2

Using the EQS Program

Now that the basic EQS notation is understood, it's time to see (a) how these symbols can be linked together to form admissible program statements and paragraphs used in the building of an SEM input file, and (b) how to execute these input files so that the model specified can be tested. My primary aim here is to present a multifaceted albeit thorough overview of these program capabilities; consequently, this chapter is necessarily extensive. However, after working, through this material, I am hopeful that you will have a better "feel" for the program and a substantially better appreciation of the rich bank of model-building and model-testing resources that await you in using EQS.

We begin this guided tour of EQS programming by first inspecting all required and most optional components of the input file. For a more comprehensive view of how these components blend together, I then review their application with respect to three different models (i.e., first-order CFA, second-order CFA, and full SEM). Along the way, I introduce you to the important concept of model (or statistical) identification. At the next stop in our tour, the question of how to create an EQS input file is addressed. Here, I present you with three different approaches to the task—manual; interactive, using the BUILD_EQS option; and graphic, using the DIAGRAMMER option. Based on the same three models demonstrated at the previous stop, I walk you through the building process as it relates to both the interactive and graphic methods. Continuing on to the next stop on the tour, we visit the output section of this process. Here, a brief albeit general overview of

EQS output resulting from an executed input file is presented. Finally, at the last stop on the tour, I present you with a summary of possible error messages that can occur when using the EQS program.

COMPONENTS OF THE EQS INPUT FILE

An EQS input file describes the model under study. It is composed of several statements clustered within paragraphs according to certain rules. In this subsection, we look at the basic dicta governing the construction of EQS input files, then examine the basic components of an EQS input file, and, finally, review three methods to formulate and execute EQS input files related to three different SEM models.

Basic Rules and Caveats in Creating EQS Input Files

Keywords

As in most computer programs, EQS uses a system of keywords that are interpreted by the program as basic commands. Primary key words in EQS are the names used to identify particular paragraphs comprising the input file (e.g., the /SPECIFICATIONS paragraph). Secondary keywords represent each statement within a paragraph; they operate as subcommands. As such, they refer to one particular aspect of the paragraph, thereby allowing the user to choose one of several options pertinent to this component. For example, one aspect of the /SPECIFICATIONS paragraph involves the method of estimation. Of the available choices, the user may wish to employ generalized least squares estimation, which would be stated as ME=GLS, where ME is the abbreviated form of the keyword METHOD, a subcommand indicating the method of estimation to be used, and GLS is the three-letter keyword indicating that generalized least squares is the estimation method to be used.

Two major rules govern the use of paragraph keywords: (a) they must always be preceded by a slash (e.g., /SPECIFICATIONS), and (b) the line in which the keyword appears must contain no other input information. One optional use of paragraph keywords is that they may be abbreviated to the first three characters (e.g., /SPE). However, the user may prefer more than three letters because it results in a more meaningful keyword, and this choice is quite acceptable. For example, abbreviating /SPECIFICATIONS to /SPEC appears to be a better choice.

Descriptive Statements

All information describing the model and data must be expressed as specific statements written within the appropriate paragraph. A semicolon (;) must separate each statement. Although not necessary, it is often helpful to construct input files in such a way that renders them easy to read. One way to do this is to start paragraph keywords in column 1, with all other statements within the paragraph indented a few characters. Finally, all input statements should be specified in no more than 80 columns per line.

At times, you may wish to insert a reminder comment in the input file regarding some aspect of the data or analysis. EQS allows you to include such in-line comments in any line of the input file by means of an exclamation mark (!), which must precede the comment. The program then ignores all material to the right of the exclamation symbol.

File Editors

For EQS to run, the input file must be a plain file that contains no hidden control characters. Because most word processors (e.g., Word™, Wordperfect™) use such symbols in the formulation of text, the invisible characters must first be stripped before they can be read by EQS. However, this constraint is easily addressed by saving the input file in ASCII format, which can be read by EQS.

Data Input

Data in the form of a covariance or correlation matrix can be embedded within the input file or reside in an external file. Raw data must always reside in a separate file. Specific details regarding the integration of data files with the EQS input file are addressed in the following subsection.

Basic Components of the EQS Input File

As discussed earlier, the basic structure of an EQS input file comprises a series of statements grouped within paragraphs, each of which is introduced by means of a keyword preceded by a slash. However, not all paragraphs are needed in any particular input file. At this time, important features related to six basic paragraphs are reviewed; other paragraphs are examined in subsequent chapters as they bear on particular model applications. For greater elaboration on each subsection, readers are referred to the manual (Bentler, 2005) or the user's guide (Bentler & Wu, 2002), both available via the Help window of the program.

/TITLE (Optional)

Although this paragraph is optional (i.e., not required to run an EQS job), it is not only highly recommended to use it but also that a generous amount of information is included in the title. Not uncommon is the situation in which what might have seemed obvious when the initial analysis of a model was conducted may not seem quite so obvious several months later when, for whatever reason, the user reexamines the data using the same input files. Liberal use of title information is particularly helpful when numerous EQS runs were implemented for a given data set. EQS allows as many lines as desired in this title section.

