between estimates and true values for each method. The row labeled "RMSE" lists root-mean-square errors associated with

the differences between estimates, calculated as the square roots of the averages of the squared differences, which provide a summarized performance measure for each of the methods. Figure 3 highlights the differences (RMSEs) with respect to true values for each of the methods.

As we can see, the performances of PLSF and FIML were similar in terms of estimation of path coefficients. In this respect, these two methods (i.e., PLSF and FIML) performed significantly better than OLS and PLS, whose corresponding RMSEs were multiple orders of magnitude higher. In terms of full collinearity VIFs the PLSF method performed significantly better than the other three methods, with the performance of FIML being the poorest.

The performances of PLSF and FIML were again comparable in terms of loadings, based on their RMSEs, which also suggest that PLSF and FIML performed significantly better than OLS and PLS. Again, the RMSEs for OLS and PLS were multiple orders of magnitude higher. The same pattern is observed with respect to weights for the PLSF method, when compared with the OLS and PLS methods. The FIML method does not generate weights.

## Monte Carlo experiment

While the analyses of the finite population provide an idea of the comparative performance of the four methods, a full Monte Carlo experiment (Paxton et al., 2001; Robert & Casella, 2005) is needed to assess performance in terms of statistical power and percentages of false positives; as well as in terms of estimation of path coefficients with respect to an infinite population, where the distorting effect of sampling error is minimized.

We generated 1,000 samples of normal and non-normal data with the following sample sizes: 100, 300 and 500. The non-normal samples were created based on independent $\chi_1^2$ distributions, with theoretical skewness and excess kurtosis values of $\sqrt{8}$ and 12 respectively, and thus severely non-normal. Exogenous factors, endogenous factor errors, and indicator errors were created independently from one another to ensure proper non-normality propagation (Kock, 2016). We also conducted two tests of normality on these variables in each non-normal sample: the classic Jarque-Bera test (Jarque & Bera, 1980; Bera & Jarque, 1981) and Gel & Gastwirth's (2008) robust version of this classic test. These tests confirmed the presence of significant non-normality.

We refer to the sample sizes of 100, 300 and 500 respectively as small, medium and large. Our simulated data generation yielded a total of 6,000 data samples, which were analyzed with the PLSF, FIML, OLS and PLS methods. With normal data the FIML method converged to solutions in all samples, and with non-normal data it failed to converge to solutions in 6.1% of the samples. The PLSF, OLS and PLS methods converged to solutions in all samples, both normal and non-normal.

Tables 3 and 4 show, for each of the path coefficients in our illustrative population model described earlier, the following estimates: the average difference between the path coefficient estimated by each method and the true values (rows labeled "Avg. diff."); the statistical power of each method (rows labeled "Power"); the standard deviation of the estimate (rows labeled "Std. dev."); the percentage of false positives yielded by each method for the path whose true value is zero (rows labeled "False pos."); and, in the final rows at the bottom, the RMSE for each method, calculated based on the average differences. Results for normal and non-normal data are shown.

**Table 3.** Monte Carlo experiment results for path coefficients (normal data)

| Sample size | 100 | | | | 300 | | | | 500 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | PLSF | FIML | OLS | PLS | PLSF | FIML | OLS | PLS | PLSF | FIML | OLS | PLS |
| CO>EU (0.418) | | | | | | | | | | | | |
| Avg. diff. | 0.0107 | -0.0047 | -0.0214 | -0.0145 | 0.0035 | -0.0017 | -0.0221 | -0.0198 | 0.0015 | -0.0022 | -0.0225 | -0.0210 |
| Power | 100.0% | 98.3% | 99.9% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% |
| Std. dev. | 0.0782 | 0.0888 | 0.0764 | 0.0749 | 0.0474 | 0.0501 | 0.0454 | 0.0452 | 0.0372 | 0.0379 | 0.0354 | 0.0353 |
| CO>GT (0.515) | | | | | | | | | | | | |
| Avg. diff. | 0.0017 | -0.0005 | -0.0554 | -0.0462 | 0.0044 | 0.0007 | -0.0505 | -0.0459 | 0.0040 | 0.0012 | -0.0499 | -0.0462 |
| Power | 100.0% | 99.8% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% |
| Std. dev. | 0.0765 | 0.0810 | 0.0699 | 0.0695 | 0.0423 | 0.0482 | 0.0389 | 0.0384 | 0.0328 | 0.0353 | 0.0299 | 0.0297 |
| CO>AC (0.288) | | | | | | | | | | | | |
| Avg. diff. | 0.0028 | 0.0049 | -0.0402 | -0.0174 | 0.0026 | 0.0084 | -0.0402 | -0.0320 | 0.0038 | 0.0080 | -0.0368 | -0.0316 |
| Power | 85.9% | 72.4% | 76.8% | 86.3% | 100.0% | 99.4% | 99.8% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% |
| Std. dev. | 0.1004 | 0.1057 | 0.0912 | 0.0879 | 0.0586 | 0.0610 | 0.0520 | 0.0516 | 0.0443 | 0.0465 | 0.0393 | 0.0389 |
| CO>SU (0.000) | | | | | | | | | | | | |
| Avg. diff. | -0.0034 | -0.0043 | 0.0848 | 0.0688 | -0.0100 | 0.0010 | 0.0814 | 0.0742 | -0.0052 | -0.0040 | 0.0836 | 0.0784 |
| False pos. | 5.2% | 5.5% | 11.6% | 8.9% | 3.7% | 1.9% | 27.2% | 23.6% | 4.1% | 2.5% | 50.0% | 44.5% |
| Std. dev. | 0.1398 | 0.1337 | 0.1089 | 0.1103 | 0.0764 | 0.0700 | 0.0606 | 0.0608 | 0.0553 | 0.0531 | 0.0441 | 0.0441 |
| EU>SU (0.157) | | | | | | | | | | | | |
| Avg. diff. | 0.0092 | 0.0009 | -0.0295 | -0.0238 | 0.0064 | 0.0016 | -0.0263 | -0.0241 | 0.0014 | 0.0025 | -0.0290 | -0.0273 |
| Power | 33.7% | 22.7% | 30.5% | 33.2% | 73.8% | 57.8% | 71.5% | 72.0% | 88.1% | 81.2% | 86.3% | 87.7% |
| Std. dev. | 0.1092 | 0.1112 | 0.0888 | 0.0898 | 0.0629 | 0.0610 | 0.0520 | 0.0522 | 0.0502 | 0.0467 | 0.0421 | 0.0419 |
| GT>SU (0.542) | | | | | | | | | | | | |
| Avg. diff. | -0.0029 | -0.0078 | -0.1427 | -0.1258 | 0.0053 | -0.0130 | -0.1372 | -0.1279 | 0.0065 | -0.0084 | -0.1367 | -0.1287 |
| Power | 99.6% | 98.3% | 99.6% | 99.8% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% |
| Std. dev. | 0.1120 | 0.1152 | 0.0836 | 0.0844 | 0.0663 | 0.0604 | 0.0498 | 0.0497 | 0.0496 | 0.0492 | 0.0379 | 0.0375 |
| AC>SU (0.254) | | | | | | | | | | | | |
| Avg. diff. | 0.0049 | 0.0103 | -0.0619 | -0.0434 | 0.0052 | 0.0126 | -0.0612 | -0.0537 | 0.0033 | 0.0093 | -0.0620 | -0.0568 |
| Power | 61.4% | 58.8% | 60.6% | 67.7% | 98.5% | 97.5% | 98.5% | 98.6% | 100.0% | 99.8% | 99.9% | 100.0% |
| Std. dev. | 0.1146 | 0.1086 | 0.0859 | 0.0881 | 0.0601 | 0.0610 | 0.0460 | 0.0461 | 0.0470 | 0.0495 | 0.0360 | 0.0359 |
| RMSE | 0.0060 | 0.0058 | 0.0731 | 0.0605 | 0.0058 | 0.0076 | 0.0702 | 0.0642 | 0.0041 | 0.0060 | 0.0704 | 0.0657 |

