

Chapter 19

The CALIS Procedure

Chapter Table of Contents

OVERVIEW	437
Structural Equation Models	440
GETTING STARTED	448
SYNTAX	454
PROC CALIS Statement	456
COSAN Model Statement	479
MATRIX Statement	481
RAM Model Statement	484
LINEQS Model Statement	488
STD Statement	490
COV Statement	491
FACTOR Model Statement	493
BOUNDS Statement	495
LINCON Statement	496
NLINCON Statement	497
NLOPTIONS Statement	498
PARAMETERS Statement	511
STRUCTEQ Statement	511
VARNAMES Statement	512
BY Statement	513
VAR Statement	513
PARTIAL Statement	514
FREQ Statement	514
WEIGHT Statement	514
SAS Program Statements	514
DETAILS	517
Input Data Sets	517
Output Data Sets	521
Missing Values	531
Estimation Criteria	531
Relationships among Estimation Criteria	534
Testing Rank Deficiency	534
Approximate Standard Errors	535

Assessment of Fit	536
Measures of Multivariate Kurtosis	544
Initial Estimates	547
Automatic Variable Selection	548
Exogenous Manifest Variables	549
Use of Optimization Techniques	551
Modification Indices	560
Constrained Estimation Using Program Code	561
Counting the Degrees of Freedom	563
Computational Problems	564
Displayed Output	568
ODS Table Names	574
EXAMPLES	577
Example 19.1 Path Analysis: Stability of Alienation	577
Example 19.2 Simultaneous Equations with Intercept	596
Example 19.3 Second-Order Confirmatory Factor Analysis	603
Example 19.4 Linear Relations Among Factor Loadings	611
Example 19.5 Ordinal Relations Among Factor Loadings	617
Example 19.6 Longitudinal Factor Analysis	623
REFERENCES	625

Chapter 19

The CALIS Procedure

Overview

Structural equation modeling using covariance analysis is an important statistical tool in economics and behavioral sciences. Structural equations express relationships among several variables that can be either directly observed variables (manifest variables) or unobserved hypothetical variables (latent variables). For an introduction to latent variable models, refer to Loehlin (1987), Bollen (1989b), Everitt (1984), or Long (1983); and for manifest variables, refer to Fuller (1987).

In structural models, as opposed to functional models, all variables are taken to be random rather than having fixed levels. For maximum likelihood (default) and generalized least-squares estimation in PROC CALIS, the random variables are assumed to have an approximately multivariate normal distribution. Nonnormality, especially high kurtosis, can produce poor estimates and grossly incorrect standard errors and hypothesis tests, even in large samples. Consequently, the assumption of normality is much more important than in models with nonstochastic exogenous variables. You should remove outliers and consider transformations of nonnormal variables before using PROC CALIS with maximum likelihood (default) or generalized least-squares estimation. If the number of observations is sufficiently large, Browne's asymptotically distribution-free (ADF) estimation method can be used.

You can use the CALIS procedure to estimate parameters and test hypotheses for constrained and unconstrained problems in

- multiple and multivariate linear regression
- linear measurement-error models
- path analysis and causal modeling
- simultaneous equation models with reciprocal causation
- exploratory and confirmatory factor analysis of any order
- canonical correlation
- a wide variety of other (non)linear latent variable models

The parameters are estimated using the criteria of

- unweighted least squares (ULS)
- generalized least squares (GLS, with optional weight matrix input)
- maximum likelihood (ML, for multivariate normal data)

- weighted least squares (WLS, ADF, with optional weight matrix input)
- diagonally weighted least squares (DWLS, with optional weight matrix input)

The default weight matrix for generalized least-squares estimation is the sample covariance or correlation matrix. The default weight matrix for weighted least-squares estimation is an estimate of the asymptotic covariance matrix of the sample covariance or correlation matrix. In this case, weighted least-squares estimation is equivalent to Browne's (1982, 1984) asymptotic distribution-free estimation. The default weight matrix for diagonally weighted least-squares estimation is an estimate of the asymptotic variances of the input sample covariance or correlation matrix. You can also use an input data set to specify the weight matrix in GLS, WLS, and DWLS estimation.

You can specify the model in several ways:

- You can do a constrained (confirmatory) first-order factor analysis or component analysis using the FACTOR statement.
- You can specify simple path models using an easily formulated list-type RAM statement similar to that originally developed by J. McArdle (McArdle and McDonald 1984).
- If you have a set of structural equations to describe the model, you can use an equation-type LINEQS statement similar to that originally developed by P. Bentler (1985).
- You can analyze a broad family of matrix models using COSAN and MATRIX statements that are similar to the COSAN program of R. McDonald and C. Fraser (McDonald 1978, 1980). It enables you to specify complex matrix models including nonlinear equation models and higher-order factor models.

You can specify linear and nonlinear equality and inequality constraints on the parameters with several different statements, depending on the type of input. Lagrange multiplier test indices are computed for simple constant and equality parameter constraints and for active boundary constraints. General equality and inequality constraints can be formulated using program statements. For more information, see the "SAS Program Statements" section on page 514.