An example of a /TITLE paragraph is as follows:

```
/TITLE
 CFA of BDI-French Version
 Initial Model
```

/SPECIFICATIONS (SPEC; Required)

This paragraph defines both the data to be analyzed and the method of analysis to be used. In particular, it details (a) information related to the data (location, matrix form); (b) the number of cases and input variables; (c) the desired method of estimation; and (c) other information that may be needed to guide the EQS run.

There are three important factors with respect to the /SPEC paragraph: (1) although input information can be placed in any order, a semicolon must delineate each operand; (2) information related to sample size and number of variables must always be provided; and (3) subcommand keywords can be abbreviated to three- and two-letter format. We now take a general look at a few aspects of this paragraph.

Data (DA). Use of this keyword is governed by two conditions: (1) the data to be analyzed exists in some external file, and (2) the computing environment involves an interactive system. As such, the statement would read DA = 'C:\EQS61\Files\FRBDI.dat';, where C:EQS61\Files represents the location of the data and FRBDI.dat represents the name of the data file, both of which are enclosed within single quotes. As explained earlier, these data can be in raw, correlation, or covariance matrix form.

Variables (VAR). The number stated here should represent the total number of variables in the data set; it does not represent the number of variables to be analyzed in any specific EQS run. The program automatically sorts out which variables to include in the analyses by its subsequent reading of the /EQUATIONS and /VARIANCES paragraphs.

Cases (CAS). This term defines the number of subjects comprising the sample data. The number should represent all cases regardless of any that may be deleted later. Such deletions, as specified by the user and considered by EQS in the analytic procedures, never actually alter the original data set; thus, the number of cases always stays the same. If the input data are in the form of a correlation or covariance matrix, it is critical that the number of cases be correct because EQS has no way of checking its accuracy; an incorrect number will lead to incorrect statistics. However, if the input data comprise raw scores, this number does not need to be specific; EQS automatically tallies the number of cases in the file, which is used in all computations. Any discrepancy between the specified and actual number of cases is noted in the output.

Method (ME). The EQS user can now select from nine methods of estimation: maximum likelihood (ML), least squares (LS), generalized least squares (GLS), elliptical LS (ELS), elliptical GLS (EGLS), elliptical reweighted LS (ERLS), heterogeneous kurtosis GLS (HKGLS), heterogeneous kurtosis RLS (HKRLS), and arbitrary distribution GLS (AGLS). If no method is specified, the program automatically uses ML estimation; in other words, ML represents the default method. For details related to these estimation methods, see the manual (Bentler, 2005).

A special feature of the EQS program is that it also allows for the specification of several estimation methods in a single job submission (e.g., ME=ML, AGLS;). A maximum of two methods can be specified when AGLS is included and a maximum of three methods otherwise. The elliptical and arbitrary distribution methods are always automatically preceded by their normal theory counterparts. More specifically, the ELS, EGLS, and ERLS methods are preceded by the normal theory methods of LS, GLS, and ML, respectively; AGLS is preceded by LS. Nonetheless, the user can override these default prior methods by simply specifying another valid method. (For more details regarding both the specification and appropriate use of these methods, see Bentler, 2005).

Finally, an extremely valuable feature unique to the EQS program is the availability of robust statistics associated with any selected method of estimation except AGLS. For example, by specifying ME=ML,ROBUST, the output provides a robust chi square statistic (χ^2) called the Satorra–Bentler scaled statistic (S-Bχ^2; Satorra & Bentler, 1988, 1994) and robust standard errors (Bentler & Dijkstra, 1985), both of which have been corrected for non-normality in large samples. The S-Bχ^2 has been shown to more closely approximate χ^2 than the usual test statistic and the robust standard errors to be correct in large samples despite the fact that the distributional assumptions regarding the variables are wrong (Bentler, 2005). Although these robust statistics are computationally demanding, they have been shown to perform better than uncorrected statistics where the assumption of normality fails to hold and better than asymptotic distribution-free GLS (AGLS) in all but the largest samples (Chou, Bentler, & Satorra, 1991; Hu, Bentler, & Kano, 1992). One important caveat regarding the use of robust statistics, however, is that they can be computed only from raw data.

Analysis (ANAL). This keyword describes the type of matrix to be analyzed if it is something other than a covariance matrix, which is default. If the analyses are to be based on the covariance matrix but the data were input as a correlation matrix, then the standard deviations must be added to the input. This is accomplished by providing a separate paragraph introduced with the keyword /STANDARD DEVIATIONS (or /STA) and then listing the standard deviations on the next line, leaving one blank between each; more than one line can be used if necessary. Three instructions must be remembered when using the /STA paragraph: (1) there must be exactly the same number of standard deviations as there are variables in the correlation matrix being read, (2) the standard deviations must be in the same order as the variables in the input matrix, and (3) no semicolon is used to end this input section.

Matrix (MA). This keyword describes the form of the input data—that is, whether it is a correlation, covariance, or raw data matrix. As noted earlier, raw data are always presumed to reside in an external file; covariance or correlation data matrices can either be embedded within the input file (i.e., as an internal file) or reside in a separate (external) file. When covariance (or correlation) matrices reside as an internal file, the data are specified within a separate paragraph, labeled /MATRIX (MAT). If, on the other hand, they reside in an external file, their location is noted via the DATA (DA) keyword. Once EQS has located the data file, it then needs to know how to read the data. If the data are in free format, which is default, EQS can read the file as long as each element in the matrix is separated by at least one blank space. If the data are not in free format, this information is provided by means of a Fortran statement, which must be immediately preceded by the FORMAT (FO) keyword. Finally, if the user wishes the analysis to not be based on the covariance matrix (which is default), the matrix to be analyzed must be noted using the ANALYSIS (ANAL) keyword.