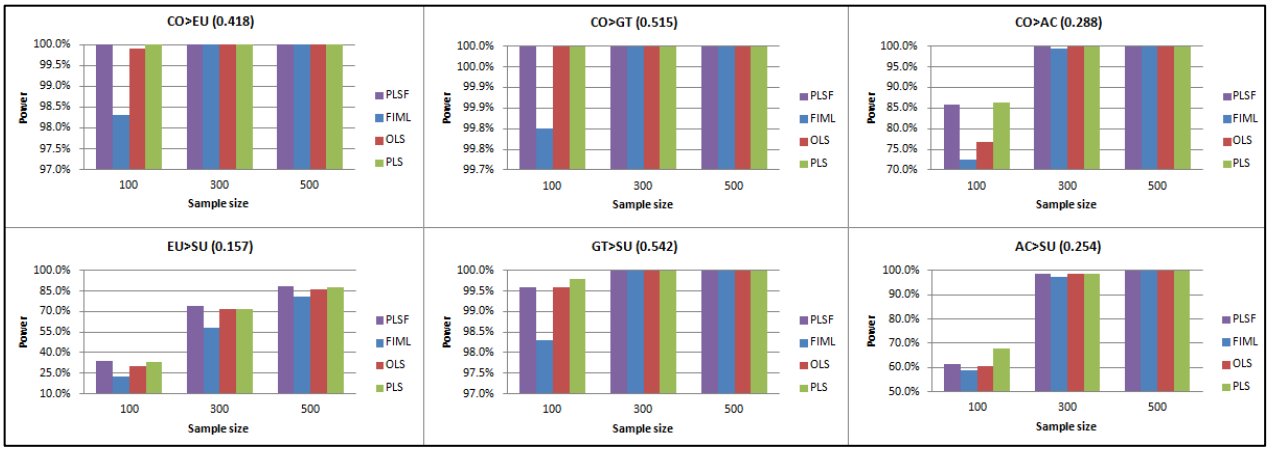
**Table 4.** Monte Carlo experiment results for path coefficients (non-normal data)

| Sample size | 100 | | | | 300 | | | | 500 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | PLSF | FIML | OLS | PLS | PLSF | FIML | OLS | PLS | PLSF | FIML | OLS | PLS |
| CO>EU (0.418) | | | | | | | | | | | | |
| Avg. diff. | 0.0096 | -0.0036 | -0.0237 | -0.0168 | 0.0042 | -0.0011 | -0.0215 | -0.0191 | 0.0016 | -0.0035 | -0.0223 | -0.0210 |
| Power | 100.0% | 93.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% |
| Std. dev. | 0.0810 | 0.1218 | 0.0780 | 0.0771 | 0.0459 | 0.0710 | 0.0435 | 0.0435 | 0.0348 | 0.0569 | 0.0330 | 0.0330 |
| CO>GT (0.515) | | | | | | | | | | | | |
| Avg. diff. | 0.0016 | -0.0042 | -0.0572 | -0.0488 | 0.0051 | -0.0065 | -0.0498 | -0.0452 | 0.0031 | -0.0042 | -0.0509 | -0.0472 |
| Power | 100.0% | 98.4% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% |
| Std. dev. | 0.0742 | 0.1223 | 0.0674 | 0.0671 | 0.0439 | 0.0696 | 0.0402 | 0.0400 | 0.0341 | 0.0549 | 0.0312 | 0.0310 |
| CO>AC (0.288) | | | | | | | | | | | | |
| Avg. diff. | 0.0099 | -0.0094 | -0.0438 | -0.0221 | 0.0073 | -0.0137 | -0.0359 | -0.0280 | 0.0014 | -0.0112 | -0.0391 | -0.0338 |
| Power | 83.0% | 68.2% | 77.0% | 84.2% | 100.0% | 99.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% |
| Std. dev. | 0.1017 | 0.1224 | 0.0891 | 0.0896 | 0.0577 | 0.0686 | 0.0510 | 0.0509 | 0.0436 | 0.0538 | 0.0384 | 0.0384 |
| CO>SU (0.000) | | | | | | | | | | | | |
| Avg. diff. | -0.0034 | -0.0015 | 0.0811 | 0.0648 | -0.0104 | 0.0027 | 0.0820 | 0.0750 | -0.0079 | 0.0024 | 0.0818 | 0.0764 |
| False pos. | 4.9% | 4.8% | 12.7% | 9.3% | 4.7% | 4.9% | 26.2% | 23.0% | 5.0% | 5.1% | 41.4% | 36.3% |
| Std. dev. | 0.1338 | 0.1365 | 0.1006 | 0.1037 | 0.0772 | 0.0734 | 0.0606 | 0.0609 | 0.0606 | 0.0580 | 0.0474 | 0.0476 |
| EU>SU (0.157) | | | | | | | | | | | | |
| Avg. diff. | 0.0081 | -0.0074 | -0.0243 | -0.0178 | 0.0050 | -0.0066 | -0.0279 | -0.0257 | 0.0048 | -0.0022 | -0.0262 | -0.0246 |
| Power | 30.8% | 26.3% | 27.3% | 29.2% | 75.2% | 67.4% | 70.3% | 71.8% | 93.4% | 91.2% | 91.9% | 92.5% |
| Std. dev. | 0.1164 | 0.1122 | 0.0935 | 0.0955 | 0.0617 | 0.0627 | 0.0513 | 0.0513 | 0.0476 | 0.0479 | 0.0394 | 0.0394 |
| GT>SU (0.542) | | | | | | | | | | | | |
| Avg. diff. | 0.0042 | 0.0079 | -0.1375 | -0.1210 | 0.0071 | 0.0033 | -0.1369 | -0.1275 | 0.0070 | 0.0036 | -0.1359 | -0.1280 |
| Power | 100.0% | 96.6% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% |
| Std. dev. | 0.1110 | 0.1400 | 0.0801 | 0.0812 | 0.0644 | 0.0753 | 0.0481 | 0.0480 | 0.0502 | 0.0571 | 0.0372 | 0.0371 |
| AC>SU (0.254) | | | | | | | | | | | | |
| Avg. diff. | 0.0102 | -0.0105 | -0.0620 | -0.0433 | 0.0047 | -0.0103 | -0.0617 | -0.0546 | 0.0030 | -0.0113 | -0.0623 | -0.0571 |
| Power | 61.5% | 53.0% | 61.1% | 67.9% | 97.6% | 96.4% | 97.7% | 98.4% | 99.9% | 99.9% | 99.9% | 99.9% |
| Std. dev. | 0.1102 | 0.1231 | 0.0833 | 0.0853 | 0.0655 | 0.0675 | 0.0491 | 0.0496 | 0.0499 | 0.0520 | 0.0370 | 0.0374 |
| RMSE | 0.0075 | 0.0071 | 0.0714 | 0.0588 | 0.0066 | 0.0075 | 0.0700 | 0.0640 | 0.0047 | 0.0066 | 0.0700 | 0.0654 |