PROC CALIS offers a variety of methods for the automatic generation of initial values for the optimization process:

- two-stage least-squares estimation
- instrumental variable factor analysis
- approximate factor analysis
- ordinary least-squares estimation
- McDonald's (McDonald and Hartmann 1992) method

In many common applications, these initial values prevent computational problems and save computer time.

Because numerical problems can occur in the (non)linearly constrained optimization process, the CALIS procedure offers several optimization algorithms:

- Levenberg-Marquardt algorithm (Moré, 1978)
- trust region algorithm (Gay 1983)
- Newton-Raphson algorithm with line search
- ridge-stabilized Newton-Raphson algorithm
- various quasi-Newton and dual quasi-Newton algorithms: Broyden-Fletcher-Goldfarb-Shanno and Davidon-Fletcher-Powell, including a sequential quadratic programming algorithm for processing nonlinear equality and inequality constraints
- various conjugate gradient algorithms: automatic restart algorithm of Powell (1977), Fletcher-Reeves, Polak-Ribiere, and conjugate descent algorithm of Fletcher (1980)

The quasi-Newton and conjugate gradient algorithms can be modified by several line-search methods. All of the optimization techniques can impose simple boundary and general linear constraints on the parameters. Only the dual quasi-Newton algorithm is able to impose general nonlinear equality and inequality constraints.

The procedure creates an OUTRAM= output data set that completely describes the model (except for program statements) and also contains parameter estimates. This data set can be used as input for another execution of PROC CALIS. Small model changes can be made by editing this data set, so you can exploit the old parameter estimates as starting values in a subsequent analysis. An OUTEST= data set contains information on the optimal parameter estimates (parameter estimates, gradient, Hessian, projected Hessian and Hessian of Lagrange function for constrained optimization, the information matrix, and standard errors). The OUTEST= data set can be used as an INEST= data set to provide starting values and boundary and linear constraints for the parameters. An OUTSTAT= data set contains residuals and, for exploratory factor analysis, the rotated and unrotated factor loadings.

Automatic variable selection (using only those variables from the input data set that are used in the model specification) is performed in connection with the RAM and LINEQS input statements or when these models are recognized in an input model file. Also in these cases, the covariances of the exogenous manifest variables are recognized as given constants. With the PREDET option, you can display the predetermined pattern of constant and variable elements in the predicted model matrix before the minimization process starts. For more information, see the section “Automatic Variable Selection” on page 548 and the section “Exogenous Manifest Variables” on page 549.

PROC CALIS offers an analysis of linear dependencies in the information matrix (approximate Hessian matrix) that may be helpful in detecting unidentified models. You also can save the information matrix and the approximate covariance matrix of the parameter estimates (inverse of the information matrix), together with parameter estimates, gradient, and approximate standard errors, in an output data set for further analysis.

PROC CALIS does not provide the analysis of multiple samples with different sample size or a generalized algorithm for missing values in the data. However, the analysis of multiple samples with equal sample size can be performed by the analysis of a moment supermatrix containing the individual moment matrices as block diagonal submatrices.

Structural Equation Models

The Generalized COSAN Model

PROC CALIS can analyze matrix models of the form

$$\mathbf{C} = \mathbf{F}_1 \mathbf{P}_1 \mathbf{F}_1' + \cdots + \mathbf{F}_m \mathbf{P}_m \mathbf{F}_m'$$

where \mathbf{C} is a symmetric correlation or covariance matrix, each matrix \mathbf{F}_k , $k = 1, \dots, m$, is the product of $n(k)$ matrices $\mathbf{F}_{k_1}, \dots, \mathbf{F}_{k_{n(k)}}$, and each matrix \mathbf{P}_k is symmetric, that is,

$$\mathbf{F}_k = \mathbf{F}_{k_1} \cdots \mathbf{F}_{k_{n(k)}} \quad \text{and} \quad \mathbf{P}_k = \mathbf{P}_k', \quad k = 1, \dots, m$$

The matrices \mathbf{F}_{k_j} and \mathbf{P}_k in the model are parameterized by the matrices \mathbf{G}_{k_j} and \mathbf{Q}_k

$$\mathbf{F}_{k_j} = \begin{cases} \mathbf{G}_{k_j} \\ \mathbf{G}_{k_j}^{-1} \\ (\mathbf{I} - \mathbf{G}_{k_j})^{-1} \end{cases} \quad j = 1, \dots, n(k) \quad \text{and} \quad \mathbf{P}_k = \begin{cases} \mathbf{Q}_k \\ \mathbf{Q}_k^{-1} \end{cases}$$

where you can specify the type of matrix desired.