Now, let's pull this all together and look at two specification paragraphs describing an input data matrix.

(i) The data matrix as an internal file:

```
/SPECIFICATIONS
 CASE=250; VAR=4; ME=ML; MA=COR;
/MATRIX
 1.00
  .34 1.00
  .55  .27 1.00
  .48  .33  .63 1.00
/STANDARD DEVIATIONS
 1.09  .59  .98 1.10
```

NOTE:

- The input of data never requires a semicolon. Thus, there is no ";" in the /MATRIX and /STANDARD DEVIATIONS paragraphs.
- Although the input data are in correlation matrix form, specification of the standard deviations enables analyses to be based on the covariance matrix.

(ii) The data matrix as an external file:

```
/SPECIFICATIONS
CASE=250; VAR=4; ME=ML, ROBUST; MA=COV;
DA='C:\EQS61\Files\FRBDI.dat'; FO=(4F2.0);
```

In addition to the subcommands of the /SPEC paragraph shown here, others are either optional or used with particular applications. Because many such applications are described in this book, they are not discussed until their appearance in subsequent chapters.

/LABELS (Optional)

This paragraph can be used to identify the names of observed (V's) and/or latent (F's) variables in the model. Labels may be one to eight characters in length and are assigned to only V- and F-type variables. Observed variables should be numbered according to their position in the data set. Thus, the specification that V5 = MATH indicates that the fifth variable in the data matrix is to be labeled "MATH." Although the numbering of latent variables is arbitrary, it should be logically sequenced within the context of the model.

Because EQS automatically assigns V1 to the first observed variable, V2 to the second variable, and so on, these designations are used as labels if the user does not provide names for the variables; likewise the latent variables are automatically assigned the labels F1, F2, and so forth. If the user provides labels for only some of the variables, these names override the default labels (e.g., V1).

/EQUATIONS (EQU; Required)[1]

This keyword signals specification information regarding the model under study. Specifically, the paragraph defines every regression path in the model. By means of a series of equation statements, the /EQU paragraph specifies all linkages among the independent and dependent variables, as well as among dependent and dependent variables, and identifies those parameters to be constrained (i.e., to zero, 1.0, or some other value) and those to be freely estimated.

Before completing this part of the input file, it is critical to construct a path diagram of your model in which (a) all observed and latent variables are clearly labeled; (b) all structural regression paths are specified; (c) all error and disturbance terms are specified, along with their related regression paths; (d) all hypothesized covariances (among independent variables) are specified; and (e) all parameters to be estimated (including factor variances) are identified by means of an asterisk. Once you have this visual representation of your model, it is easy to complete all equations in the /EQU paragraph by simply reading off the path diagram.

[1]With EQS, there is now a newly established /MODEL paragraph that allows the user to express model specifications in a more simplified and concise manner. In addition, the E and D variables are generated automatically by the program. When this paragraph is used, it replaces the /EQU keyword as well as the /VAR and /COV keywords.

How to Write Equations. As noted in chapter 1, the completion of these equations is carried out as follows. For each *dependent* variable (i.e., any variable having an arrow pointing toward it), write one equation summarizing the direct impact on it from other variables in the model. Thus, there will always be as many equations in the /EQUATIONS paragraph as there are dependent variables in the model being tested. The dependent variable will always be on the left-hand side of the equation, with all *independent* (i.e., explanatory) variables appearing on the right. (Recall from Chap. 1 that in EQS, the terms *dependent* and *independent* variables are defined within the context of the Bentler–Weeks model.) Finally, once the equations have been formed, be sure to insert asterisks next to all parameters to be estimated. EQS provides estimates for only the asterisked parameters; all others are regarded as fixed.

The Use of Start Values. Start values refer to the point at which a program begins an iterative process to establish parameter estimates. Users can either allow the program to supply these values (EQS uses default values such as 0.0 for covariances) or provide their own values. User-provided start values represent a "best guess" of what the expected value of a particular parameter estimate will be. These best-guess values are included in the equation, acting as modifiers of the parameters to be estimated; they always precede the asterisk.

Although start values need not be specified for most EQS jobs, they often facilitate the iterative process when complex models are under study; such models can generate problems of nonconvergence if the start values provided by EQS are inadequate. However, for the EQS user, the problem of trying to determine appropriate start values is gone forever as a result of an ingenious feature called the RETEST option, which automatically generates these start values and then allows the user to edit the input file accordingly. This unique time-saving tool is demonstrated in chapter 6. (For a more extensive discussion of start values, see Bollen, 1989a.)

Now, let's examine a few /Equations paragraphs.

(i) Equations without start values:

```
/EQUATIONS
V1 = F1 + E1;
F3 = *F1 + *F2 + D3;
```

(ii) Equations with start values:

```
/EQUATIONS
V1 = .9 F1 + E1;
F3 = .6 *F1 + -.2*F2 + D3;
```

NOTE:
- The start value associated with F1 is fixed at .9 by choice.
- The start values associated with E1 and D3 have no value specified; these parameters are considered to be known and are fixed to 1.00 by program default. (A more detailed explanation of this phenomenon is provided later.)