In terms of path coefficient estimation accuracy, assessed through average differences between estimated and true values, the performances of the PLSF and FIML methods were similar with both normal and non-normal data, across the three sample sizes. Both methods converged to the true values as sample sizes increased, with PLSF converging significantly faster. In this respect, the performances of PLSF and FIML were significantly better than OLS and PLS, mimicking the results with respect to the finite population.

Figure 4 highlights the performance in terms of statistical power for each of the methods. This figure reflects the fact that PLSF has greater power than FIML for all paths in all of the sample sizes considered. The focus here is on normal data; the results for the non-normal data display similar patterns. Six bar charts are shown. At the top of each chart the respective path is listed. Next to the vertical axes we show the power values achieved for each sample size. The sample sizes are shown underneath the horizontal axes.

**Figure 4.** Performance in terms of statistical power




In summary, in terms of statistical power, assessed through confidence intervals (Dijkstra & Henseler, 2015; Goodhue et al., 2012), PLSF and PLS presented similar performance, and generally better performance than FIML and OLS. In terms of avoidance of false positives, PLSF and FIML presented similar performance, and much better performance overall than OLS and PLS. With large samples ($N$=500) OLS and PLS performed particularly poorly with respect to avoidance of false positives.

## Discussion and conclusion

There has been a continuing and often antagonistic debate among proponents and detractors of classic PLS methods. This debate has frequently centered around one main problem with PLS methods, which is that they do not deal with factors, which are aggregations of indicators and measurement errors, but with composites. We made here what is arguably an important contribution to this debate by discussing the PLSF method, which combines elements of classic PLS methods and covariance-based SEM and provides estimates of the true composites and factors in a path model.

We showed evidence that PLSF is statistically consistent, like covariance-based SEM; but has greater statistical power, more in line with PLS. For example, for the path CO>AC (0.288), only

PLSF and PLS displayed power greater than 80% for a small sample size ($N$=100): respectively 85.9% and 86.3% with normal data, and 83.0% and 84.2% with non-normal data. For this same path and sample size, covariance-based SEM had a power of 72.4% with normal data, and 68.2% with non-normal data. Since between PLSF and PLS the only statistically consistent method is PLSF, as PLSF asymptotically converges to the true values and PLS does not, this suggests that PLSF is a good candidate in the context of SEM for the statistical property of asymptotic "efficiency" (Nikitin & Nikitin, 1995). A method is statistically efficient in an asymptotic sense if it is statistically consistent and also achieves a given level of power with the smallest sample size.

The PLSF method combines elements of PLS and covariance-based SEM methods. Like PLS it generates parameter estimates after it estimates factor scores, with the key difference that PLSF yields estimates of the true factors while PLS produces approximations. Also like PLS, it makes no data distribution assumptions, which is a characteristic of robust nonparametric methods (Siegel & Castellan, 1998). Like covariance-based SEM the PLSF method fits covariance matrices, with the key difference that PLSF fits factor covariance matrices while in covariance-based SEM the fitting involves indicator covariance matrices.

Since the PLSF method builds on the consistent PLS technique, it can be seen as an endorsement of the use of that technique as a basis for the development of factor-based path analysis methods. While PLSF is not a parameter correction approach like consistent PLS, the two approaches not being directly comparable in our view, it seems that overall PLSF presents greater power. Dijkstra & Henseler's (2015, p. 13) results suggest that with small samples consistent PLS has lower power than FIML, PLS and OLS: "… for small sample sizes, the statistical power decreased, and, in particular, PLSc [i.e., consistent PLS] had a somewhat lower statistical power than the other techniques". Given this, the PLSF method can also be seen not only as a major extension of consistent PLS, but one with improved performance.

Nevertheless, even if PLSF's performance was identical to that of consistent PLS with respect to path coefficients and loadings, it would still constitute an important methodological contribution. The reason for this is that in parameter correction techniques, such as consistent PLS, typically a different equation has to be developed to correct each parameter class; e.g., one equation to correct path coefficients, one equation to correct loadings etc. The PLSF method, on the other hand, estimates prototypical elements from which parameters are directly derived without any need for corrections. This places a large number of parameters in the hands of researchers (e.g., indicator weights and model-wide full collinearity VIFs), which can then be used in a variety of tests, including tests that currently do not exist because of limited access to parameter estimates.

While flexibility has not been directly addressed in our discussion, it is worth noting that the PLSF method is very flexible, arguably more so than PLS and related methods, allowing for many constraints to be imposed or relaxed. In this aspect it is similar to covariance-based SEM. For example, while in our analyses we assumed the common factor model property that indicator errors are uncorrelated, this assumption can be relaxed. To do this, we would use an appropriately modified version of the equation relating weights and loadings employed in the true composite estimation stage of PLSF. On the other hand, we could impose constraints by fixing parameters instead of relaxing assumptions. This could be done in any of the four stages.