The matrices \mathbf{G}_{k_j} and \mathbf{Q}_k can contain

- constant values
- parameters to be estimated
- values computed from parameters via programming statements

The parameters can be summarized in a parameter vector $\mathbf{X} = (x_1, \dots, x_t)$. For a given covariance or correlation matrix \mathbf{C} , PROC CALIS computes the unweighted least-squares (ULS), generalized least-squares (GLS), maximum likelihood (ML), weighted least-squares (WLS), or diagonally weighted least-squares (DWLS) estimates of the vector \mathbf{X} .

Some Special Cases of the Generalized COSAN Model**Original COSAN (Covariance Structure Analysis) Model (McDonald 1978, 1980)**

Covariance Structure:

$$\mathbf{C} = \mathbf{F}_1 \cdots \mathbf{F}_n \mathbf{P} \mathbf{F}_n' \cdots \mathbf{F}_1'$$

RAM (Reticular Action) Model (McArdle 1980; McArdle and McDonald 1984)

Structural Equation Model:

$$\mathbf{v} = \mathbf{A}\mathbf{v} + \mathbf{u}$$

where \mathbf{A} is a matrix of coefficients, and \mathbf{v} and \mathbf{u} are vectors of random variables. The variables in \mathbf{v} and \mathbf{u} can be manifest or latent variables. The endogenous variables corresponding to the components in \mathbf{v} are expressed as a linear combination of the remaining variables and a residual component in \mathbf{u} with covariance matrix \mathbf{P} .

Covariance Structure:

$$\mathbf{C} = \mathbf{J}(\mathbf{I} - \mathbf{A})^{-1} \mathbf{P} ((\mathbf{I} - \mathbf{A})^{-1})' \mathbf{J}'$$

with selection matrix \mathbf{J} and

$$\mathbf{C} = \mathcal{E}\{\mathbf{J}\mathbf{v}\mathbf{v}'\mathbf{J}'\} \quad \text{and} \quad \mathbf{P} = \mathcal{E}\{\mathbf{u}\mathbf{u}'\}$$

LINEQS (Linear Equations) Model (Bentler and Weeks 1980)

Structural Equation Model:

$$\boldsymbol{\eta} = \boldsymbol{\beta}\boldsymbol{\eta} + \boldsymbol{\gamma}\boldsymbol{\xi}$$

where $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are coefficient matrices, and $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$ are vectors of random variables. The components of $\boldsymbol{\eta}$ correspond to the endogenous variables; the components of $\boldsymbol{\xi}$ correspond to the exogenous variables and to error variables. The variables in $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$ can be manifest or latent variables. The endogenous variables in $\boldsymbol{\eta}$ are expressed as a linear combination of the remaining endogenous variables, of the exogenous variables of $\boldsymbol{\xi}$, and of a residual component in $\boldsymbol{\xi}$. The coefficient matrix $\boldsymbol{\beta}$ describes the relationships among the endogenous variables of $\boldsymbol{\eta}$, and $\mathbf{I} - \boldsymbol{\beta}$ should be nonsingular. The coefficient matrix $\boldsymbol{\gamma}$ describes the relationships between the endogenous variables of $\boldsymbol{\eta}$ and the exogenous and error variables of $\boldsymbol{\xi}$.

Covariance Structure:

$$\mathbf{C} = \mathbf{J}(\mathbf{I} - \mathbf{B})^{-1} \mathbf{\Gamma} \mathbf{\Phi} \mathbf{\Gamma}' ((\mathbf{I} - \mathbf{B})^{-1})' \mathbf{J}'$$

with selection matrix \mathbf{J} , $\mathbf{\Phi} = \mathcal{E}\{\boldsymbol{\xi}\boldsymbol{\xi}'\}$, and

$$\mathbf{B} = \begin{pmatrix} \beta & 0 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{\Gamma} = \begin{pmatrix} \gamma \\ \mathbf{I} \end{pmatrix}$$

Keesling - Wiley - Jöreskog LISREL (Linear Structural Relationship) Model

Structural Equation Model and Measurement Models:

$$\boldsymbol{\eta} = \mathbf{B}\boldsymbol{\eta} + \mathbf{\Gamma}\boldsymbol{\xi} + \boldsymbol{\zeta}, \quad \mathbf{y} = \mathbf{\Lambda}_y\boldsymbol{\eta} + \boldsymbol{\varepsilon}, \quad \mathbf{x} = \mathbf{\Lambda}_x\boldsymbol{\xi} + \boldsymbol{\delta}$$

where $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$ are vectors of latent variables (factors), and \mathbf{x} and \mathbf{y} are vectors of manifest variables. The components of $\boldsymbol{\eta}$ correspond to endogenous latent variables; the components of $\boldsymbol{\xi}$ correspond to exogenous latent variables. The endogenous and exogenous latent variables are connected by a system of linear equations (the structural model) with coefficient matrices \mathbf{B} and $\mathbf{\Gamma}$ and an error vector $\boldsymbol{\zeta}$. It is assumed that matrix $\mathbf{I} - \mathbf{B}$ is nonsingular. The random vectors \mathbf{y} and \mathbf{x} correspond to manifest variables that are related to the latent variables $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$ by two systems of linear equations (the measurement model) with coefficients $\mathbf{\Lambda}_y$ and $\mathbf{\Lambda}_x$ and with measurement errors $\boldsymbol{\varepsilon}$ and $\boldsymbol{\delta}$.