/VARIANCES (VAR; Required)

This paragraph specifies the status of variances related to independent variables in the model. As such, each variance must be identified either as a fixed parameter in the model or as one to be freely estimated. (Recall that variances for dependent variables are never specified, regardless of whether they are fixed or free.) As in the /EQUATIONS paragraph, variances to be estimated are identified with an asterisk. Although further elaboration of this paragraph is provided later in this chapter, let's look at a couple of examples of the /VAR paragraph.

(i) Variances without start values:

```
/VARIANCES
F1, F2 = *;
F1 to F3 = *;
F4-F6 = *;
```

NOTE:
- Consecutive lists of variables can be specified using either "to" or a dash. As such, F1 to F3 indicates that the variances of F1, F2, and F3 are to be estimated. Likewise, F4, through F6 specifies that the variances of F4, F5, and F6 are to be estimated.
- Variable labels (as defined in the /LABEL paragraph) can be used in lieu of the V- or F-labels. However, because labels with fewer than eight characters can cause an error, it is recommended that, in this case, the hyphen be used rather than the "to" convention.

(ii) Variances with start values:

```
/VARIANCES
F1,F2 = .3*;
F1 to F3 = .4*;
F4-F6 = .6*;
```

/COVARIANCES (COV; Optional)

For obvious reasons, this paragraph is necessary only when covariances are specified in the model. As such, it is used to specify both fixed nonzero and free covariances among the independent variables. However, any variable involved in a covariance must also have its variance specified in the /VAR paragraph; consequently, covariances also cannot be specified for dependent variables.[2] In specifying a covariance, the pair of variables is stated and separated by a comma, as exemplified in the following:

```
/COVARIANCES
E1,E3 = *;
F1 to F3 = *;
F4-F6 = *;
```

NOTE:
- Use of "to" and "-" indicate that the covariances related to all possible pairs of variables are to be estimated. For example, the specification of "F1 to F3 = *;" indicates that estimated parameters are F1,F2; F1,F3; F2,F3.

The final paragraph in all EQS input files must be /END. This keyword marks the termination of program input.

Input File Examples

To provide a more comprehensive view of how the various sections of the EQS input file relate to the path diagram of a particular model, we now review three simple albeit different models: (a) a first-order CFA model (Fig. 2.1), (b) a second-order CFA model (Fig. 2.2), and (c) a full SEM model (Fig. 2.3). However, due to space limitations, a complete input file is shown only for Fig. 2.1; the /SPECIFICATIONS and /LABELS paragraphs are absent for the two remaining models. Although

[2]If a user wishes to have two dependent variables covary, one option is to specify a covariance between their disturbance terms because these residuals are always independent variables in a model.

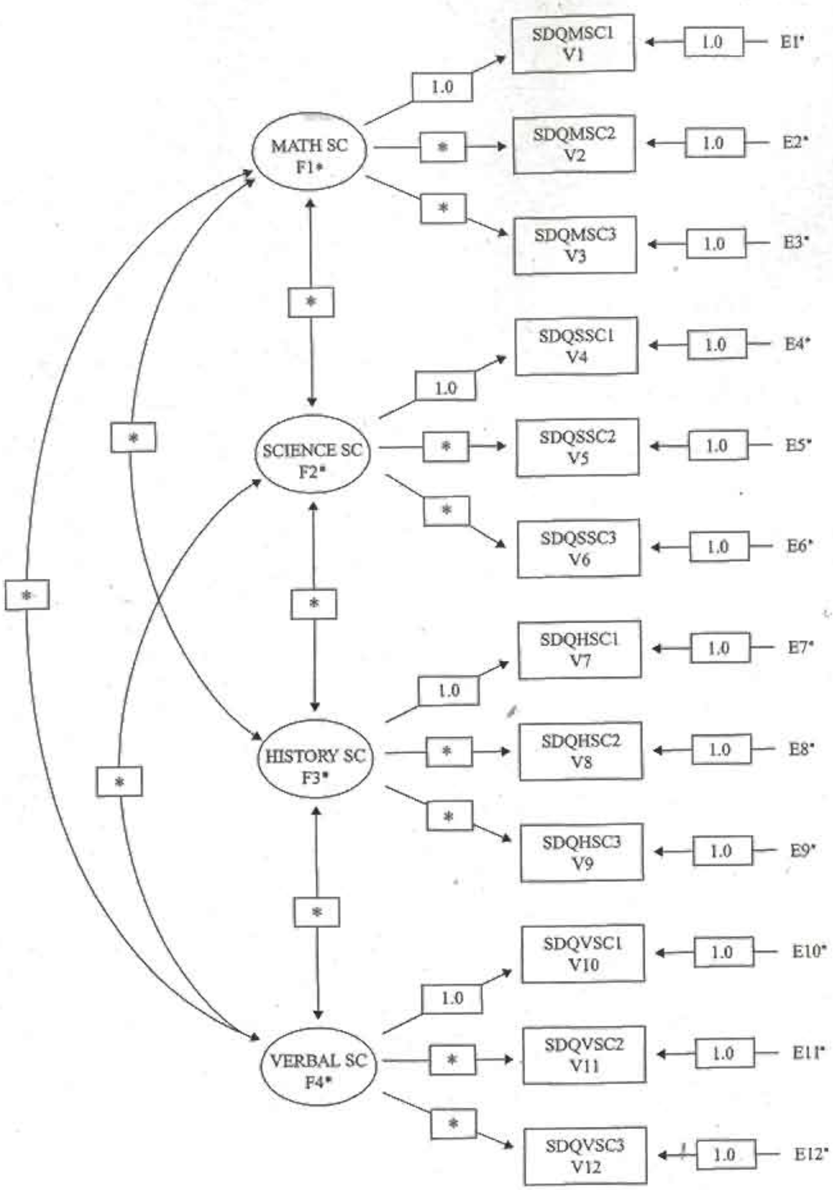

FIG. 2.1. Hypothesized first-order CFA model with EQS notation and assigned start values.