The IS field is closely linked with the development, implementation, assessment, and use of PLS and related methods. Dijkstra & Henseler (2015) noted that IS research has not only made extensive use of PLS methods, but also has contributed greatly to related methodological

extensions and advances. We are in full agreement, and hope that the above discussion will contribute to this tradition. Those authors also state that (p. 17): “The advent of [consistent PLS] has substantial consequences for methodological research in [IS].” By building on consistent PLS, the PLSF method provides unequivocal evidence of this. After all, covariance-based SEM is often presented as a step beyond Wright’s (1934; 1960) path analysis method; because covariance-based SEM, unlike path analysis, deals with factors. However, while covariance-based SEM is a factor-based technique in a mathematical sense, since its underlying mathematics assumes the existence of factors, it does not directly estimate factors as part of its parameter estimation process. In covariance-based SEM factors are akin to “black holes” in that they indirectly and greatly influence the estimation of parameters, but are never directly “seen”. Arguably the PLSF method fills this gap; in it, SEM is truly an extension of Wright’s classic path analysis, with factors estimated directly and subsequently used to estimate model parameters.

## References


Adelman, I., & Lohmoller, J.-B. (1994). Institutions and development in the nineteenth century: A latent variable regression model. *Structural Change and Economic Dynamics*, 5(2), 329-359.

Bartlett, M.S. (1937). The statistical conception of mental factors. *British Journal of Psychology*, 28(1), 97-104.

Bartholomew, D.J., Deary, I.J., & Lawn, M. (2009). The origin of factor scores: Spearman, Thomson and Bartlett. *British Journal of Mathematical and Statistical Psychology*, 62(3), 569-582.

Bentler, P M., & Huang, W. (2014). On components, latent variables, PLS and simple methods: Reactions to Rigdon's rethinking of PLS. *Long Range Planning*, 47(3), 138-145.

Bera, A.K., & Jarque, C.M. (1981). Efficient tests for normality, homoscedasticity and serial independence of regression residuals: Monte Carlo evidence. *Economics Letters*, 7(4), 313-318.

Chin, W.W. (1998). Issues and opinion on structural equation modeling. *MIS Quarterly*, 22(1), vii-xvi.

Chin, W.W., Marcolin, B.L., & Newsted, P.R. (2003). A partial least squares latent variable modeling approach for measuring interaction effects: Results from a Monte Carlo simulation study and an electronic-mail emotion/adoption study. *Information Systems Research*, 14(2), 189-218.

Dijkstra, T.K., & Henseler, J. (2015). Consistent partial least squares path modeling. *MIS Quarterly*, 39(2), 297-316.

Dijkstra, T.K., & Schermelleh-Engel, K. (2014). Consistent partial least squares for nonlinear structural equation models. *Psychometrika*, 79(4), 585-604.

Dillon, W.R., & Goldstein, M. (1984). *Multivariate analysis: Methods and applications*. New York, NY: Wiley.

DiStefano, C., Zhu, M., & Mindrila, D. (2009). Understanding and using factor scores: Considerations for the applied researcher. *Practical Assessment, Research & Evaluation*, 14(20), 1-11.

Gel, Y.R., & Gastwirth, J.L. (2008). A robust modification of the Jarque–Bera test of normality. *Economics Letters*, 99(1), 30-32.

Goodhue, D., Lewis, W., & Thompson, R. (2007). Statistical power in analyzing interaction effects: Questioning the advantage of PLS with product indicators. *Information Systems Research*, 18(2), 211-227.

Goodhue, D.L., Lewis, W., and Thompson, R. (2012). Does PLS have advantages for small sample size or non-normal data? *MIS Quarterly*, 36(3), 981-1001.

Guo, K.H., Yuan, Y., Archer, N.P., & Connelly, C.E. (2011). Understanding nonmalicious security violations in the workplace: A composite behavior model. *Journal of Management Information Systems*, 28(2), 203-236.

Henseler, J., Dijkstra, T.K., Sarstedt, M., Ringle, C.M., Diamantopoulos, A., Straub, D.W., ... & Calantone, R.J. (2014). Common beliefs and reality about PLS – comments on Rönkkö and Evermann (2013). *Organizational Research Methods*, 17(2), 182-209.

Hershberger, S.L. (2005). Factor scores. In B.S. Everitt and D.C. Howell (Eds.) *Encyclopedia of Statistics in Behavioral Science*. (pp. 636-644). New York, NY: John Wiley.

Jarque, C.M., & Bera, A.K. (1980). Efficient tests for normality, homoscedasticity and serial independence of regression residuals. *Economics Letters*, 6(3), 255-259.
Kline, R.B. (2010). *Principles and practice of structural equation modeling*. New York, NY: The Guilford Press.
Kock, N. (2003). Communication-focused business process redesign: Assessing a communication flow optimization model through an action research study at a defense contractor. *IEEE Transactions on Professional Communication*, 46(1), 35-54.
Kock, N. (2007). *Systems analysis and design fundamentals: A business process redesign approach*. Thousand Oaks, CA: Sage Publications.
Kock, N. (2010). Using WarpPLS in e-collaboration studies: An overview of five main analysis steps. *International Journal of e-Collaboration*, 6(4), 1-11.
Kock, N. (2015). How likely is Simpson's paradox in path models? *International Journal of e-Collaboration*, 11(1), 1-7.
Kock, N. (2016). Non-normality propagation among latent variables and indicators in PLS-SEM simulations, *Journal of Modern Applied Statistical Methods*, 15(1), 299-315.
Kock, N., & Lynn, G.S. (2012). Lateral collinearity and misleading results in variance-based SEM: An illustration and recommendations. *Journal of the Association for Information Systems*, 13(7), 546-580.
Kock, N., & Mayfield, M. (2015). PLS-based SEM algorithms: The good neighbor assumption, collinearity, and nonlinearity. *Information Management and Business Review*, 7(2), 113-130.
Kock, N., & Murphy, F. (2001). *Redesigning acquisition processes: A new methodology based on the flow of knowledge and information*. Fort Belvoir, VA: Defense Acquisition University Press.
Kock, N., Danesh, A., & Komiak, P. (2008). A discussion and test of a communication flow optimization approach for business process redesign. *Knowledge and Process Management*, 15(1), 72-85.
Kock, N., Verville, J., Danesh-Pajou, A., & DeLuca, D. (2009). Communication flow orientation in business process modeling and its effect on redesign success: Results from a field study. *Decision Support Systems*, 46(2), 562-575.
Lohmöller, J.-B. (1989). *Latent variable path modeling with partial least squares*. Heidelberg, Germany: Physica-Verlag.
McIntosh, C. N., Edwards, J. R., & Antonakis, J. (2014). Reflections on partial least squares path modeling. *Organizational Research Methods*, 7(2), 210-251.
Mendling, J., Strembeck, M., & Recker, J. (2012). Factors of process model comprehension—Findings from a series of experiments. *Decision Support Systems*, 53(1), 195-206.
Mueller, R.O. (1996). *Basic principles of structural equation modeling*. New York, NY: Springer.
Moqbel, M., Nevo, S., & Kock, N. (2013). Organizational members' use of social networking sites and job performance: An exploratory study. *Information Technology & People*, 26(3), 240-264.
Nikitin, I.I., & Nikitin, Y. (1995). *Asymptotic efficiency of nonparametric tests*. Cambridge, England: Cambridge University Press.
Nunnaly, J.C. (1978). *Psychometric Theory*. New York, NY: McGraw Hill.
Nunnally, J.C., & Bernstein, I.H. (1994). *Psychometric theory*. New York, NY: McGraw-Hill.