Covariance Structure:

$$\mathbf{C} = \mathbf{J}(\mathbf{I} - \mathbf{A})^{-1} \mathbf{P} ((\mathbf{I} - \mathbf{A})^{-1})' \mathbf{J}'$$

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & \mathbf{\Lambda}_y & 0 \\ 0 & 0 & 0 & \mathbf{\Lambda}_x \\ 0 & 0 & \mathbf{B} & \mathbf{\Gamma} \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{P} = \begin{pmatrix} \boldsymbol{\Theta}_\varepsilon & & & \\ & \boldsymbol{\Theta}_\delta & & \\ & & \boldsymbol{\Psi} & \\ & & & \boldsymbol{\Phi} \end{pmatrix}$$

with selection matrix \mathbf{J} , $\mathbf{\Phi} = \mathcal{E}\{\boldsymbol{\xi}\boldsymbol{\xi}'\}$, $\boldsymbol{\Psi} = \mathcal{E}\{\boldsymbol{\zeta}\boldsymbol{\zeta}'\}$, $\boldsymbol{\Theta}_\delta = \mathcal{E}\{\boldsymbol{\delta}\boldsymbol{\delta}'\}$, and $\boldsymbol{\Theta}_\varepsilon = \mathcal{E}\{\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}'\}$.

Higher-Order Factor Analysis Models

First-order model:

$$\mathbf{C} = \mathbf{F}_1 \mathbf{P}_1 \mathbf{F}_1' + \mathbf{U}_1^2$$

Second-order model:

$$\mathbf{C} = \mathbf{F}_1 \mathbf{F}_2 \mathbf{P}_2 \mathbf{F}_2' \mathbf{F}_1' + \mathbf{F}_1 \mathbf{U}_2^2 \mathbf{F}_1' + \mathbf{U}_1^2$$

First-Order Autoregressive Longitudinal Factor Model

Example of McDonald (1980): $k=3$: Occasions of Measurement; $n=3$: Variables (Tests); $m=2$: Common Factors

$$\begin{aligned} \mathbf{C} &= \mathbf{F}_1 \mathbf{F}_2 \mathbf{F}_3 \mathbf{L} \mathbf{F}_3^{-1} \mathbf{F}_2^{-1} \mathbf{P} (\mathbf{F}_2^{-1})' (\mathbf{F}_3^{-1})' \mathbf{L}' \mathbf{F}_3' \mathbf{F}_2' \mathbf{F}_1' + \mathbf{U}^2 \\ F_1 &= \begin{pmatrix} B_1 & & \\ & B_2 & \\ & & B_3 \end{pmatrix}, \quad F_2 = \begin{pmatrix} I_2 & & \\ & D_2 & \\ & & D_2 \end{pmatrix}, \quad F_3 = \begin{pmatrix} I_2 & & \\ & I_2 & \\ & & D_3 \end{pmatrix} \\ L &= \begin{pmatrix} I_2 & o & o \\ I_2 & I_2 & o \\ I_2 & I_2 & I_2 \end{pmatrix}, \quad P = \begin{pmatrix} I_2 & & \\ & S_2 & \\ & & S_3 \end{pmatrix}, \quad U = \begin{pmatrix} U_{11} & U_{12} & U_{13} \\ U_{21} & U_{22} & U_{23} \\ U_{31} & U_{32} & U_{33} \end{pmatrix} \\ S_2 &= I_2 - D_2^2, \quad S_3 = I_2 - D_3^2 \end{aligned}$$

For more information on this model, see Example 19.6 on page 623.

A Structural Equation Example

This example from Wheaton et al. (1977) illustrates the relationships among the RAM, LINEQS, and LISREL models. Different structural models for these data are in Jöreskog and Sörbom (1985) and in Bentler (1985, p. 28). The data set contains covariances among six (manifest) variables collected from 932 people in rural regions of Illinois:

Variable 1: $V1, y_1$: Anomia 1967
 Variable 2: $V2, y_2$: Powerlessness 1967
 Variable 3: $V3, y_3$: Anomia 1971
 Variable 4: $V4, y_4$: Powerlessness 1971
 Variable 5: $V5, x_1$: Education (years of schooling)
 Variable 6: $V6, x_2$: Duncan's Socioeconomic Index (SEI)

It is assumed that anomia and powerlessness are indicators of an alienation factor and that education and SEI are indicators for a socioeconomic status (SES) factor. Hence, the analysis contains three latent variables:

Variable 7: $F1, \eta_1$: Alienation 1967
 Variable 8: $F2, \eta_2$: Alienation 1971
 Variable 9: $F3, \xi_1$: Socioeconomic Status (SES)

The following path diagram shows the structural model used in Bentler (1985, p. 29) and slightly modified in Jöreskog and Sörbom (1985, p. 56). In this notation for the path diagram, regression coefficients between the variables are indicated as one-headed arrows. Variances and covariances among the variables are indicated as two-headed arrows. Indicating error variances and covariances as two-headed arrows with the same source and destination (McArdle 1988; McDonald 1985) is helpful in transforming the path diagram to RAM model list input for the CALIS procedure.