portions of the same input files appear later in the chapter, they are presented in their original form as structured by the BUILD_EQS option. Thus, the files presented in this subsection provide an idea of how to make the file more concise if so desired. To derive maximum benefit from this subsection, study each figure while referring to its respective input statement.

First-Order CFA Model

```
/TITLE
 Example 1st-order CFA Input File (Figure 2.1)
 Initial Model
/SPECIFICATIONS
DA='C:\EQS61\FILES\CFASC.DAT'; FO=(12F1.0);
CASE=361; VAR=12; ME=ML; MA=RAW;
/LABELS
 V1=SDQMSC1;   V2=SDQMSC2;   V3=SDQMSC3;
 V4=SDQSSC1;   V5=SDQSSC2;   V6=SDQSSC3;
 V7=SDQHSC1;   V8=SDQHSC3;   V9=SDQHSC3;
 V10=SDQVSC1;  V11=SDQVSC2;  V12=SDQVSC3;
 F1=MSC;  F2=SSC;  F3=HSC;  F4=VSC;
/EQUATIONS
 V1 =  F1 + E1;
 V2 = *F1 + E2;
 V3 = *F1 + E3;
          V4 =  F2 + E4;
          V5 = *F2 + E5;
          V6 = *F2 + E6;
                   V7 =  F3 + E7;
                   V8 = *F3 + E8;
                   V9 = *F3 + E9;
                           V10 =  F4 + E10;
                           V11 = *F4 + E11;
                           V12 = *F4 + E12;
/VARIANCES
 F1 to F4 = *;
 E1 to E12 = *;
/COVARIANCES
 F1 to F4 = *;
/END
```

Although this input file appears to be fairly straightforward, three features are worthy of elaboration. First, the "to" convention has been used in both the /VARIANCES and /COVARIANCES paragraphs, thereby indicating that the variances of all the independent variables (F1-F4) are to be freely estimated as well as their covariances. Second, recall that an independent variable can have either its path or its variance estimated but not both. Thus, because we are only interested in the variances of the error terms, note that in the /EQUATIONS paragraph, their related beta weights are automatically assigned by the program a fixed value of 1.0. Although these values are not shown in the input file (see E1-E12), they are visible in Fig. 2.1. Finally, note that the first measurement indicator for each factor (V1, V4, V7, V10) has been specified as fixed (i.e., there is no asterisk next to these parameters). As shown in Fig. 2.1, these parameters are constrained to equal a value of 1.0; EQS automatically assigns this value to these fixed parameters—thus, the value of 1.0 does not appear in the input file. Two points need to be made in this regard: (a) the value assigned to these parameters need not be 1.0—although any numeric may be assigned to these parameters, a value of 1.0 has typically been the assignment of choice; and (b) the constrained parameter need not be limited to the first indicator variable; any one of a congeneric set[3] of parameters may be chosen. Beyond these technical notations, however, the most important point regarding the fixed factor loadings is that they address the issue of *model identification* (also termed *statistical identification*), a topic to which we turn shortly.

Critical to knowing whether a model is statistically identified is understanding the number of estimable parameters in the model. Thus, one extremely important caveat in working with structural equation models is to always tally the number of freely estimated parameters prior to running the analyses. As a prerequisite to the discussion of model identification, then, let's count the number of parameters to be estimated for the model portrayed in Fig. 2.1. Reviewing the figure, we ascertain that there are 12 regression coefficients (factor loadings) and 16 variances (12 error variances, 4 factor variances), and 6 factor covariances. The 1's assigned to one of each set of regression-path parameters represent a fixed value of 1.00; as such, these parameters are *not* to be estimated. In total, then, there are 30 parameters to be estimated for the CFA model depicted in Fig. 2.1. A brief discussion of the important concept of model identification follows.

The Concept of Model Identification. Model identification is a complex topic that is difficult to explain in nontechnical terms. Although a thorough

[3] A set of measures is said to be "congeneric" if each measure in the set purports to assess the same construct, except for errors of measurement (Jöreskog, 1971a). For example, as shown in Fig. 2.1, SDQMSC1, SDQMSC2, and SDQMSC3 all serve as measures of math SC (self-concept); therefore, they represent a congeneric set of indicator variables.

explanation of the identification principle exceeds the scope of this book, it is not critical to the reader's understanding and use of the book. Nonetheless, because some insight into the general concept of the identification issue undoubtedly helps to better understand why, for example, particular parameters are specified as having certain fixed values, what follows is a brief nonmathematical explanation of the basic idea underlying this concept. Essentially, only the so-called "t-rule" is addressed, one of several tests associated with identification. The reader is encouraged to consult the following texts for a more comprehensive treatment of the topic: Bollen, 1989a; Kline, 1998; Long, 1983a, 1983b; and Saris & Stronkhorst, 1984. I also recommend a clear and readable description of this topic in a book chapter by MacCallum (1995).