Paxton, P., Curran, P.J., Bollen, K.A., Kirby, J., & Chen, F. (2001). Monte Carlo experiments: Design and implementation. *Structural Equation Modeling*, 8(2), 287-312.
Peterson, R.A., & Yeolib, K. (2013). On the relationship between coefficient alpha and composite reliability. *Journal of Applied Psychology*, 98(1), 194-198.
Robert, C.P., & Casella, G. (2005). *Monte Carlo statistical methods*. New York, NY: Springer.
Rönkkö, M. (2014). The effects of chance correlations on partial least squares path modeling. *Organizational Research Methods*, 17(2), 164-181.
Rönkkö, M., & Evermann, J. (2013). A critical examination of common beliefs about partial least squares path modeling. *Organizational Research Methods*, 16(3), 425-448.
Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software*, 48(2), 1-36.
Siegel, S., & Castellan, N.J. (1998). *Nonparametric statistics for the behavioral sciences*. Boston, MA: McGraw-Hill.
Thurstone, L.L. (1935). *The vectors of mind*. Chicago, IL: University of Chicago Press.
Wold, H. (1980). Model construction and evaluation when theoretical knowledge is scarce. In J. Kmenta and J. B. Ramsey (Eds.), *Evaluation of econometric models* (pp. 47-74). Waltham, MA: Academic Press.
Wright, S. (1934). The method of path coefficients. *The Annals of Mathematical Statistics*, 5(3), 161-215.
Wright, S. (1960). Path coefficients and path regressions: Alternative or complementary concepts? *Biometrics*, 16(2), 189-202.

## Appendix A: Derivations of equations

Equations (A.1), (A.2) and (A.3) below provide the foundation of $\mathcal{F}_2$, the true composite estimation function. In these equations $x_i$ is a matrix where each column refers to one of the indicators associated with composite $\hat{C}_i$ (and thus with factor $\hat{F}_i$); $\hat{\lambda}_i{}'$ is the transpose of $\hat{\lambda}_i$, the column vector storing the loadings associated with the indicators; $\hat{\theta}_i$ is the matrix of indicator error terms; $\hat{\omega}_{iC}$ is the composite weight; $\hat{\omega}_{i\varepsilon}$ is the measurement error weight; $\hat{\omega}_i$ is the column vector of indicator weights; the superscript $-1$ denotes the classic matrix inversion; and the superscript $+$ denotes the Moore–Penrose pseudoinverse transformation..

$$\hat{F}_i = Stdz\left(\hat{C}_i\hat{\omega}_{iC} + \hat{\varepsilon}_i\hat{\omega}_{i\varepsilon}\right). \quad (A.1)$$

$$\hat{\theta}_i = x_i - \hat{F}_i\hat{\lambda}_i{}'. \quad (A.2)$$

$$\hat{\omega}_i = \Sigma_{x_ix_i}{}^{-1}\left(\Sigma_{x_ix_i} - diag\left(\Sigma_{x_i\hat{\theta}_i}\right)\right)\hat{\lambda}_i{}'^{+}. \quad (A.3)$$

**Derivation of (A.1).** From our previous discussion on composites and factors, we know that
$F_i = x_i\omega_i + \varepsilon_i\omega_{i\varepsilon}$, $x_i\omega_i = C_i\omega_{iC}$.
Thus it follows that
$F_i = C_i\omega_{iC} + \varepsilon_i\omega_{i\varepsilon}$,
where $F_i$ is expected to be standardized.

**Derivation of (A.2).** From our previous discussion on composites and factors, we know that
$x_i = F_i\lambda_i{}' + \theta_i$.
Thus it follows that
$\theta_i = x_i - F_i\lambda_i{}'$.

**Derivation of (A.3).** From our previous discussion on composites and factors, we know that
$x_i = F_i\lambda_i{}' + \theta_i$, $F_i = x_i\omega_i + \varepsilon_i\omega_{i\varepsilon}$.
Combining these two equations we obtain
$x_i = (x_i\omega_i + \varepsilon_i\omega_{i\varepsilon})\lambda_i{}' + \theta_i \rightarrow$
$x_i = x_i\omega_i\lambda_i{}' + \varepsilon_i\omega_{i\varepsilon}\lambda_i{}' + \theta_i$.
Applying covariance properties to the above we obtain
$\Sigma_{x_ix_i} = \Sigma_{x_ix_i}\omega_i\lambda_i{}' + \Sigma_{x_i\varepsilon_i}\omega_{i\varepsilon}\lambda_i{}' + \Sigma_{x_i\theta_i} \rightarrow$
$\Sigma_{x_ix_i} = \Sigma_{x_ix_i}\omega_i\lambda_i{}' + diag\left(\Sigma_{x_i\theta_i}\right) \rightarrow$
$\Sigma_{x_ix_i}\omega_i\lambda_i{}' = \Sigma_{x_ix_i} - diag\left(\Sigma_{x_i\theta_i}\right) \rightarrow$
$\omega_i\lambda_i{}' = \Sigma_{x_ix_i}{}^{-1}\left(\Sigma_{x_ix_i} - diag\left(\Sigma_{x_i\theta_i}\right)\right)$,
where the superscript $-1$ denotes the classic matrix inversion.

To isolate $\omega_i$ in the equation above we need to use the Moore–Penrose pseudoinverse transformation, since the classic matrix inversion transformation cannot be applied to a vector. Doing this, we obtain

$$\omega_i = \Sigma_{x_ix_i}{}^{-1}\left(\Sigma_{x_ix_i} - diag\left(\Sigma_{x_i\theta_i}\right)\right)\lambda_i{}'^{+},$$

where the superscript $+$ denotes the Moore–Penrose pseudoinverse transformation.

