STRUCTURAL EQUATION MODELING, *8*(2), 287–312

TEACHER'S CORNER

Monte Carlo Experiments: Design and Implementation

Pamela Paxton
Department of Sociology
The Ohio State University

Patrick J. Curran
L. L. Thurstone Quantitative Laboratory
Department of Psychology
University of North Carolina at Chapel Hill

Kenneth A. Bollen
Carolina Population Center
Department of Sociology
The University of North Carolina at Chapel Hill

Jim Kirby
Agency for Health Care Policy and Research
Rockville, Maryland

Feinian Chen
Carolina Population Center
Department of Sociology
The University of North Carolina at Chapel Hill

The use of Monte Carlo simulations for the empirical assessment of statistical estimators is becoming more common in structural equation modeling research. Yet, there is little guidance for the researcher interested in using the technique. In this article we il-

Requests for reprints should be sent to Pamela Paxton, Department of Sociology, 300 Bricker Hall, 190 North Oval Mall, The Ohio State University, Columbus, OH 43210–1353. E-mail: paxton.36@osu.edu

lustrate both the design and implementation of Monte Carlo simulations. We present 9 steps in planning and performing a Monte Carlo analysis: (1) developing a theoretically derived research question of interest, (2) creating a valid model, (3) designing specific experimental conditions, (4) choosing values of population parameters, (5) choosing an appropriate software package, (6) executing the simulations, (7) file storage, (8) troubleshooting and verification, and (9) summarizing results. Throughout the article, we use as a running example a Monte Carlo simulation that we performed to illustrate many of the relevant points with concrete information and detail.

Monte Carlo simulations have become common in evaluating statistical estimators for structural equation models. Although analytical statistical theory can address some research questions, finite sample properties of structural equation model estimators are often beyond the reach of the established asymptotic theory. In other cases the distributions are not known even asymptotically (e.g., many fit indexes). In such situations, Monte Carlo simulations provide an excellent method for evaluating estimators and goodness-of-fit statistics under a variety of conditions, including sample size, nonnormality, dichotomous or ordinal variables, model complexity, and model misspecification. Examples of Monte Carlo studies in structural equation modeling (SEM) include Anderson and Gerbing's (1984) examination of fit indexes, nonconvergence, and improper solutions; Curran, West, and Finch's (1996) study of likelihood ratio test statistics; Hu and Bentler's (1999) analysis of cutoff criteria for goodness-of-fit statistics; and Muthén and Kaplan's (1985, 1992) study of the effects of coarse categorization in structural equation model estimation (see Gerbing & Anderson, 1993, for a review of Monte Carlo studies on goodness-of-fit statistics). Despite the rapid growth of these techniques, many topics in SEM would benefit from an empirical analysis through Monte Carlo methods.

Designing a Monte Carlo simulation is not an easy task, however. Although there are a few books on the technique (e.g., Mooney, 1997; Rubinstein, 1981; Smith, 1973), none directly relates the method to structural equation models. With the numerous factors to consider in a Monte Carlo simulation, there is a great deal to be learned from experience. The purpose of this article is to provide that experience through an introduction to the design and implementation of a Monte Carlo simulation in the area of SEM. We will lead the reader through the steps of a simulation, provide suggestions on planning and execution at each step, and outline potential pitfalls in execution. At every stage we stress three interconnected goals: theory, relevance, and practicality.

We present nine steps in planning and performing a structural equation Monte Carlo analysis: (1) developing a theoretically derived research question, (2) creating a valid model, (3) designing specific experimental conditions, (4) choosing values of population parameters, (5) choosing an appropriate software package, (6) executing the simulations, (7) file storage, (8) troubleshooting and verification, and (9) summarizing results. Although we present the design of

Monte Carlo models as a step-by-step process, the steps are actually interconnected. For example, choosing a statistical package for estimation (Step 5) can influence the experimental conditions (Step 3). Ultimately the process of design is more simultaneous than we are able to portray here.

After a brief introduction to and justification for Monte Carlo simulations in general, we discuss each step in detail. Throughout the article, we use a running example of a Monte Carlo simulation that we performed. The running example illustrates many of the relevant points with concrete information and detail.

A BRIEF INTRODUCTION TO MONTE CARLO SIMULATIONS

In the Monte Carlo method "properties of the distributions of random variables are investigated by use of simulated random numbers" (Gentle, 1985, p. 612). Typically, the asymptotic properties of an estimator are known, but its finite sampling properties are not. Monte Carlo simulations allow researchers to assess the finite sampling performance of estimators by creating controlled conditions from which sampling distributions of parameter estimates are produced. Knowledge of the sampling distribution is the key to evaluation of the behavior of a statistic. For example, a researcher can determine the bias of a statistic from the sampling distribution, as well as its efficiency and other desirable properties. Sampling distributions are theoretical and unobserved, however, so with the Monte Carlo method a researcher artificially creates the sampling distribution.

The researcher begins by creating a model with known population parameters (i.e., the values are set by the researcher). The analyst then draws repeated samples of size N from that population and, for each sample, estimates the parameters of interest. Next, a sampling distribution is estimated for each population parameter by collecting the parameter estimates from all the samples. The properties of that sampling distribution, such as its mean or variance, come from this estimated sampling distribution.

The Monte Carlo method is thus an *empirical* method for evaluating statistics. Through computational "brute force," a researcher creates sampling distributions of relevant statistics. Suppose that we have a new consistent estimator of coefficients in a structural equation model: We want to assess bias in the estimator in small and moderate sample sizes. To do so, we create a structural equation model with known coefficients and distributions for the observed variables. Then we draw, say, 500 samples of size 50 from that known population. For each sample, we would use our new estimator and obtain the values of the coefficients. All of the coefficient estimates (500 for each parameter) would then be put into a distribution and the mean of that sampling distribution calculated. Comparing the mean of the

coefficient estimates to the population value of the coefficient would help us to assess the bias of the estimator.

Monte Carlo simulations are appropriate for questions that we cannot evaluate with asymptotic theory. Statistical theory is superior because it often covers broader classes of models than can a Monte Carlo experiment. However, in SEM as well as other areas, the statistical properties rely on unrealistic conditions such as the availability of a large sample or ideal distributional assumptions for variables. It is in these instances that Monte Carlo methods step in to fill the gap and augment analytical results. For example, a practitioner working with a moderately sized structural equation model (say, 1,000 cases) and normality is less in need of information from Monte Carlo simulations—asymptotic theory provides the relevant information about the sampling distribution of the estimator. But if a practitioner is working with 100 cases and variables from a distribution with high kurtosis, Monte Carlo simulations may be the only way to determine the properties of the sampling distribution of an estimator. Monte Carlo methods are set up as an experiment, where we gather data to test specific theoretically derived hypotheses. For introductions to the method, see Mooney (1997), Rubinstein (1981), or Smith (1973). A basic brief exposition of the technique is available in Kennedy (1992, chapter 2).

NINE STEPS IN DESIGNING AND PERFORMING A MONTE CARLO ANALYSIS

Step 1: Developing a Theoretically Derived Research Question of Interest

The validity and utility of a simulation study is only as strong as the quality of the questions being assessed. One of the key criticisms of Monte Carlo studies is the lack of strong theory guiding the design and analysis of the simulation. Without strong theory, simulation studies are often thought to be akin to randomly looking for a needle in a haystack. It is thus imperative that the research questions of interest be strongly tied to statistical theory and that the simulation serve as a method to collect data to empirically evaluate the proposed hypotheses. Because Monte Carlo simulations can be huge undertakings, with multiple conditions and massive amounts of resultant data, a fortunate by-product of the careful identification and selection of research hypotheses is that the scope of the simulation study can be more focused and manageable.