In broad terms, the issue of identification focuses on whether there is a unique set of parameters consistent with the data. This question bears directly on the transposition of the variance–covariance matrix of observed variables (the data) into the structural parameters of the model under study. If a unique solution for the values of the structural parameters can be found, the model is considered identified. As a consequence, the parameters are considered estimable and the model therefore testable. On the other hand if a model cannot be identified, it indicates that the parameters are subject to arbitrariness, thereby implying that different parameter values define the same model. Such being the case, attainment of consistent estimates for all parameters is not possible and, thus, the model cannot be evaluated empirically. By way of a simple example, the process would be conceptually akin to trying to determine unique values for X and Y when the only information available is that $X + Y = 15$. Generalizing this example to covariance-structure analysis, the model identification issue focuses on the extent to which a unique set of values can be inferred for the unknown parameters from a given covariance matrix of analyzed variables reproduced by the model.

Structural models may be *just-identified*, *overidentified*, or *underidentified*. A just-identified model is one in which there is a one-to-one correspondence between the data and the structural parameters. That is, the number of data variances and covariances equals the number of parameters to be estimated. However, despite the capability of the model to yield a unique solution for all parameters, the just-identified model is not scientifically interesting because it has no degrees of freedom and therefore can never be rejected. An overidentified model is one in which the number of estimable parameters is less than the number of data points (i.e., variances and covariances of the observed variables). This situation results in positive degrees of freedom that allow for rejection of the model, thereby rendering it of scientific use. The aim in SEM is to specify a model such that it meets the criterion of overidentification. Finally, an underidentified model is one in which the number of parameters to be estimated exceeds the number of variances and covariances (i.e., data points). As such, the model contains insufficient information (from the input data) to attain a determinate solution of parameter estimation; that is, an infinite number of solutions are possible for an underidentified model.

Reviewing the CFA model in Fig. 2.1, let's determine how many data points there are to work with (i.e., How much information do we have with respect to our data?). As noted previously, these data points constitute the variances and covariances of the observed variables; with p variables, there are $p(p + 1)/2$ such elements. Given that there are 12 observed variables, this means there are $12(12 + 1)/2 = 78$ data points. Before this discussion of identification, we determined a total of 30 unknown parameters. Thus, with 78 data points and 30 parameters to be estimated, we have an overidentified model with 48 degrees of freedom.

However, it is important to note that the specification of an overidentified model is a necessary but not sufficient condition to resolve the identification problem. Indeed, the imposition of constraints on particular parameters can sometimes be beneficial in helping the researcher to attain an overidentified model. Examples of such a constraint are illustrated in chapters 5 and 10 with the application of a second-order CFA model.

Linked to the issue of identification is the requirement that every latent variable must have its scale determined. This requirement arises because latent variables are unobserved and, therefore, have no definite metric scale. This requirement can be accomplished in one of two ways. The first approach is tied to specification of the measurement model, whereby the unmeasured latent variable is mapped onto its related observed indicator variable. This scaling requisite is satisfied by constraining to some nonzero value (typically 1.0), one factor loading parameter in each congeneric set of loadings. This constraint holds for both independent and dependent latent variables. In reviewing Fig. 2.1, this means that for one of the three regression paths leading from each self-concept (SC) factor to a set of observed indicators, some fixed value should be specified; this fixed parameter is termed a *reference* variable. With respect to the model in Fig. 2.1, for example, the scale was established by constraining to a value of 1.0, the first parameter in each set of observed variables.

With a better idea of important aspects of the specification of a CFA model in general, specification using the EQS program in particular, and the basic notions associated with model identification, we continue on our walk-through of the two remaining models reviewed in this chapter.

Second-Order CFA Model

```
/TITLE
 Example 2nd-order CFA Input File (Figure 2.2)
 Initial Model

/EQUATIONS
 V1 =  F1 + E1;
 V2 = *F1 + E2;
 V3 = *F1 + E3;
           V4 =  F2 + E4;
           V5 = *F2 + E5;
           V6 = *F2 + E6;
                     V7 =  F3 + E7;
                     V8 = *F3 + E8;
                     V9 = *F3 + E9;
                               V10 =  F4 + E10;
                               V11 = *F4 + E11;
                               V12 = *F4 + E12;
 F1 = *F5 + D1;
   F2 = *F5 + D2;
     F3 = *F5 + D3;
       F4 = *F5 + D4;
/VARIANCES
 F5 = 1.0;
 D1 to D4 = *;
 E1 to E12 = *;
/END
```

This model supports the notion that math, science, history, and verbal self-concepts (SCs) are caused by some higher order global academic self-concept construct. Thus, although the first-order structure of this model remains basically the same as in Fig. 2.1, four important features must be noted. First, in contrast to the first-order CFA example in which all equations in the /EQUATIONS paragraph represented regression paths among all observed variables and their related underlying factors, the equations specified for this second-order model also include regression paths among factors. Specifically, these parameters represent the impact of F5 (an independent variable) on F1, F2, F3, and F4. Second, because the estimation of all higher order factor loadings are typically of interest in second-order models, the variance of the single higher order factor, F5, has been constrained to 1.0, as specified in the /VARIANCES paragraph.[4] This constraint, of course, addresses

[4]Alternatively, if the variance of F5 had been of interest, one of the higher order loadings would have to be constrained to 1.0 (or another nonzero value).