Equations (A.4), (A.5) and (A.6) below provide the foundation of $\mathcal{F}_3$, the true factor estimation function. Each of these equations includes a variable being updated with a "mix" of itself and other variables, which causes that variable to receive or lose variation that resides in those other variables. Whether variation is gained or lost depends on the sign of the multiplier attached to those other variables. For example, let us consider the simple assignment $Y = Stdz(Y + aX)$, where both $Y$ and $X$ are standardized. The variable $Y$ gains variation from the variable $X$ if $a > 0$, and loses variation from $X$ if $a < 0$. In either case the amount of variation gained or lost is reflected in the correlation between $Y$ and $X$, which itself is a function of $a$. The more variation $Y$ and $X$ share, the greater is the correlation between them. If $a = 0$ no variation is gained or lost.

$$\hat{\varepsilon}_i = Stdz\left(\hat{\varepsilon}_i + \left(\hat{\Sigma}_{F_iF_j} - \Sigma_{\hat{F}_i\hat{F}_j}\right)\frac{\hat{\Sigma}_{F_iF_j}}{\widehat{\omega}_{i\varepsilon}}\left(\hat{C}_j\widehat{\omega}_{jC} + \hat{\varepsilon}_j\widehat{\omega}_{j\varepsilon}\right)\right). \tag{A.4}$$

$$\hat{F}_i = Stdz\left(\hat{F}_i + \left(\widehat{\omega}_{iC} - \Sigma_{\hat{F}_i\hat{C}_i}\right)\hat{C}_i\widehat{\omega}_{iC}\right). \tag{A.5}$$

$$\hat{\varepsilon}_i = Stdz\left(\hat{\varepsilon}_i - \Sigma_{\hat{C}_i\hat{\varepsilon}_i}\hat{C}_i\widehat{\omega}_{iC} + \left(\widehat{\omega}_{i\varepsilon} - \Sigma_{\hat{F}_i\hat{\varepsilon}_i}\right)\hat{F}_i\widehat{\omega}_{i\varepsilon}\right). \tag{A.6}$$

**Derivation of (A.4).** From our previous discussion on composites and factors, we know that for each pair of correlated factors $F_i$ and $F_j$ we have

$F_i = C_i\omega_{iC} + \varepsilon_i\omega_{i\varepsilon}$, $F_j = C_j\omega_{jC} + \varepsilon_j\omega_{j\varepsilon}$, $F_i = \Sigma_{F_iF_j}F_j + \delta_{ij}$,

where $\delta_{ij}$ is an error term that accounts the variance in $F_i$ that is not explained by $F_j$.

Combining these equations we have

$C_i\frac{\omega_{iC}}{\omega_{i\varepsilon}} + \varepsilon_i = \frac{1}{\omega_{i\varepsilon}}\Sigma_{F_iF_j}\left(C_j\omega_{jC} + \varepsilon_j\omega_{j\varepsilon}\right) + \frac{\delta_{ij}}{\omega_{i\varepsilon}}$.

We can see that $\varepsilon_i$ shares variation with $C_j$ and $\varepsilon_j$ according to

$\frac{\Sigma_{F_iF_j}}{\omega_{i\varepsilon}}\left(C_j\omega_{jC} + \varepsilon_j\omega_{j\varepsilon}\right)$.

Thus, in order to make $\Sigma_{\hat{F}_i\hat{F}_j} = \hat{\Sigma}_{F_iF_j}$ we iteratively assign

$\hat{\varepsilon}_i = Stdz\left(\hat{\varepsilon}_i + \left(\hat{\Sigma}_{F_iF_j} - \Sigma_{\hat{F}_i\hat{F}_j}\right)\frac{\hat{\Sigma}_{F_iF_j}}{\widehat{\omega}_{i\varepsilon}}\left(\hat{C}_j\widehat{\omega}_{jC} + \hat{\varepsilon}_j\widehat{\omega}_{j\varepsilon}\right)\right)$.

Note that as $\hat{\varepsilon}_i$ changes so does $\hat{F}_i$ because $F_i = C_i\omega_{iC} + \varepsilon_i\omega_{i\varepsilon}$, and also that this assignment is only made if $\widehat{\omega}_{i\varepsilon} > 0$.

**Derivation of (A.5).** As noted above

$F_i = C_i\omega_{iC} + \varepsilon_i\omega_{i\varepsilon}$.

We can see that $F_i$ shares variation with $C_i$ according to

$C_i\omega_{iC}$.

Thus, in order to make $\Sigma_{\hat{F}_i\hat{C}_i} = \widehat{\omega}_{iC}$ we iteratively assign

$\hat{F}_i = Stdz\left(\hat{F}_i + \left(\widehat{\omega}_{iC} - \Sigma_{\hat{F}_i\hat{C}_i}\right)\hat{C}_i\widehat{\omega}_{iC}\right)$.

**Derivation of (A.6).** From our previous discussion on composites and factors, we know that

$\varepsilon_i \perp C_i$, $\varepsilon_i = F_i\omega_{i\varepsilon} + \theta_{i\varepsilon}$.

where $\perp$ means "orthogonal to", and $\theta_{i\varepsilon}$ is an error term that accounts for the variation in $\varepsilon_i$ that is not explained by $F_i$. Note that $\omega_{i\varepsilon} = \lambda_{i\varepsilon}$.

We can see that $\varepsilon_i$ shares no variation with $C_i$, and also that $\varepsilon_i$ shares variation with $F_i$ according to

$F_i\omega_{i\varepsilon}$.

Thus, in order to make $\Sigma_{\hat{C}_i\hat{\varepsilon}_i} = 0$ and $\Sigma_{\hat{F}_i\hat{\varepsilon}_i} = \widehat{\omega}_{i\varepsilon}$ we iteratively assign

$\hat{\varepsilon}_i = Stdz\left(\hat{\varepsilon}_i - \Sigma_{\hat{C}_i\hat{\varepsilon}_i}\hat{C}_i\widehat{\omega}_{iC} + \left(\widehat{\omega}_{i\varepsilon} - \Sigma_{\hat{F}_i\hat{\varepsilon}_i}\right)\hat{F}_i\widehat{\omega}_{i\varepsilon}\right)$.

Note that the multiplier $-\Sigma_{\hat{C}_i\hat{\varepsilon}_i}$ is derived from $0 - \Sigma_{\hat{C}_i\hat{\varepsilon}_i}$.

# Appendix B: Algorithmic formulation of PLSF

In this appendix we provide all of the equations that make up the PLSF method, for each of the four functions, as well as the algorithmic sequence of their execution and related explanatory notes. We do this with the goal of facilitating the implementation of the method in any numeric computing environment, including open source environments such as R and GNU Octave.

## Function $\mathcal{F}_1$: The consistent PLS function

In the steps below, $i = 1 \ldots N_C$, and $j = 1 \ldots n_i$, where $N_C$ is the number of composites in the model (the same as the number of factors), and $n_i$ is the number of indicators associated with each composite $C_i$. The steps 1.1 to 1.7 implement the basic design of PLS Mode A, also known as PLS Mode A employing the centroid scheme.