Outlining specific, theory-based questions at the outset of the project is one of the best ways to increase manageability and scientific relevance. For example, our Monte Carlo project initially had three main research interests: an examination of goodness-of-fit statistics under varying degrees of misspecification, an investiga-

tion of a global goodness-of-fit statistic, and a comparison of the maximum likelihood (ML) estimator to the newly developed two-stage least squares (2SLS) estimator (Bollen, 1996). Outlining our research questions in the beginning made it clear that our Monte Carlo design would need to include misspecifications (where the estimated model does not perfectly correspond to the population model). We also knew from the outset that we would need two estimation methods. More important, our initial research hypotheses did not relate to nonnormal distributions, which subsequently allowed us to reduce the number of experimental conditions with consideration of only multivariate normal distributions.

As another example, Anderson and Gerbing (1984) were interested in the effects of sampling error on test statistics, nonconvergence, and improper solutions. With their research question in mind, they knew they needed to vary sample size, the size of the model, and the population parameters (all factors that could affect sampling error). Because they were only interested in sampling error, however, they did not need to include misspecification as an experimental condition. Ultimately, the goal is to construct an optimal match between the research question and the experimental design. Outlining specific research questions early in the project aids in that task.

Step 2: Creating Representative Models

A second major criticism of Monte Carlo simulation studies is a lack of external validity. Often only a small number of model types are examined, or the models that are tested bear little resemblance to those commonly estimated in applied research. A key step in designing a Monte Carlo experiment is therefore to create a model that is representative from an applied standpoint.

To address this issue, the Monte Carlo researcher should review structural equation model applications across a large number of journals in several areas of research to which they would like to generalize the subsequent results. In our literature review, we focused on structural equation model applications published in key sociological and psychological journals over the previous 5 years. Based on such a review, the researcher can make an informed, subjective judgment about the general types of structural equation models common in applied research. Typically, the goal of maximizing external validity will be parallel to the goal of optimally testing a proposed research hypothesis. However, there may be situations in which the research hypothesis demands a particular model and external validity is less important.

There are several specific questions to consider in the construction of a model. First, what will be the overall structure of the model—a confirmatory factor analysis (CFA) or a full structural equation model? CFAs are typical in simulation designs (see Hu & Bentler, 1998, for a discussion), but practitioners often use general

structural equation models in practice. Second, how large should the model be? The number of latent factors and the number of indicators for each factor determine the size of the model. Third, how complex should the model be? Should the model have cross-loaded indicators? Reciprocal paths? Exogenous predictors? Each of these factors increases complexity and are not uncommon in applications. Fourth, should the model incorporate categorical variables? Categorical variables are very common in practice, especially with the use of survey data.

We developed three models that we concluded were commonly encountered in applied research: those with a small number of latent factors and a small number of indicators per factor; those with a small number of latent factors and a large number of indicators per factor; and those with two or more latent factors regressed on two or more measured exogenous variables. We selected one target model to represent each of these general model types. Our first model, Model 1, contained nine measured variables and three latent factors. Six of the nine measured variables loaded on a single factor (simple loadings), and the remaining three measured variables loaded on two factors (complex loadings). Further, Factor 2 was regressed on Factor 1, and Factor 3 was regressed on Factor 2 (creating a chain of causality for the latent variables). These three models are presented in Figures 1A, 1B, and 1C.

Our second model, Model 2, had the same basic structure as Model 1 but contained 15 measured variables, with five indicators per factor. Twelve of the measured variables loaded on a single factor and three measured variables loaded on two factors, and regression parameters were again present between adjacent latent factors. Finally, Model 3 contained 13 measured variables with the same measurement structure as Model 1 (three indicators per factor) but added four observed exogenous variables. Factor 1 depended on all four correlated exogenous variables, and Factors 2 and 3 depended on just the first and third exogenous variables.

The chain of causality between the latent variables made these models general structural equation models rather than CFAs. We felt this was important because chains of causality among latent variables are common in published research but rare in structural equation model Monte Carlo simulations. Also, our models vary in size. In fact, the similarity in structure between Models 1 and 2 allowed us to compare results solely on the basis of the size of the model. We also introduced increasing complexity, such as the exogenous variables in Model 3.

There are tradeoffs in any Monte Carlo design. Choosing a general structural equation model meant that we would not study CFAs. Introducing cross-loadings meant that our measurement models were not "clean." Though the external validity of the models in a Monte Carlo simulation will always be subject to criticism, researchers can strive for a balance between external validity, manageability, and answering specific research questions. One of the most useful strategies that a researcher can follow is to choose models that resemble those in published research.

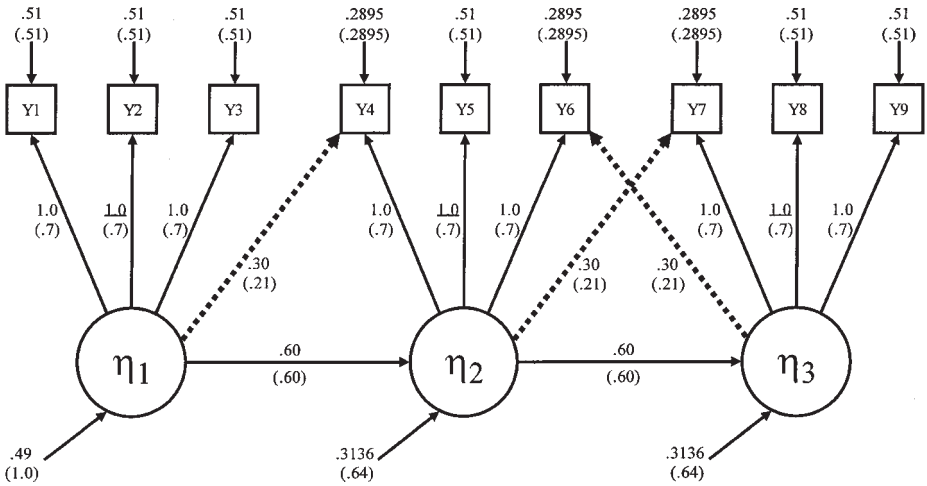

FIGURE 1A Model 1, three latent variables with three indicators and cross-loadings (dashed lines indicate misspecifications).

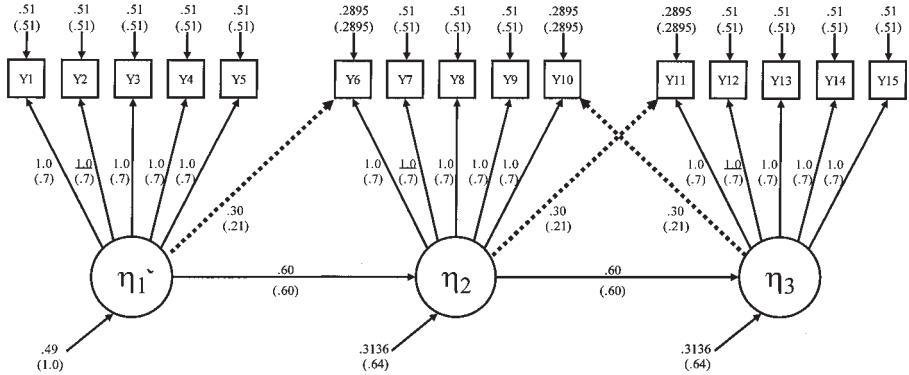

FIGURE 1B Model 2, three latent variables with five indicators and cross-loadings (dashed lines indicate misspecifications).