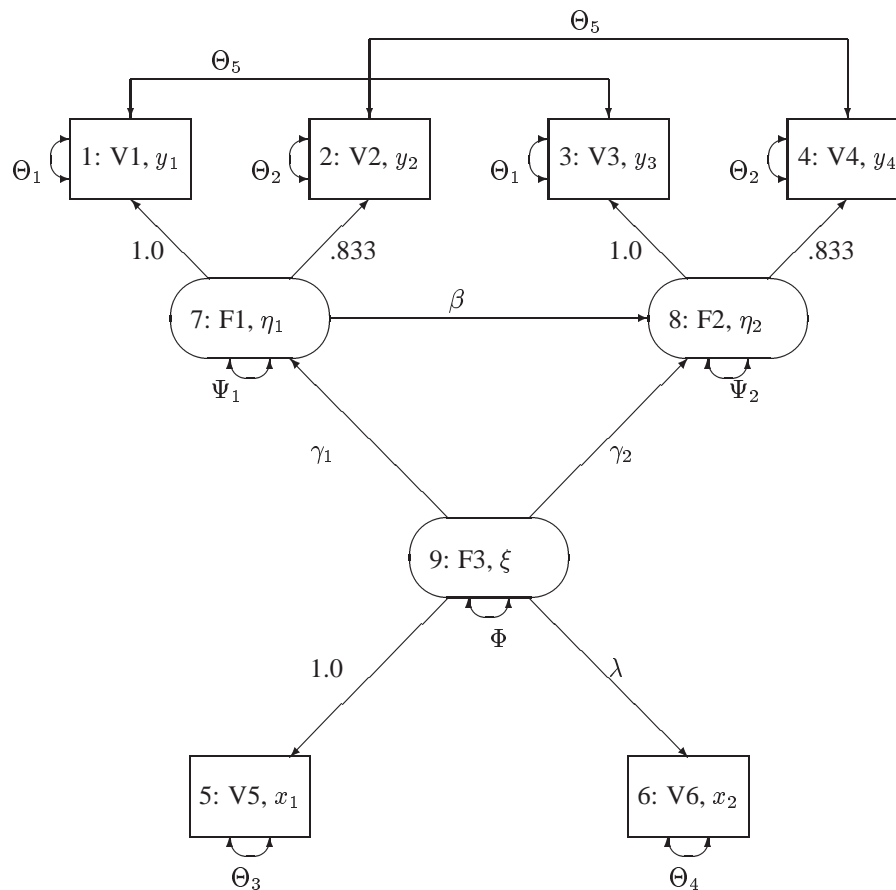


Figure 19.1. Path Diagram of Stability and Alienation Example

Variables in Figure 19.1 are as follows:

- | | |
|-------------|------------------------------------------------|
| Variable 1: | $V1, y_1$: Anomia 1967 |
| Variable 2: | $V2, y_2$: Powerlessness 1967 |
| Variable 3: | $V3, y_3$: Anomia 1971 |
| Variable 4: | $V4, y_4$: Powerlessness 1971 |
| Variable 5: | $V5, x_1$: Education (years of schooling) |
| Variable 6: | $V6, x_2$: Duncan's Socioeconomic Index (SEI) |
| Variable 7: | $F1, \eta_1$: Alienation 1967 |
| Variable 8: | $F2, \eta_2$: Alienation 1971 |
| Variable 9: | $F3, \xi_1$: Socioeconomic Status (SES) |

RAM Model

The vector \mathbf{v} contains the six manifest variables $v_1 = V1, \dots, v_6 = V6$ and the three latent variables $v_7 = F1, v_8 = F2, v_9 = F3$. The vector \mathbf{u} contains the corresponding error variables $u_1 = E1, \dots, u_6 = E6$ and $u_7 = D1, u_8 = D2, u_9 = D3$. The path diagram corresponds to the following set of structural equations of the RAM model:

$$\begin{aligned}
 v_1 &= 1.000v_7 + u_1 \\
 v_2 &= 0.833v_7 + u_2 \\
 v_3 &= 1.000v_8 + u_3 \\
 v_4 &= 0.833v_8 + u_4 \\
 v_5 &= 1.000v_9 + u_5 \\
 v_6 &= \lambda v_9 + u_6 \\
 v_7 &= \gamma_1 v_9 + u_7 \\
 v_8 &= \beta v_7 + \gamma_2 v_9 + u_8 \\
 v_9 &= u_9
 \end{aligned}$$