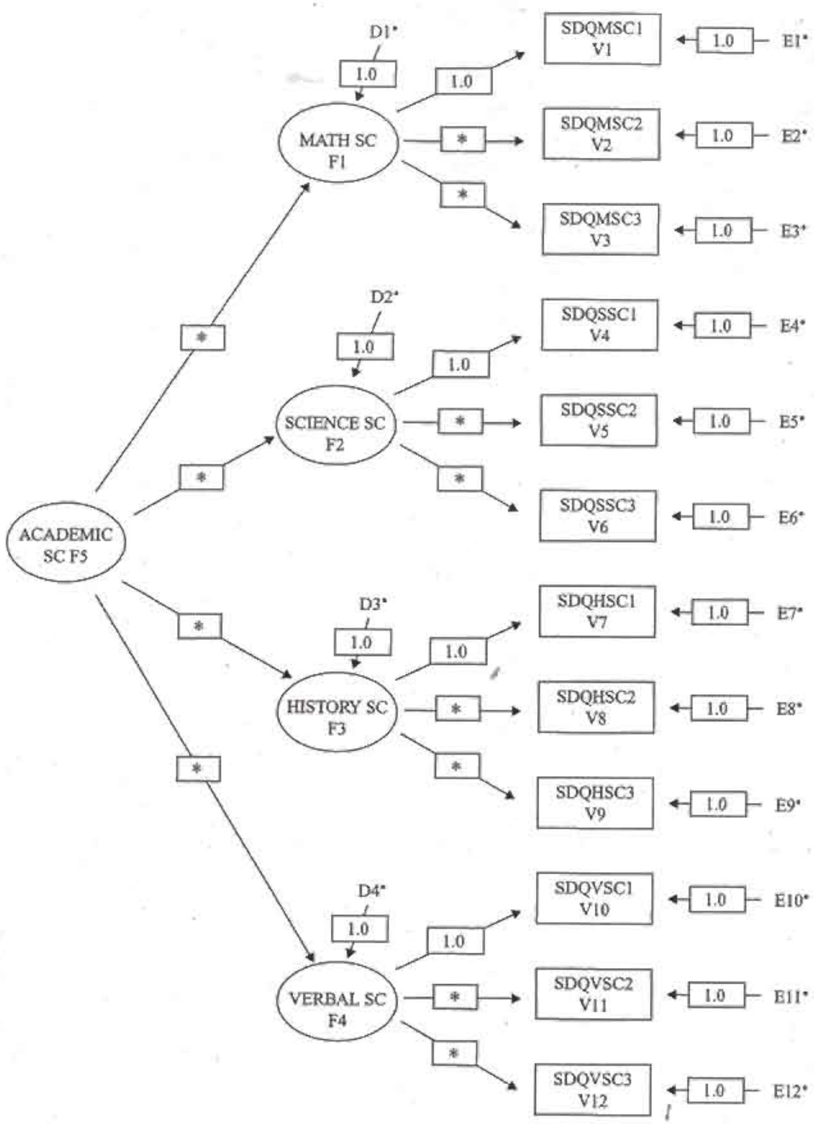

FIG. 2.2. Hypothesized second-order CFA model with EQS notation and assigned start values.

the issue of model identification discussed earlier. Third, given that academic SC is hypothesized to cause each of the four first-order factors, F1 through F4 now represent dependent variables in the Bentler–Weeks sense. As such, math, science, history, and verbal SCs are hypothesized as being predicted from academic SC but with some degree of error, which is captured by the disturbance term associated with each of these factors. Thus, in the /VARIANCES paragraph, note also that the variances of the disturbances (the D's) are designated as freely estimated. Relatedly, their paths are automatically constrained to 1.0 by the program, as shown in Fig. 2.2. Finally, in second-order models, any covariance among the first-order factors is presumed to be explained by the higher order factor(s). Accordingly, note the absence of double-headed arrows linking the four first-order factors in the path diagram and the absence of a /COVARIANCES paragraph in the input file.

Full Structural Equation Model

```
/TITLE
 Example Full SEM Input File (Figure 2.3)
 Initial Model
/EQUATIONS
 V1 =  F1 + E1;
 V2 = *F1 + E2;
 V3 = *F1 + E3;
           V4 =  F2 + E4;
           V5 = *F2 + E5;
           V6 = *F2 + E6;
                     V7 =  F3 + E7;
                     V8 = *F3 + E8;
                     V9 = *F3 + E9;
                               V10 =  F4 + E10;
                               V11 = *F4 + E11;
F3 = *F1 + *F2 + D3;
F4 = *F3 + D4;
/VARIANCES
F1 = *; F2 = *;
D3 = *; D4 = *;
E1 to E11 = *;
/COVARIANCES
F1, F2 = *;
/END
```