Step 3: Designing Specific Experimental Conditions

With a target model in place, the next step is to determine the experimental conditions to vary in the simulation. As discussed previously, the actual conditions that a researcher considers will vary depending on the research question. In this section,

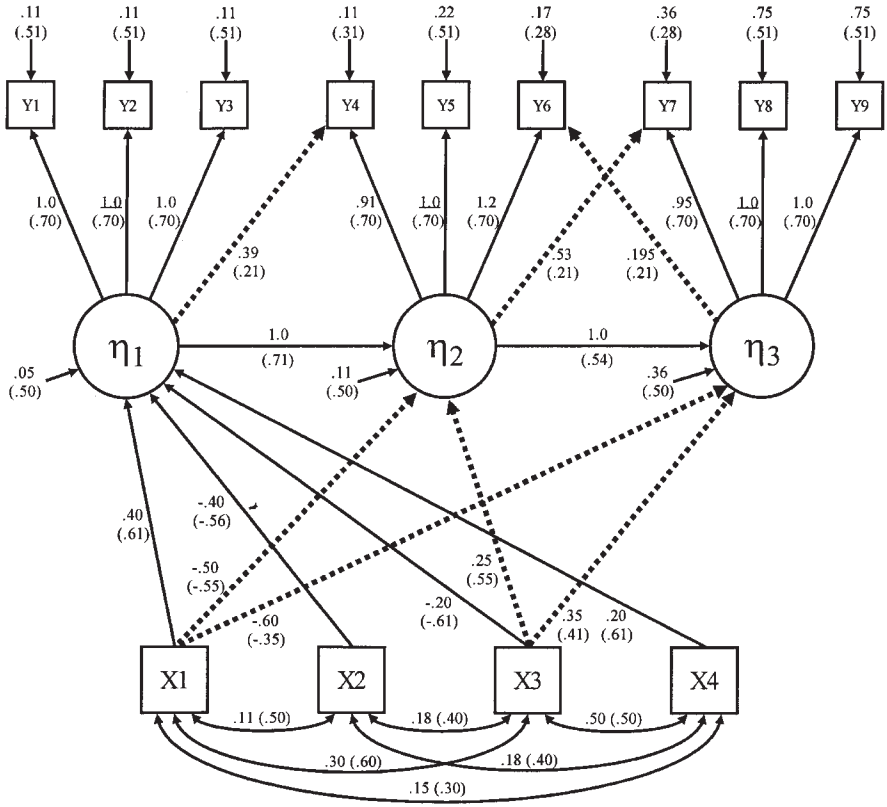

FIGURE 1C Model 3, three latent variables with three indicators, cross-loadings, and exogenous variables (dashed lines indicate misspecifications).

we discuss some of the most common experimental conditions a researcher may choose to evaluate.

One of the most important variables in a simulation is sample size. Often we do not know the properties of estimators or goodness-of-fit statistics for small to moderate sample sizes. Therefore, almost universally, Monte Carlo simulations vary sample size. The researcher has extensive choice in the number of sample sizes to consider, but sample sizes under 100 are especially important. Some areas of research, such as psychology or cross-national analyses in sociology or political science, routinely use sample sizes under 100. Much remains to be understood about the properties of estimators at such small sample sizes.

Two other conditions that are often varied in structural equation model Monte Carlo simulations are the distribution of the observed variables (multinormally distributed or not) and the estimation method (ML, generalized least squares,

2SLS, asymptotic distribution free, etc.). Whether a researcher chooses to vary these conditions will depend on his or her research questions.

An additional potential condition to vary is the extent of misspecification. If we acknowledge that researchers rarely work with perfect models, then it is important to consider the effect of misspecification on estimation, goodness of fit, and other outcomes. In a Monte Carlo simulation the true population model is known (i.e., created by the researcher). Samples are drawn from this known population and models are estimated on those samples. The models can be correct or misspecified. That is, the researcher can choose to estimate a model that is a mirror of the population model, or he or she can estimate a model that is different (i.e., misspecified) to some extent. In choosing misspecifications, we advise the researcher to pick ones that range from trivial to severe and that are reasonable theoretically.

An important question for misspecifications in Monte Carlos is whether to omit paths or include paths that are not in the true population model (see Curran, 1994, for a discussion of the benefits of each approach). Another issue is that more than one misspecification of a model is possible. Researchers may want to consider including several misspecifications of increasing severity. That is, one path could be omitted, then two, then three, in three separate specifications. Other strategies for increasing severity are also possible. Depending on the question of interest, researchers might also consider estimating the null independence model (no relation between the observed variables) as a specification.

To return to our running simulation example, our research questions dictated that we vary sample size, estimation method, and misspecification. We chose seven sample sizes—50, 75, 100, 200, 400, 800, and 1,000—along with two estimators: ML and 2SLS. We chose to limit the complexity of our simulation by ignoring the issue of what happens under conditions of nonnormality. Our decision was based on the belief that systematic examination of nonnormally distributed variables would require a separate simulation with a wide variety of distributions. This would multiply the number of experimental design conditions beyond what we could comfortably generate and analyze. So part of simulation design is to know how to restrict your questions to a manageable size.

The misspecifications we chose for our models involved the cross-loadings and the exogenous variables. We judged these omitted paths to be the most likely in empirical research.[1] The misspecified paths were omitted and are denoted by dashed lines in Figures 1A, 1B, and 1C. For each model type we selected five model specifications: a properly specified model (labeled Specification 1), where the estimated model perfectly corresponded to the population model; and four

[1]These misspecifications were theoretically reasonable because it is likely that cross-loadings might be ignored by researcher, and it is also likely that a researcher might have the effect of exogenous variables go only to the first variable in the chain.

misspecified models (Specifications 2, 3, 4, and 5). For Model 1, Specification 2 omitted the cross-loading linking Factor 2 with item 7; Specification 3 additionally omitted the cross-loading linking Factor 6 with item 3; Specification 4 additionally omitted the complex loading linking Factor 4 with item 1; and Specification 5 was the standard null independence model. The four misspecifications of Model 1 therefore introduced increasing misspecification—first one cross-loading was omitted, then two, and so on. Thus, Specification 4 was more severely misspecified than Specification 3, which was in turn more severely misspecified than Specification 2.

We similarly defined the misspecifications of Model 2. Specification 2 omitted the cross-loading linking Factor 2 with item 11; Specification 3 additionally omitted the cross-loading linking Factor 3 with item 10; Specification 4 additionally omitted the cross-loading linking Factor 1 with item 6; and Specification 5 was the standard null (uncorrelated variables) model.

The misspecifications of Model 3 were somewhat different: Specification 2 jointly omitted the set of three cross-loadings (Factor 2 to item 7, Factor 3 to item 6, and Factor 1 to item 4); Specification 3 jointly omitted the set of four regression parameters (Factor 2 on Predictor 1, Factor 3 on Predictor 1, Factor 2 on Predictor 3, and Factor 3 on Predictor 3); Specification 4 jointly combined the omissions of Specifications 2 and 3 (omission of the set of three factor loadings and the set of four regression parameters); and Specification 5 was the standard null (uncorrelated variables) model.