This gives the matrices \mathbf{A} and \mathbf{P} in the RAM model:

$$\mathbf{A} = \begin{pmatrix} o & o & o & o & o & o & 1.000 & o & o \\ o & o & o & o & o & o & 0.833 & o & o \\ o & o & o & o & o & o & o & 1.000 & o \\ o & o & o & o & o & o & o & 0.833 & o \\ o & o & o & o & o & o & o & o & 1.000 \\ o & o & o & o & o & o & o & o & \lambda \\ o & o & o & o & o & o & o & o & \gamma_1 \\ o & o & o & o & o & o & \beta & o & \gamma_2 \\ o & o & o & o & o & o & o & o & o \end{pmatrix}$$

$$\mathbf{P} = \begin{pmatrix} \theta_1 & o & \theta_5 & o & o & o & o & o & o \\ o & \theta_2 & o & \theta_5 & o & o & o & o & o \\ \theta_5 & o & \theta_1 & o & o & o & o & o & o \\ o & \theta_5 & o & \theta_2 & o & o & o & o & o \\ o & o & o & o & \theta_3 & o & o & o & o \\ o & o & o & o & o & \theta_4 & o & o & o \\ o & o & o & o & o & o & \psi_1 & o & o \\ o & o & o & o & o & o & o & \psi_2 & o \\ o & o & o & o & o & o & o & o & \phi \end{pmatrix}$$

The RAM model input specification of this example for the CALIS procedure is in the “RAM Model Specification” section on page 451.

LINEQS Model

The vector $\boldsymbol{\eta}$ contains the six endogenous manifest variables $V1, \dots, V6$ and the two endogenous latent variables $F1$ and $F2$. The vector $\boldsymbol{\xi}$ contains the exogenous error variables $E1, \dots, E6, D1$, and $D2$ and the exogenous latent variable $F3$. The path diagram corresponds to the following set of structural equations of the LINEQS model:

$$\begin{aligned}
 V1 &= 1.0F1 + E1 \\
 V2 &= .833F1 + E2 \\
 V3 &= 1.0F2 + E3 \\
 V4 &= .833F2 + E4 \\
 V5 &= 1.0F3 + E5 \\
 V6 &= \lambda F3 + E6 \\
 F1 &= \gamma_1 F3 + D1 \\
 F2 &= \beta F1 + \gamma_2 F3 + D2
 \end{aligned}$$

This gives the matrices $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ and $\boldsymbol{\Phi}$ in the LINEQS model:

$$\boldsymbol{\beta} = \begin{pmatrix} o & o & o & o & o & o & 1. & o \\ o & o & o & o & o & o & .833 & o \\ o & o & o & o & o & o & o & 1. \\ o & o & o & o & o & o & o & .833 \\ o & o & o & o & o & o & o & o \\ o & o & o & o & o & o & o & o \\ o & o & o & o & o & o & o & o \\ o & o & o & o & o & o & \beta & o \end{pmatrix}, \quad \boldsymbol{\gamma} = \begin{pmatrix} 1 & o & o & o & o & o & o & o & o \\ o & 1 & o & o & o & o & o & o & o \\ o & o & 1 & o & o & o & o & o & o \\ o & o & o & 1 & o & o & o & o & o \\ o & o & o & o & 1 & o & o & o & 1. \\ o & o & o & o & o & 1 & o & o & \lambda \\ o & o & o & o & o & o & 1 & o & \gamma_1 \\ o & o & o & o & o & o & o & 1 & \gamma_2 \end{pmatrix}$$

$$\boldsymbol{\Phi} = \begin{pmatrix} \theta_1 & o & \theta_5 & o & o & o & o & o & o \\ o & \theta_2 & o & \theta_5 & o & o & o & o & o \\ \theta_5 & o & \theta_1 & o & o & o & o & o & o \\ o & \theta_5 & o & \theta_2 & o & o & o & o & o \\ o & o & o & o & \theta_3 & o & o & o & o \\ o & o & o & o & o & \theta_4 & o & o & o \\ o & o & o & o & o & o & \psi_1 & o & o \\ o & o & o & o & o & o & o & \psi_2 & o \\ o & o & o & o & o & o & o & o & \phi \end{pmatrix}$$

The LINEQS model input specification of this example for the CALIS procedure is in the section “LINEQS Model Specification” on page 450.