**Step 1.1.** Initialize each indicator weight $\widehat{\omega}_{ij}$ with 1.

**Step 1.2.** Store each indicator weight in $\overline{\overline{\omega}}_{ij}$ for later comparison.

**Step 1.3.** Estimate each composite $\hat{C}_i$ as

$$\hat{C}_i = Stdz\left(\sum_{j=1}^{n_i} \widehat{\omega}_{ij}\, x_{ij}\right).$$

**Step 1.4.** Set each inner weight $\hat{v}_{ij}$ as

$$\hat{v}_{ij} = Sign\left(\Sigma_{\hat{C}_i \hat{C}_j}\right).$$

Here the inner weights are set as the signs ($-1$ or $+1$) of the estimated correlations among "neighbor" composites. Neighbor composites are those that are linked to a composite by arrows, either by pointing at or being pointed at by the composite.

**Step 1.5.** Estimate each composite $\hat{C}_i$ as

$$\hat{C}_i = Stdz\left(\sum_{j=1}^{A_i} \hat{v}_{ij}\, \hat{C}_j\right),$$

where $Stdz(\cdot)$ is the standardization function; and $A_i$ is the number of composites $\widehat{C}_j$ ($j = 1 \ldots A_i$) that are neighbors of the composite $\widehat{C}_i$.

**Step 1.6.** Solve for each indicator weight $\widehat{\omega}_{ij}$ the equation

$$x_{ij} = \hat{C}_i \widehat{\omega}_{ij} + \hat{\epsilon}_{ij},$$

where $\hat{\epsilon}_{ij}$ is an error term that accounts for the variation in $x_{ij}$ that is not explained by $\hat{C}_i$.

**Step 1.7.** Go back to Step 1.2 if any indicator weight $\widehat{\omega}_{ij}$ differs from the previously stored estimate $\overline{\overline{\omega}}_{ij}$ by more than a small fraction.

**Step 1.8.** Estimate each true reliability $\hat{\rho}_i$ and loading vector $\hat{\lambda}_i$ as

$$\hat{\rho}_i = (\widehat{\omega}_i{}'\widehat{\omega}_i)^2 \left(\widehat{\omega}_i{}'\left(\Sigma_{x_i x_i} - diag(\Sigma_{x_i x_i})\right)\widehat{\omega}_i\right)/(\widehat{\omega}_i{}'(\widehat{\omega}_i\widehat{\omega}_i{}' - diag(\widehat{\omega}_i\widehat{\omega}_i{}'))\widehat{\omega}_i),$$

$$\hat{\lambda}_i = \left(\widehat{\omega}_i\sqrt{\hat{\rho}_i}\right)/(\widehat{\omega}_i{}'\widehat{\omega}_i),$$

where the superscript $'$ denotes the transpose operation, $\Sigma_{x_i x_i}$ is the covariance matrix of the indicators associated with composite $C_i$, and the function $diag(\cdot)$ returns the diagonal of a matrix.

### Function $\mathcal{F}_2$: The true composite estimation function

In the steps below, $i = 1 \ldots N_C$, where $N_C$ is the number of composites (the same as the number of factors) in the model. From the previous function come estimates of reliabilities, indicator loadings, measurement errors, composite weights, and measurement error weights. Also from the previous function come initial estimates of composites and indicator weights.

**Step 2.1.** Set each measurement error $\hat{\varepsilon}_i$, composite weight $\omega_{iC}$, and measurement error weight $\omega_{i\varepsilon}$ as

$$\hat{\varepsilon}_i = Stdz\big(Rnd(N)\big),$$
$$\omega_{iC} = \sqrt{\rho_i},$$
$$\omega_{i\varepsilon} = \sqrt{1 - \rho_i},$$

where $Rnd(N)$ is a function that returns an independent and identically distributed (i.i.d.) variable with $N$ rows, with $N$ being the sample size. In software implementations the random seed may be set to a fixed value prior to setting $\hat{\varepsilon}_i$ in order to avoid different results each time an analysis is conducted with the same model and empirical data.

**Step 2.2.** Store all weight vectors $\widehat{\omega}_i$ in $\overline{\overline{\omega}}_i$ for later comparison.

**Step 2.3.** Set each factor $\hat{F}_i$ as

$$\hat{F}_i = Stdz\big(\hat{C}_i \widehat{\omega}_{iC} + \hat{\varepsilon}_i \widehat{\omega}_{i\varepsilon}\big).$$

**Step 2.4.** Set each indicator error matrix $\hat{\theta}_i$ as

$$\hat{\theta}_i = x_i - \hat{F}_i \hat{\lambda}_i{}',$$

where $x_i$ is the matrix of indicators associated with factor $\hat{F}_i$, $\hat{\lambda}_i$ is the vector of loadings associated with the factor, and the $'$ superscript indicates the transpose operation.

**Step 2.5.** Estimate each weight vector $\widehat{\omega}_i$ as

$$\widehat{\omega}_i = \Sigma_{x_i x_i}{}^{-1} \left(\Sigma_{x_i x_i} - diag\big(\Sigma_{x_i \hat{\theta}_i}\big)\right) \hat{\lambda}_i{}'^{+},$$

where $\Sigma_{x_i x_i}$ is the covariance matrix of the indicators associated with factor $\hat{F}_i$, $\Sigma_{x_i \hat{\theta}_i}$ is the matrix of covariances among the indicators and their errors, $diag(\cdot)$ is a function that returns the diagonal version of a matrix, and the superscript $+$ denotes the Moore–Penrose pseudoinverse transformation.

**Step 2.6.** Estimate each composite $\hat{C}_i$ as

$$\hat{C}_i = \frac{1}{\widehat{\omega}_{iC}} (x_i \widehat{\omega}_i) .$$

**Step 2.7.** Go back to Step 2.2 if any element of any of the weight vectors $\widehat{\omega}_i$ differs from the previously stored estimates in $\overline{\overline{\omega}}_i$ by more than a small fraction.

### Function $\mathcal{F}_3$: The true factor estimation function

In the steps below $i, j = 1 \ldots N_F$. Here $N_F$ is the number of factors in the model. Each combination $(i, j)$ refers to a pair of correlated elements in the model; factors, composites, or measurement errors. From the previous function come estimates of composites, indicator weights, composite weights, and measurement error weights. Also come from the previous function initial estimates of measurement errors.