In sum, when the dust had cleared and we had decided on the specific experimental conditions to vary, there were 210 unique experimental conditions. Each of three models had five specifications (one proper and four misspecified). Each of these 15 "model types" had seven sample sizes (50, 75, 100, 200, 400, 800, 1,000). We estimated each with both the ML and 2SLS estimators. This resulted in 210 unique experimental conditions—three models by five specifications by seven sample sizes by two estimation methods.[2] It should now be apparent how exponential growth (and consequent problems with manageability) are easy to achieve in Monte Carlo simulations.

Step 4: Choosing Values of Population Parameters

At this point a researcher has selected a model (or models) and has determined the appropriate experimental conditions to be varied. The next step is to select specific values for the population model parameters. Like the other steps, this process should be a combination of theory, research, and utility. There are five issues to

[2]A rather amusing side note is that we originally considered this project to be a "pilot" project in preparation for a much larger study!

consider in picking the values of the parameters. First, they should reflect values commonly encountered in applied research. A traditional cutoff for a practically (vs. statistically) significant coefficient is a standardized coefficient of .10. As such, this forms a useful lower bound for coefficient values. Second, the R^2 values the chosen coefficients produce should also be reasonable for applied research. We recommend that the R^2s take values that are representative of the areas of research with which one is most concerned. For instance, R^2 values with cross-sectional, individual-level data frequently range between .2 and .8. Third, the parameters of the model should be statistically significant, even at the smallest sample size of the simulation.

Choosing values for the parameters becomes more complicated when misspecifications are part of the design. A fourth important consideration in that case is whether the model has enough power to detect the proposed misspecifications at a reasonable sample size. Alternatively, models may have too much power to detect misspecifications at all sample sizes. Researchers should investigate the power of their models (e.g., MacCallum, Browne, & Sugawara, 1996; Satorra & Saris, 1985) and select particular values for the coefficients to produce a reasonable power to detect them.[3] Table 1 presents the power estimates for our running example (we discuss the values we chose later in this article). It illustrates that, as would be expected, the power to detect misspecifications increases with sample size, model complexity, and extent of misspecification. The smallest model, Model 1, at the smallest sample sizes, has little power to detect (.065) the most minor misspecification. At the largest sample size, the power to detect this misspecification rises to .635. At more severe misspecifications, or in more complicated models, we reach power estimates of 1.0. The most important feature of Table 1 is the wide range of calculated estimates.

The fifth consideration in choosing the values of the misspecified models is the amount of "bias" in the estimates that will be introduced by the misspecifications. At the most extreme misspecifications we would like to see nonnegligible bias in the other estimated parameters. Table 2 presents a table of the expected bias for Model 1's four specifications when the ML fitting function is applied to the population covariance matrix. Specification 1, which is the perfectly specified model, shows no bias. The other specifications show bias increasing and encompassing more parameters, and several of the biases are clearly nonnegligible, with values exceeding 30%.

To summarize our decision process in choosing our parameter values, we based these decisions on issues of effect size (e.g., selection of R^2 values and bias that would be substantively interpretable), statistical significance (e.g., all parameters

[3]By power to detect, we mean the power to reject a false model, using the chi-square test. If power is too low, strongly misspecified models will likely not be rejected. If power is too high, then minor misspecifications may be rejected too often.

TABLE 1
Power Estimates as a Function of Sample Size and Misspecification

	Model df	*Min FCN*	*N = 50*	*N = 75*	*N = 100*	*N = 200*	*N = 400*	*N = 800*	*N = 1000*
Model 1									
Specification 2 (omit $\lambda_{7,2}$)	23	.01656	.065	.074	.084	.127	.239	.509	.635
Specification 3 (omit $\lambda_{7,2}$, $\lambda_{6,3}$)	24	.03763	.087	.111	.138	.267	.568	.926	.977
Specification 4 (omit $\lambda_{7,2}$, $\lambda_{6,3}$, $\lambda_{4,1}$)	25	.09441	.162	.243	.334	.689	.975	.999	1.0
Model 2									
Specification 2 (omit $\lambda_{11,2}$)	86	.03962	.068	.079	.091	.150	.313	.687	.823
Specification 3 (omit $\lambda_{11,2}$, $\lambda_{10,3}$)	87	.08275	.096	.127	.163	.348	.744	.992	.999
Specification 4 (omit $\lambda_{11,2}$, $\lambda_{10,3}$, $\lambda_{6,1}$)	88	.14081	.133	.194	.267	.602	.960	.999	1.0
Model 3									
Specification 2 (omit $\lambda_{7,2}$, $\lambda_{6,3}$, $\lambda_{4,1}$)	53	.12625	.151	.225	.312	.675	.977	1.0	1.0
Specification 3 (omit $\gamma_{2,1}$, $\gamma_{2,3}$, $\gamma_{3,1}$, $\gamma_{3,3}$)	54	.38457	.497	.746	.898	.999	1.0	1.0	1.0
Specification 4 (omit $\lambda_{7,2}$, $\lambda_{6,3}$, $\lambda_{4,1}$, $\gamma_{2,1}$, $\gamma_{2,3}$, $\gamma_{3,1}$, $\gamma_{3,3}$)	57	.53688	.684	.907	.981	1.0	1.0	1.0	1.0

TABLE 2
Bias in Model 1

	Specification 1		*Specification 2*		*Specification 3*		*Specification 4*	
Parameter	*Population Value*	*Percent Bias*	*Population Value*	*Percent Bias*	*Population Value*	*Percent Bias*	*Population Value*	*Percent Bias*
$\lambda_{1,1}$	1.0	—	1.0	0	1.0	0	1.0	0
$\lambda_{2,1}$	1.0[a]	—	1.0[a]	—	1.0[a]	—	1.0[a]	—
$\lambda_{3,1}$	1.0	—	1.0	0	1.0	0	1.0	0
$\lambda_{4,1}$	.300	—	.300	0	.380	26.7	0.0[b]	100
$\lambda_{4,2}$	1.0	—	1.0	0	.936	–6.4	1.206	20.6
$\lambda_{5,2}$	1.0[a]	—	1.0[a]	—	1.0[a]	—	1.0[a]	—
$\lambda_{6,2}$	1.0	—	.960	–4.0	1.286	28.6	1.237	23.7
$\lambda_{6,3}$	.300	—	.337	12.3	0.0[b]	100	0.0[b]	100
$\lambda_{7,2}$	.300	—	0.0[b]	100	0.0[b]	100	0.0[b]	100
$\lambda_{7,3}$	1.0	—	1.328	32.8	1.338	33.8	1.341	34.1
$\lambda_{8,3}$	1.0[a]	—	1.0[a]	—	1.0[a]	—	1.0[a]	—
$\lambda_{9,3}$	1.0	—	1.0	0	1.0	0	1.0	0
$\beta_{2,1}$	.600	—	.600	0	.555	–7.5	.631	5.2
$\beta_{3,2}$	.600	—	.653	8.8	.736	22.7	.715	19.2
$\psi_{1,1}$	.490	—	.490	0	.490	0	.490	0
$\psi_{2,2}$	.314	—	.314	0	.310	–1.3	.272	–13.4
$\psi_{3,3}$	.314	—	.232	–26.1	.188	–40.1	.198	–36.9

[a]Fixed parameter. [b]Omitted parameter.

were statistically significant even at the smallest sample size), and statistical power (e.g., selecting values that would result in a broad range of power to detect the misspecification across all sample sizes). For Model 1, the primary factor loadings were set to a standardized value of .70 (unstandardized value 1.0) to represent a communality of 49%. The complex loadings were set to a standardized value of .21 (unstandardized value .30). Finally, the regression parameters among the latent factors were set to a standardized value of .60 (unstandardized value of .60) to represent a multiple R^2 of 36%. For Model 2, all of the values were precisely those of Model 1 except for the addition of two measured variables per factor. For Model 3, the values of the factor loadings were equal to those of Model 1. However, the standardized regression parameter between Factors 1 and 2 was .71 and between Factors 2 and 3 was .54. These values differed from those of Models 1 and 2 given that these are now partial regressions with the inclusion of the four exogenous variables. The population values we chose for each model are included in their respective figures (1A, 1B, and 1C).