LISREL Model

The vector \mathbf{y} contains the four endogenous manifest variables $y_1 = V1, \dots, y_4 = V4$, and the vector \mathbf{x} contains the exogenous manifest variables $x_1 = V5$ and $x_2 = V6$. The vector $\boldsymbol{\varepsilon}$ contains the error variables $\varepsilon_1 = E1, \dots, \varepsilon_4 = E4$ corresponding to \mathbf{y} , and the vector $\boldsymbol{\delta}$ contains the error variables $\delta_1 = E5$ and $\delta_2 = E6$ corresponding to \mathbf{x} . The vector $\boldsymbol{\eta}$ contains the endogenous latent variables (factors) $\eta_1 = F1$ and $\eta_2 = F2$, while the vector $\boldsymbol{\xi}$ contains the exogenous latent variable (factor) $\xi_1 = F3$. The vector $\boldsymbol{\zeta}$ contains the errors $\zeta_1 = D1$ and $\zeta_2 = D2$ in the equations (disturbance terms) corresponding to $\boldsymbol{\eta}$. The path diagram corresponds to the following set of structural equations of the LISREL model:

$$\begin{aligned} y_1 &= 1.0\eta_1 + \varepsilon_1 \\ y_2 &= .833\eta_1 + \varepsilon_2 \\ y_3 &= 1.0\eta_2 + \varepsilon_3 \\ y_4 &= .833\eta_2 + \varepsilon_4 \\ x_1 &= 1.0\xi_1 + \delta_1 \\ x_2 &= \lambda\xi_1 + \delta_2 \\ \eta_1 &= \gamma_1\xi_1 + \zeta_1 \\ \eta_2 &= \beta\eta_1 + \gamma_2\xi_1 + \zeta_2 \end{aligned}$$

This gives the matrices $\mathbf{\Lambda}_y$, $\mathbf{\Lambda}_x$, \mathbf{B} , $\mathbf{\Gamma}$, and $\mathbf{\Phi}$ in the LISREL model:

$$\mathbf{\Lambda}_y = \begin{pmatrix} 1. & o \\ .833 & o \\ o & 1. \\ o & .833 \end{pmatrix}, \mathbf{\Lambda}_x = \begin{pmatrix} 1. \\ \lambda \end{pmatrix}, \mathbf{B} = \begin{pmatrix} o & o \\ \beta & o \end{pmatrix}, \mathbf{\Gamma} = \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix}$$

$$\mathbf{\Theta}_{\varepsilon}^2 = \begin{pmatrix} \theta_1 & o & \theta_5 & o \\ o & \theta_2 & o & \theta_5 \\ \theta_5 & o & \theta_1 & o \\ o & \theta_5 & o & \theta_2 \end{pmatrix}, \mathbf{\Theta}_{\delta}^2 = \begin{pmatrix} \theta_3 & o \\ \theta_4 & o \end{pmatrix}, \mathbf{\Psi} = \begin{pmatrix} \psi_1 & o \\ o & \psi_2 \end{pmatrix}, \mathbf{\Phi} = (\phi)$$

The CALIS procedure does not provide a LISREL model input specification. However, any model that can be specified by the LISREL model can also be specified by using the COSAN, LINEQS, or RAM model specifications in PROC CALIS.

Getting Started

There are four sets of statements available in the CALIS procedure to specify a model. Since a LISREL analysis can be performed easily by using a RAM, COSAN, or LINEQS statement, there is no specific LISREL input form available in the CALIS procedure.

For COSAN-style input, you can specify the following statements:

COSAN *analysis model in matrix notation ;*
MATRIX *definition of matrix elements ;*
VARNAMES *names of additional variables ;*
BOUNDS *boundary constraints ;*
PARAMETERS *parameter names from program statements ;*

For linear equations input, you can specify the following statements:

LINEQS *analysis model in equations notation ;*
STD *variance pattern ;*
COV *covariance pattern ;*
BOUNDS *boundary constraints ;*
PARAMETERS *parameter names from program statements ;*

For RAM-style input, you can specify the following statements:

RAM *analysis model in list notation ;*
VARNAMES *names of latent and error variables ;*
BOUNDS *boundary constraints ;*
PARAMETERS *parameter names from program statements ;*

For (confirmatory) factor analysis input, you can specify the following statements:

FACTOR *options ;*
MATRIX *definition of matrix elements ;*
VARNAMES *names of latent and residual variables ;*
BOUNDS *boundary constraints ;*
PARAMETERS *parameter names from program statements ;*

The model can also be obtained from an INRAM= data set, which is usually a version of an OUTRAM= data set produced by a previous PROC CALIS analysis (and possibly modified).

If no INRAM= data set is specified, you must use one of the four statements that defines the input form of the analysis model: COSAN, RAM, LINEQS, or FACTOR.

COSAN Model Specification

You specify the model for a generalized COSAN analysis with a COSAN statement and one or more MATRIX statements. The COSAN statement determines the name, dimension, and type (identity, diagonal, symmetric, upper, lower, general, inverse, and so forth) of each matrix in the model. You can specify the values of the constant elements in each matrix and give names and initial values to the elements that are to be estimated as parameters or functions of parameters using MATRIX statements. The resulting displayed output is in matrix form.