**Step 3.1.** Initialize each factor $\hat{F}_i$ as

$$\hat{F}_i = Stdz\left(\hat{C}_i\hat{\omega}_{iC} + \hat{\varepsilon}_i\hat{\omega}_{i\varepsilon}\right).$$

**Step 3.2.** Set each element of the estimated matrix of correlations among factors $\hat{\Sigma}_{F_iF_j}$ as

$$\hat{\Sigma}_{F_iF_j} = \frac{\Sigma_{\hat{C}_i\hat{C}_j}}{\sqrt{\hat{\rho}_i\hat{\rho}_j}},$$

where $\Sigma_{\hat{C}_i\hat{C}_j}$ is the corresponding element of the matrix of correlations among estimated composites.

**Step 3.3.** Calculate the matrix of correlations among estimated factors $\Sigma_{\hat{F}\hat{F}}$ and store it in $\bar{\bar{\Sigma}}_{\hat{F}\hat{F}}$ for later comparison. Note that this is not the same as the estimated matrix of correlations among factors $\hat{\Sigma}_{FF}$, which is fixed after Step 3.2.

**Step 3.4.** Add or remove variation in each measurement error $\hat{\varepsilon}_i$, if $\hat{\omega}_{i\varepsilon} > 0$, by making

$$\hat{\varepsilon}_i = Stdz\left(\hat{\varepsilon}_i + \left(\hat{\Sigma}_{F_iF_j} - \Sigma_{\hat{F}_i\hat{F}_j}\right)\frac{\hat{\Sigma}_{F_iF_j}}{\hat{\omega}_{i\varepsilon}}\left(\hat{C}_j\hat{\omega}_{jC} + \hat{\varepsilon}_j\hat{\omega}_{j\varepsilon}\right)\right),$$

where $\Sigma_{\hat{F}_i\hat{F}_j}$ is the correlation among each pair of estimated factors.

**Step 3.5.** Add or remove variation in each factor $\hat{F}_i$, if $\hat{\omega}_{i\varepsilon} > 0$, by making

$$\hat{F}_i = Stdz\left(\hat{F}_i + \left(\hat{\omega}_{iC} - \Sigma_{\hat{F}_i\hat{C}_i}\right)\hat{C}_i\hat{\omega}_{iC}\right),$$

where $\Sigma_{\hat{F}_i\hat{C}_i}$ is the correlation among an estimated factor and its true composite.

**Step 3.6.** Add or remove variation in each measurement error $\hat{\varepsilon}_i$, if $\hat{\omega}_{i\varepsilon} > 0$, by making

$$\hat{\varepsilon}_i = Stdz\left(\hat{\varepsilon}_i - \Sigma_{\hat{C}_i\hat{\varepsilon}_i}\hat{C}_i\hat{\omega}_{iC} + \left(\hat{\omega}_{i\varepsilon} - \Sigma_{\hat{F}_i\hat{\varepsilon}_i}\right)\hat{F}_i\hat{\omega}_{i\varepsilon}\right),$$

where $\Sigma_{\hat{C}_i\hat{\varepsilon}_i}$ is the correlation between an estimated composite and its corresponding measurement error, and $\Sigma_{\hat{F}_i\hat{\varepsilon}_i}$ is the correlation between an estimated factor and its measurement error.

**Step 3.7.** Estimate each factor $\hat{F}_i$ as

$$\hat{F}_i = Stdz\left(\hat{C}_i\hat{\omega}_{iC} + \hat{\varepsilon}_i\hat{\omega}_{i\varepsilon}\right).$$

**Step 3.8.** Estimate each measurement error $\hat{\varepsilon}_i$ as

$$\hat{\varepsilon}_i = Stdz\left(\frac{1}{\hat{\omega}_{i\varepsilon}}\left(\hat{F}_i - \hat{C}_i\hat{\omega}_{iC}\right)\right).$$

**Step 3.9.** Go back to Step 3.3 if the absolute sum of the differences in $\Sigma_{\hat{F}\hat{F}} - \hat{\Sigma}_{FF}$ and in $\Sigma_{\hat{F}\hat{F}} - \bar{\bar{\Sigma}}_{\hat{F}\hat{F}}$ both fall above a small fraction.

### Function $\mathcal{F}_4$: The full parameter estimation function

In the steps below, $i = 1 \ldots N_F$, where $N_F$ is the number of factors in the model. From the previous function come estimates of factors, measurement errors, composite weights, and measurement error weights.

**Step 4.1.** Update each composite $\hat{C}_i$ as

$$\hat{C}_i = Stdz\left(\frac{1}{\hat{\omega}_{iC}}\left(\hat{F}_i - \hat{\varepsilon}_i\hat{\omega}_{i\varepsilon}\right)\right).$$

**Step 4.2.** Update each weight vector $\hat{\omega}_i$ as

$$\hat{\omega}_i = x_i{}^{+}\hat{C}_i\hat{\omega}_{iC}.$$

**Step 4.3.** Update each loading vector $\hat{\lambda}_i$ as

$$\hat{\lambda}_i = x_i{}'\hat{F}_i{}'^{+}.$$

**Step. 4.4.** Estimate each indicator residual $\hat{\theta}_i$ as

$$\hat{\theta}_i = x_i - \hat{F}_i\hat{\lambda}_i{}'.$$

**Step 4.5.** Solve for each path coefficient $\hat{\beta}_{ij}$ the equation involving an endogenous factor

$$\hat{F}_i = \sum_{j=1}^{N_i}\hat{\beta}_{ij}\,\hat{F}_j + \hat{\beta}_i\hat{I}_i + \hat{\zeta}_i,$$

where $\hat{\zeta}_i$ is the residual associated with the endogenous factor $\hat{F}_i$, and $\hat{F}_j$ ($j = 1 \ldots N_i$) are the factors that point at the endogenous factor. The instrumental variables $\hat{I}_i$ implement a two-stage least squares estimation; they exist for all endogenous factors in the model that contain variation from other factors but are not directly linked with those factors.

**Step 4.6.** Estimate each endogenous factor residual $\hat{\zeta}_i$ as

$$\hat{\zeta}_i = \hat{F}_i - \sum_{j=1}^{N_i}\hat{\beta}_{ij}\,\hat{F}_j - \hat{\beta}_i\hat{I}_i.$$

At the end of the above steps for the four functions, which implement the four stages that make up the PLSF method, we are left with estimates of the true values of the following: $\hat{F}$, $\hat{C}$, $\hat{\varepsilon}$, $\hat{\zeta}$, $\hat{\omega}$, $\hat{\omega}_C$, $\hat{\omega}_\varepsilon$, $\hat{\lambda}$, $\hat{\beta}$ and $\hat{\theta}$. To the best of our knowledge, no other SEM method provides such an extensive set of estimates. Given this, it is reasonable to expect that these estimates could serve as the basis for the development of a number of new tests that are not currently possible.