Some research questions may require that the values of the coefficients be varied as an experimental condition. That is, a researcher may choose multiple values for the coefficients and run each set of values as a separate condition to be analyzed. An example of this can be found in Anderson and Gerbing (1984). In addition, the previous suggestions are guidelines. In specific applications, other criteria may make more sense for the given question of interest. So it would be a mistake and too confining to consider our guidelines as "hard and fast" rules.

Step 5: Choosing an Appropriate Software Package

The choice of a Monte Carlo modeling package should be based on the requirements of the modeling design. Some simulation capability is available in most SEM packages, including AMOS, EQS, GAUSS/MECOSA, SAS/CALIS/IML, Fortran (ISML), MPLUS, and PRELIS/LISREL. These packages have been reviewed in general elsewhere (e.g., Hox, 1995; Waller, 1993). To avoid repetition, we only briefly discuss factors to consider when choosing a software package.

Packages have different strengths and weaknesses depending on the research question. Also, packages can change dramatically over time, adding new features and altering old ones. Researchers should, therefore, at the time of their simulation, review the possible software options to identify the optimal fit of software to the particular research design. For example, some research designs may require the software package to create nonnormal data, whereas others might require analysis of missing values. Bear in mind that multiple software packages may be needed to produce all the data relevant to a particular study.

For our Monte Carlo design, after much research we decided to utilize Version 5 of EQS (Bentler, 1995) for four reasons: a record of successful simulations in

previous studies, an ability to generate nonnormal distributions, an ability to generate data and fit the model in a single step, and an ability to fit a misspecified model to data generated from a different population model. However, EQS alone was not adequate to meet all of our analytic needs. We thus also used SAS extensively for data management, creation of additional fit statistics, and 2SLS estimation (using Proc SYSLIN).

Step 6: Executing the Simulations

With a target model designed and the values of the population parameters determined, a researcher can now create population covariance matrices. In our case, we created three population covariance matrices, one for each population model. The actual process of running a simulation will vary by statistical package, so we describe the process by which we generated the data for our example in EQS. This gives the general flavor of simulation and introduces technical considerations that cross all statistical packages.

Our simulated raw data was generated in EQS as random draws from our three population matrices. Although there were 210 unique experimental conditions, we generated 21 × 500 raw data sets in EQS. Specifically, we created 500 raw data sets at each sample size for each model type (with three model types and seven sample sizes, 21 × 500 raw data sets result). We then fit each of the five specifications within model type to the corresponding 500 data sets and produced parameter estimates and fit statistics. For example, consider a single raw data set generated for Model 1 at $N = 100$—we fit all five specifications of Model 1 at $N = 100$ to the same data set. Parameter estimates varied across specifications because of increased bias. In addition, fit statistics changed to reflect the increasing misspecification.

There are a number of technical considerations to consider in performing the simulations, regardless of what package generates the data. The first is the selection of samples. The random draws can produce any number of data sets, but some of these may suffer from problems. Specifically, some data sets may not converge or converge to "improper solutions." We call these "imperfect" samples, and the first technical consideration is whether they should be kept in the analysis.[4]

There is debate about whether nonconverged samples should remain in Monte Carlo simulations. Unless the object of interest is nonconverged samples, however, we suggest that a researcher avoid including them in the analysis. If the purpose of the Monte Carlo analysis is to provide realistic information to users of the

[4]Of course, the definition of a "nonconverged" sample depends on the maximum number of iterations a researcher will allow before declaring the sample to be nonconverged. Based on our experience with nonconverged models, we choose 100 iterations as our limit.

technique, then nonconverged samples, which are rarely assessed in practice, will provide irrelevant information and subsequently threaten external validity. It is important to make the conditions of the Monte Carlo simulation match practice to the greatest extent possible.

The researcher must also decide the fate of converged analyses with improper solutions. Improper solutions are estimates that take on values that would be impossible for the corresponding parameters (such as a negative error variance). This question is less clear-cut than nonconvergent samples, because researchers sometimes analyze models that have improper estimates whereas they should never interpret models that have not converged. Perhaps the safest strategy is to do analyses with and without improper solutions. This strategy allows the assessment of their influence on the conclusions of the experiment. Also, a researcher can conduct tests of whether the improper solution estimates are within sampling error of a proper value (e.g., Chen, Bollen, Paxton, Curran, & Kirby, forthcoming). A researcher should bear in mind that removing nonconverged and improper solutions can reduce the "useable" number of samples for the Monte Carlo analysis, and he or she should generate sufficient number of samples to take account of the loss of cases due to these two factors.

In our analysis, we decided the goal was to generate 500 "good" replications for each perfect specification at each sample size.[5] To achieve that many good replications, we initially generated 550 raw data sets for most sample sizes (650 data sets were generated for the smallest sample sizes [N = 50 and N = 75] because nonconverged samples and improper solutions are more likely). We next fit the properly specified models to these 550 raw data sets, some of which converged and some of which did not. We selected the first 500 replications of the initial pool of 550 replications that provided proper solutions and discarded the remainder. For example, if Model 2, N = 100, replication number 15 resulted in an improper solution, then it was discarded. If Model 1, N = 75, replication number 125 converged and was proper, it was included. However, if Model 3, N = 200, replication number 546 was converged and proper but 500 good replications had been obtained after replication number 532, then number 546 would be discarded anyway. This strategy resulted in 500 good solutions for each of the three properly specified conditions across all seven sample sizes.

In the misspecified models, any samples that were identified as "imperfect" in the perfectly specified model were removed. Therefore, in practice, only those 500 good solutions from the perfectly specified model were considered for the misspecified models. It is also possible, however, that the misspecified model

[5]Remember that replications at each condition are necessary to generate the sampling distribution. In the analysis of the generated data, means and other statistics can be estimated using the 500 replications. Five hundred replications provide a large enough sample size to accurately calculate statistics while taking less computer time to generate than other options, like 1,000 replications.

would create an improper solution in the misspecified estimation of that sample (we did not encounter any nonconverged samples in the misspecified models, although it is theoretically possible). The researcher would again need to decide whether to keep or exclude those samples. Our decision was to remove them. This resulted in fewer than 500 samples for analysis for some of the misspecified models.[6] As a concrete example, say for Model 1, $N = 100$, replication number 223 failed to converge under proper specification. This case would be excluded from the perfect specification *and all misspecifications as well*. Similarly, if Model 2, $N = 75$, replication number 32 estimated an improper solution for the proper solution, it would be removed from all misspecifications. Even if the estimated solutions for a misspecified model were all proper, the replication would not be part of the final Monte Carlo analysis. To illustrate our selection process, we provide Figure 2.