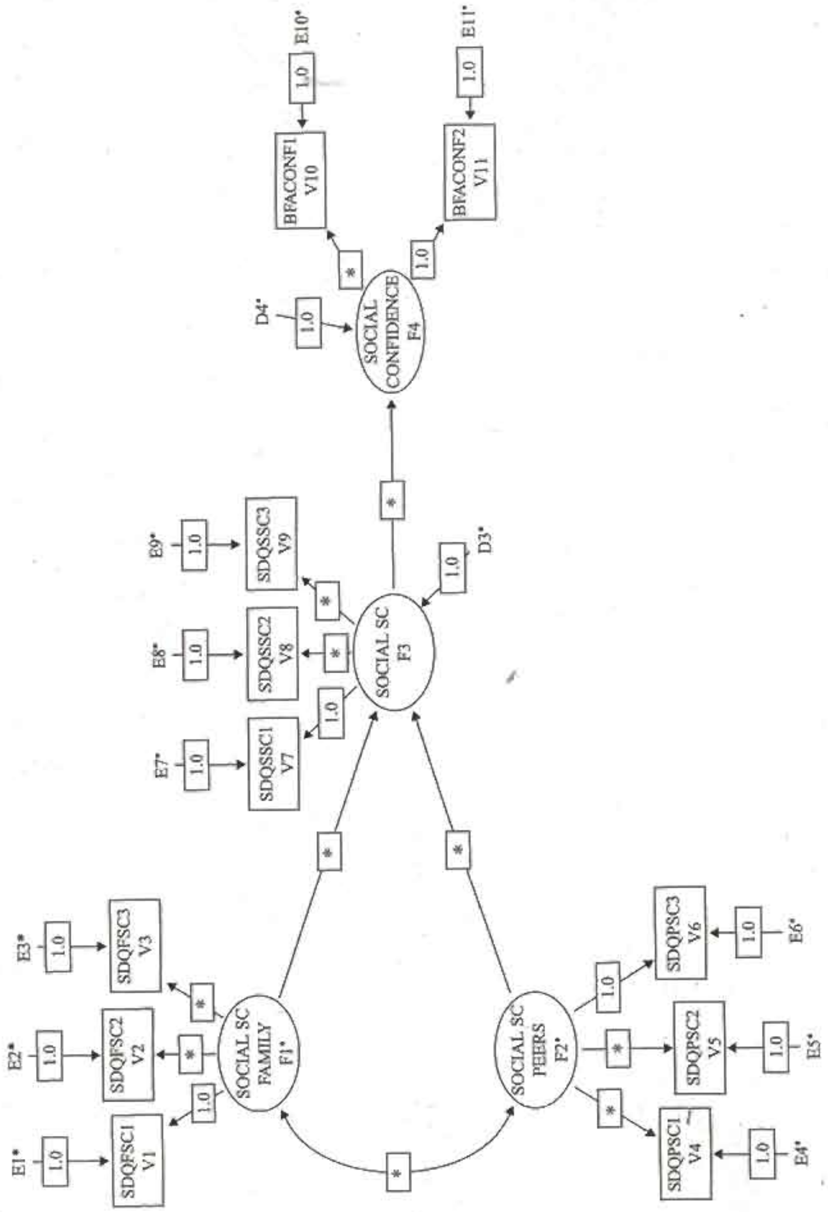

FIG. 2.3. Hypothesized full structural equation model with EQS notation and assigned start values.

In reviewing both the path diagram and the input file for this full SEM model, note three particular specifications. First, of the four factors composing this model, only F1 and F2 are independent variables (in the Bentler–Weeks sense); all others are dependent variables in the model. Consequently, only the variances for F1 and F2, as well as their covariances, can be estimated as specified in the input file. Second, as with the second-order model shown in Fig. 2.2, again, the regression equations involve sets of factors. In the present case, these equations are specified only for F3 and F4 because each is explained by other factors in the model. Third, given the dependent variable status of F3 and F4 in the model, note that the variance of neither is specified in the /VARIANCES paragraph; relatedly, note the specification of variances associated with their respective disturbances, D3 and D4, respectively. Finally, although Factor 3 appears to operate as both an independent and a dependent variable, this is not so. Once a variable is defined as a dependent variable in a model, it maintains that designation throughout all analyses bearing on the hypothesized model.

CREATING THE EQS INPUT FILE

With this understanding of the basic components of an EQS input file, let's proceed to the three methods used to build these files: (a) manually; (b) interactively, using the BUILD_EQS option; and (c) graphically, using the DIAGRAMMER option.

Building an Input File Manually

EQS input files can be structured manually using any word-processing program (e.g., Word™, Wordperfect™). However, because EQS files cannot interpret the hidden control-text characters inserted by these programs, any invisible symbols must be stripped from the file before an EQS run can be executed. This task is easily accomplished by saving the file in ASCII format.[5] Files created using the manual approach should be saved using the .eqs extension (e.g., bdigender.eqs). Indeed, this method was used in structuring the three input files presented previously. Although the manual approach has served us well over the past 30 years, as the modus operandi for building SEM input files, these last 10 years have rapidly catapulted us into a "New Age" of interactive computing. Keeping in sync with this important trend, EQS holds a unique position among SEM programs in that it offers users two state-of-the-art methods for building input files interactively. We now turn to these new program-building methods.

[5]Alternatively, use the BUILD_EQS command line editor to accomplish this task.