Another technical consideration in simulation is the use of "seeds." Random draws can be controlled by giving the program a starting point. That is, if the same seed (the starting point) is given in two subsequent analyses, the same data will be produced. This is a way of controlling the "randomness" of the random number generation. It allows for replication and is the way we used the same raw data across specifications. To acquire the seeds, we utilized a random number book (e.g., Beyer, 1984; Research and Education Association, 1984) because random numbers from calculators are not completely random.

Data generation takes time, which introduces several other technical considerations. First, to achieve convergence as quickly as possible for each replication, we suggest that the population parameters be used as starting values.[7] Second, we suggest that researchers determine a protocol for their data generation because it is unlikely that all the data will be generated in a single sitting. Having a protocol means that the data generation will be consistent across both time and collaborators. In general, it is impossible to have too much documentation during a Monte Carlo simulation. The protocol for our simulation was simple:

EQS: Changes to Program Before Each Run

- Change sample size.
- Change fit output file name.
- Change data output file name.

[6]Another way to view the exclusion criteria for the misspecified models is as follows. We utilized the same initial pool of raw data sets generated for the perfectly specified models (550 replications for $N = 100$ and greater, 650 replications for $N = 50$ and $N = 75$). For the misspecified models, however, there were two criteria for exclusion: (a) Cases were excluded if the misspecified model's solution was improper, and (b) cases were excluded if that case had resulted in an improper solution for the proper specification, regardless of the properness of the solution under misspecification.

[7]Starting values are needed because EQS estimates the model and produces fit statistics as part of its data generation.

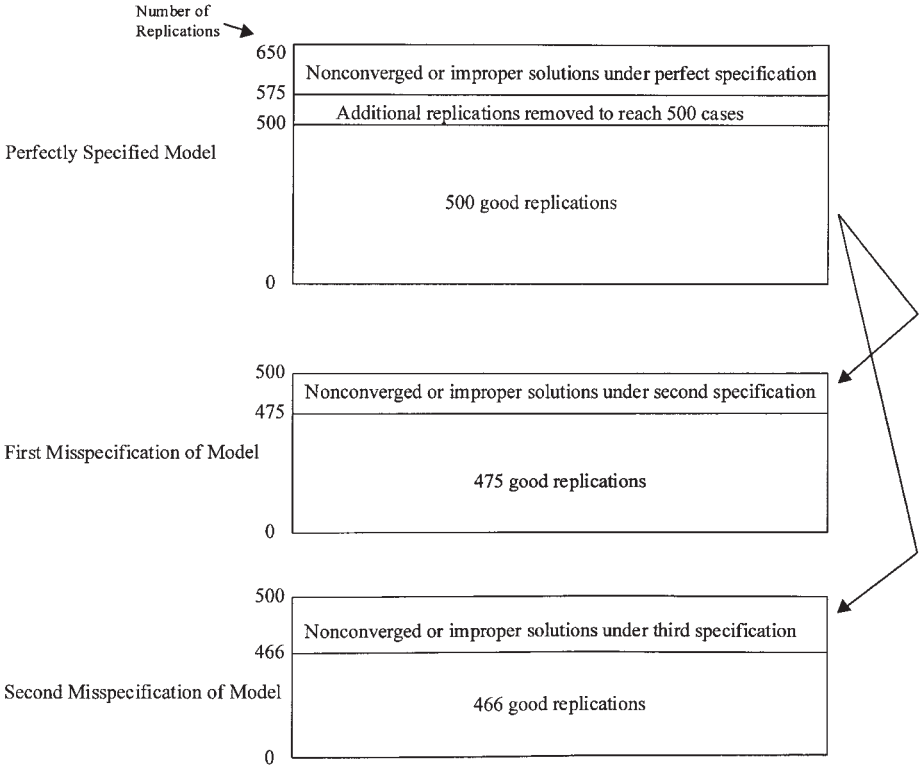

FIGURE 2 Selection of samples ($N = 50$ or $N = 75$). For $N > 75$, the picture looks similar, with 550 replications to begin.

- Change seed.
- Save as new program.

Modifications to Files After Each Run

- Delete output file.
- Rename data file.
- Move data file to data subdirectory.
- Move "fit" file to fit subdirectory.

A Monte Carlo simulation creates an enormous amount of output. Organization is therefore key to control over the project both at the time of data generation and during later analyses. Without clear, consistent naming conventions for files, it is

likely that a researcher will be unable to determine what was done after the simulation is completed. Researchers should determine a naming convention for files early in the project (before any data is generated) and modify it only if absolutely necessary. Our naming convention was MxSxNxxx.xxx, where *M* indicated model type (1, 2, or 3), *S* indicated specification (perfectly specified and the four misspecifications), and *N* indicated sample size. Various extension labels indicated whether the file was an EQS program (.eqs), a fit file (.fit), a SAS program (.sas), or various other types of programs.

To complete our discussion of the technical aspects of running a simulation, we provide two sample programs for EQS and SAS as Appendixes A and B. The EQS sample program presents Specification 1 of Model 1, which is the perfectly specified model. The paths described under the /EQUATIONS command therefore match the diagram in Figure 1A perfectly. The misspecified models (such as Model 1, Specification 2) would thus omit some paths in the /EQUATIONS command. The matrix provided in the /MATRIX command is the population covariance matrix. It is this matrix from which the random samples are drawn. The lines at the end of the program are the commands for the simulation itself. Of special interest are the requested parameter estimates and standard errors under /OUTPUT, and the specification of the seed (19922197) and number of replications (650) under /SIMULATION.

The SAS program reads in the data from the fit files, transforms it, and outputs a working SAS data set for future analyses. The input statement provides the variables placed into the .fit file by EQS. These include technical information, such as the method of estimation (method) and whether the sample was nonconverged (nonconv), information about the goodness-of-fit statistics (e.g., modchi, gfi), the parameter estimates (e.g., v7f3pe), and the standard errors (e.g., v7f3se). The next section removes all nonconverged and improper solutions and keeps 500 good samples. A few additional variables are created for reference and the data is output to a SAS data set.[8]

Step 7: File Storage

A researcher beginning a Monte Carlo simulation is unlikely to realize how much data will be created in the process. Ultimately, a vast amount of computer space will be needed for storage. For example, our 500 sets of raw data for $N = 1{,}000$ for Model 2, Specification 2 took up 108,900 kilobytes of space and the corresponding .fit file took up 679 kilobytes. Indeed, our current master subdirectory requires over 300 megabytes of data storage. Space is cheap and getting cheaper, but researchers should still consider space storage issues when performing a simulation. Another

[8]It should also be noted that while EQS gave us the ML estimates, we used SAS (and the raw datasets output by EQS) to create the 2SLS (Bollen, 1996) estimates.

consideration is that the original files need to be kept safe during later analyses, and multiple users may need to access the files. In this section, we discuss file storage issues in Monte Carlo simulations.

The first question a researcher should ask is whether the raw data needs to be saved. The raw data files are typically those that take up the most amount of space. Under some circumstances, saving the parameter estimates and goodness-of-fit statistics is all that is required. In that case, the raw data can be deleted, especially if seeds are recorded and data generation programs saved (in that case the raw data sets could be recreated if desired).

When there will be multiple users of the programs and data sets, it becomes crucial to keep original programs and data sets safe. Often questions arise later in a project that can only be answered by going back to the data creation programs. If these have been modified by one or more users, then it may be impossible to remember or recreate the simulation conditions. Therefore, an "archive" should be created to hold all original programs. A separate "working" area should be designated for programming. Files should not be modified outside the working area. Within the working area, researchers can have three main subdirectories for programming, a "testing" directory, a "finished programs" directory, and a "back-up" directory. All unfinished programs are housed in the testing directory and, once finished, are moved to the programs directory and the back-up directory. One way to make your data both accessible and safe is to implement read-only conventions for all archived, finished, and back-up files.[9]

To help present how a working Monte Carlo project could operate, we provide Figure 3, a picture of the ML portion of the file structure for our Monte Carlo simulation. To begin, consider the subdirectory labeled *archive*. This directory contains all the original programs: the EQS simulation programs, the raw data files created by EQS, the goodness-of-fit files created by EQS, and the original SAS programs to determine improper solutions and obtain 500 good replications. These files are designated as read-only. Any member of the simulation group can access them for reference, but no one can change them. They serve as a record of our process. The working .fit files and SAS data sets have their own subdirectories. The safe method of testing and saving working programs is illustrated under the "sasprograms" directory. Once these were established, we burned the entire file structure on compact discs for permanent storage and backup.

Step 8: Troubleshooting and Verification

Once the simulation is completed, the data stored, and programming begun, how do you know whether you did it correctly? Fortunately, you can perform a number of

[9]If multiple users are involved in the project, then file access becomes as important as file storage. There are many options for file access, including networks, mainframes, and the web.

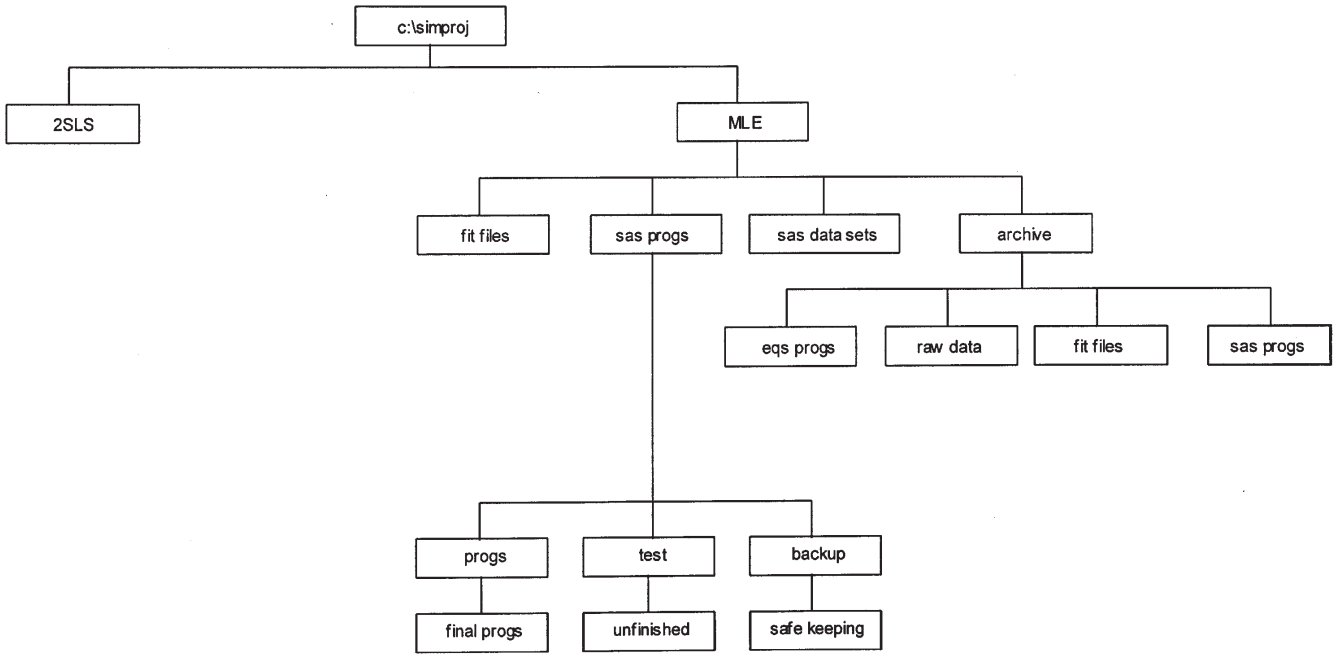

FIGURE 3 File structure.

checks to provide verification that the simulation worked. For example, do raw data files with larger sample sizes take up more storage space? Does the number of resultant replications match what you asked the program to create? These are checks for logical consistency. Another check is whether the results seem reasonable. For example, at sample sizes at or above 1,000, parameter estimates should be within .01 of the population parameter. Another check would be whether empirical power estimates match the theoretically determined power estimates. That is, if a given model, specification, and sample size has .10 power to detect a misspecification, then the chi-square test of fit should be rejected at that approximate frequency across the 500 replications.

Step 9: Summarizing Results

Once the Monte Carlo simulation is completed, results obtained, and the data verified, a researcher must determine how to present the results. Because Monte Carlo simulations can produce so much data, data summary and presentation is not an easy task. In this section we briefly discuss some ways to summarize and present results.

There are three main ways to present output: descriptive, graphical, and inferential. Descriptive statistics present information concisely and simply. Researchers may only need to present a mean or variance of a sampling distribution to illustrate a point. Other descriptive statistics to consider reporting include the mean relative bias {[(coefficient estimate–population parameter value)/population parameter value] × 100}, or the mean, mean square error [(coefficient estimate–population parameter)^2]. Correlation or covariance tables can also concisely represent great quantities of data.

Graphical techniques are also extremely useful in presenting data. Figures such as box plots, scattergrams, or power curves can succinctly demonstrate the findings. Of course, the graphical technique a researcher chooses will depend on the research question and findings to be reported. Graphical representations of data are reviewed in a number of excellent sources (e.g., Cleveland, 1993; Tufte, 1983).

Inferential statistics can augment the descriptive and graphical analysis. For example, design factors such as sample size, model type, and estimation method can be dummy or effect coded, and main effects and interactions among design factors can be evaluated using standard regression procedures. Logistic regressions can be used for categorical outcomes, and generalized least squares methods can be used to account for heteroskedastic distributions of errors commonly found in simulation outcomes. Taken together, these techniques allow for the formal testing of the significance the design factors as well as the computation of various effect size estimates (e.g., percent of variance explained). Given the tremendous number of observations stemming from the multiple replications (e.g., for our 210 conditions

we generated approximately 105,000 samples), it is common for all estimated effects to drastically exceed traditional significance levels. It is thus important to interpret the meaningfulness of effects using effect sizes as well.

CONCLUSION

Monte Carlo simulations are growing in popularity, as researchers consider the small sample properties of estimators and goodness-of-fit statistics in SEM. Although every simulation is different, they also hold many features in common. This article attempted to provide an overview of the design and implementation of Monte Carlo simulations for structural equation models with the goal of aiding future researchers in their projects. We used a running example throughout the article to provide a concrete example of a working Monte Carlo project. Researchers are likely to encounter many unique situations in their own modeling, but we hope that this article provides a useful general orientation to get them started.

REFERENCES

Anderson, J. C., & Gerbing, D. W. (1984). The effect of sampling error on convergence, improper solutions, and goodness of fit indices for MLE CFA. *Psychometrika, 49,*155–173.

Bentler, P. M. (1995). *EQS: Structural equations program manual* (Version 5.0). Los Angeles: BMDP Statistical Software.

Beyer, W. H. (1984). *CRC standard mathematical tables* (27th ed.). Boca Raton, FL: CRC Press.

Bollen, K. A. (1996). An alternative 2SLS estimator for latent variable models. *Psychometrika, 61,* 109–121.

Chen, F., Bollen, K. A., Paxton, P., Curran, P. J., & Kirby, J. (forthcoming). Improper solutions in structural equation models: Causes, consequences, and strategies. *Sociological Methods and Research.*

Cleveland, W. S. (1993). *Visualizing data.* Summit, NJ: Hobart Press.

Curran, P. J. (1994). The robustness of confirmatory factor analysis to model misspecification and violations of normality. *Dissertation Abstracts International, 55,* 1220.

Curran, P. J., West, S. G., & Finch, J. (1996). The robustness of test statistics to non-normality and specification error in confirmatory factor analysis. *Psychological Methods, 1,* 16–29.

Gentle, J. E. (1985). Monte Carlo methods. In S. Kotz & N. L. Johnson (Eds.), *The encyclopedia of statistical sciences* (Vol. 5, pp. 612–617). New York: Wiley.

Gerbing, D. W., & Anderson, J. C. (1993). Monte Carlo evaluations of goodness-of-fit indices for structural equation models. In K. A. Bollen & J. S. Long (Eds.), *Testing structural equation models* (pp. 40–65). Newbury Park, CA: Sage.

Hox, J. J. (1995). AMOS, EQS, and LISREL for windows: A comparative approach. *Structural Equation Modeling, 1,* 79–91.

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

Kennedy, P. (1992). *A guide to econometrics* (3rd ed.). Cambridge, MA: MIT Press.

MacCallum, R. C., Browne, M. W., & Sugawara, H. M. (1996). Power analysis and determination of sample size for covariance structure modeling. *Psychological Methods, 1,* 130–149.

Mooney, C. Z. (1997). *Monte Carlo simulation.* Thousand Oaks, CA: Sage.
Muthén, B. O., & Kaplan, D. (1985). A comparison of some methodologies for the factor analysis of non-normal Likert variables. *British Journal of Mathematical and Statistical Psychology, 38,* 171–189.
Muthén, B. O., & Kaplan, D. (1992). A comparison of some methodologies for the factor analysis of non-normal Likert variables: A note on the size of the model. *British Journal of Mathematical and Statistical Psychology, 45,* 19–30.
Research and Education Association (staff, with M. Fogiel, Director). (1984). *Handbook of mathematical, scientific, and engineering formulas, tables, functions, graphs, transforms.* New York: Author.
Rubinstein, R. Y. (1981). *Simulation and the Monte Carlo method.* New York: Wiley
Satorra, A., & Saris, W. (1985). Power of the likelihood ratio test in covariance structure analysis. *Psychometrika, 51,* 83–90.
Smith, V. K. (1973). *Monte Carlo methods: Their role for econometrics.* Lexington, MA: Lexington Books.
Tufte, E. R. (1983). *The visual display of quantitative information.* Cheshire, CT: Graphics Press.
Waller, N. G. (1993). Software review: Seven confirmatory factor analysis programs: EQS, EzPath, LINCS, LISCOMP, LISREL 7, SIMPLIS and CALIS. *Applied Psychological Measurement, 17,* 73–100.

APPENDIX A

Sample EQS Program for Model 1, Specification 1, Sample Size 50

```
/TITLE  Model 1: 3-factor 9-indicator; primary loadings=1.0; secondary loadings=.30; regression beta=.60;
/SPECIFICATIONS
VAR=9; CAS=50; MET=ML; MAT=COV;
/EQUATIONS
v1=1*f1+e1;
v2=1f1+e2;
v3=1*f1+e3;
v4=1*f2+.3*f1+e4;
v5=1f2+e5;
v6=1*f2+.3*f3+e6;
v7=1*f3+.3*f2+e7;
v8=1f3+e8;
v9=1*f3+e9;
f1=d1;
f2=.6*f1+d2;
f3=.6*f2+d3;
/VARIANCES
e1=.51*; e2=.51*; e3=.51*; e4=.2895*; e5=.51*;
e6=.2895*; e7=.2895*; e8=.51*; e9=.51*;
d1=.49*; d2=.3136*; d3=.3136*;
```

```
/MATRIX
1
0.49      1
0.49      0.49      1
0.441     0.441     0.441     1
0.294     0.294     0.294     0.5782    1
0.34692   0.34692   0.34692   0.682276  0.5782    1
0.2646    0.2646    0.2646    0.52038   0.441     0.61446   1
0.1764    0.1764    0.1764    0.34692   0.294     0.441     0.5782    1
0.1764    0.1764    0.1764    0.34692   0.294     0.441     0.5782
0.49      1
/TEC
itr=100;
/OUTPUT
pa; se;
data='c:\simproj\MLE\archive\fit\m1s1n050.fit';
/SIMULATION
seed=19922197;
replication=650;
population=matrix;
data='n50';
save=con;
/PRINT
digit=6;
/END
```

APPENDIX B

Sample SAS Program for Model 1, Specification 1, Sample Size 50

```
    * MODEL 1
    SPECIFICATION 1

    50 CASES

filename in1 'c:\simproj\MLE\archive\fit\m1s1n050.fit';
libname out1 'c:\simproj\sasdata\';

*Model 1, Specification 1, N=50;
*'pe'=parameter estimate, 'se'=standard error, 'rb'=relative bias;
*'sq'=squared error;
```

```
* note: rather than the typical 550 replications, this one had 650;

data a; infile in1;
 input   method concode nonconv nullchi modchi moddf modprob
         nfi tli cfi gfi agfi rmsr srmsr rmsea rmscilo rmscihi iterate
         e1e1pe e2e2pe e3e3pe e4e4pe e5e5pe e6e6pe e7e7pe e8e8pe e9e9pe
         d1d1pe d2d2pe d3d3pe v1f1pe v3f1pe v4f1pe v4f2pe v6f2pe v6f3pe
         v7f2pe v7f3pe v9f3pe f2f1pe f3f2pe
         e1e1se e2e2se e3e3se e4e4se e5e5se e6e6se e7e7se e8e8se e9e9se
         d1d1se d2d2se d3d3se v1f1se v3f1se v4f1se v4f2se v6f2se v6f3se
         v7f2se v7f3se v9f3se f2f1se f3f2se;

*creating index variable that indexes observations;
index=_N_;

* going through and picking out the converged solutions only;
* first I keep only the converged solutions;
* then I also pick out only the ones with proper solutions;
* then I keep only the first 500 of the converged solutions;
data b; set a; where nonconv=0;
data bb; set b; where concode=0;
index2=_N_;
data c; set bb; where index2 < 501;

data working; set c;
* creating a reject variable to record if chi-square was rejected;
reject=0; if modprob lt .050 then reject=1;

* cleaning the RMSEA confidence intervals;
if rmscilo=–99 then rmscilo=.; if rmscihi=99 then rmscihi=.;
model=1; spec=1; n=50;
run;

proc means data=working; run;

data out1.m1s1n050; set working;
```