The following statements define the structural model of the alienation example as a COSAN model:

```
Cosan J(9, Ide) * A(9, Gen, Imi) * P(9, Sym);
Matrix A
      [ ,7] = 1. .833 5 * 0. Beta (.5) ,
      [ ,8] = 2 * 0. 1. .833 ,
      [ ,9] = 4 * 0. 1. Lamb Gam1-Gam2 (.5 2 * -.5);
Matrix P
      [1,1] = The1-The2 The1-The4 (6 * 3.) ,
      [7,7] = Psi1-Psi2 Phi (2 * 4. 6.) ,
      [3,1] = The5 (.2) ,
      [4,2] = The5 (.2) ;
```

The matrix model specified in the COSAN statement is the RAM model

$$\mathbf{C} = \mathbf{J}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{P}((\mathbf{I} - \mathbf{A})^{-1})'\mathbf{J}'$$

with selection matrix \mathbf{J} and

$$\mathbf{C} = \mathcal{E}\{\mathbf{J}\mathbf{v}\mathbf{v}'\mathbf{J}'\}, \quad \mathbf{P} = \mathcal{E}\{\mathbf{u}\mathbf{u}'\}$$

The COSAN statement must contain only the matrices up to the central matrix \mathbf{P} because of the symmetry of each matrix term in a COSAN model. Each matrix name is followed by one to three arguments in parentheses. The first argument is the number of columns. The second and third arguments are optional, and they specify the form of the matrix. The selection matrix \mathbf{J} in the RAM model is specified by the 6×9 identity (IDE) (sub)matrix \mathbf{J} because the first six variables in vector \mathbf{v} correspond to the six manifest variables in the data set. The 9×9 parameter matrix \mathbf{A} has a general (GEN) form and is used as $(\mathbf{I} - \mathbf{A})^{-1}$ in the analysis, as indicated by the identity-minus-inverse (IMI) argument. The central 9×9 matrix \mathbf{P} is specified as a symmetric (SYM) matrix.

The MATRIX statement for matrix \mathbf{A} specifies the values in columns 7, 8, and 9, which correspond to the three latent variables $F1$, $F2$, and $F3$, in accordance with the RAM model. The other columns of \mathbf{A} are assumed to be zero. The initial values for the parameter elements in \mathbf{A} are chosen as in the path diagram to be

$$\lambda = \beta = .5, \quad \gamma_1 = \gamma_2 = -.5$$

In accordance with matrix **P** of the RAM model and the path model, the nine diagonal elements of matrix **P** are parameters with initial values

$$\theta_1 = \theta_2 = \theta_3 = \theta_4 = 3, \quad \psi_1 = \psi_2 = 4, \quad \phi = 6$$

There are also two off-diagonal elements in each triangle of **P** that are constrained to be equal, and they have an initial value of 0.2.

See the section “COSAN Model Statement” on page 479 for more information about the COSAN statement.

LINEQS Model Specification

You can also describe the model by a set of linear equations combined with variance and covariance specifications, using notation similar to that originally developed by P. Bentler for his EQS program. The displayed output can be in either equation form or matrix form.

The following statements define the structural model of the alienation example as a LINEQS model:

```

Lineqs
  V1 =          F1                      + E1,
  V2 =      .833 F1                      + E2,
  V3 =          F2                      + E3,
  V4 =      .833 F2                      + E4,
  V5 =          F3                      + E5,
  V6 = Lamb (.5) F3                      + E6,
  F1 = Gaml(-.5) F3                      + D1,
  F2 = Beta (.5) F1 + Gam2(-.5) F3 + D2;

Std
  E1-E6 = The1-The2 The1-The4 (6 * 3.),
  D1-D2 = Psi1-Psi2 (2 * 4.),
  F3     = Phi (6.) ;

Cov
  E1 E3 = The5 (.2),
  E4 E2 = The5 (.2);

```

The LINEQS statement shows the equations in the section “LINEQS Model” on page 446, except that in this case the coefficients to be estimated can be followed (optionally) by the initial value to use in the optimization process. If you do not specify initial values for the parameters in a LINEQS statement, PROC CALIS tries to assign these values automatically. The endogenous variables used on the left side can be manifest variables (with names that must be defined by the input data set) or latent variables (which must have names starting with F). The variables used on the right side can be manifest variables, latent variables (with names that must start with an F), or error variables (which must have names starting with an E or D). Commas separate the equations. The coefficients to be estimated are indicated by names. If no name is used, the coefficient is constant, either equal to a specified number or, if no number is used, equal to 1. The VAR statement in Bentler’s notation is replaced here by the STD statement, because the VAR statement in PROC CALIS defines the subset of manifest variables in the data set to be analyzed. The variable names used

in the STD or COV statement must be exogenous (that is, they should not occur on the left side of any equation). The STD and COV statements define the diagonal and off-diagonal elements in the Φ matrix. The parameter specifications in the STD and COV statements are separated by commas. Using k variable names on the left of an equal sign in a COV statement means that the parameter list on the right side refers to all $k(k - 1)/2$ distinct variable pairs in the Φ matrix. Identical coefficient names indicate parameters constrained to be equal. You can also use prefix names to specify those parameters for which you do not need a precise name in any parameter constraint.

See the section “LINEQS Model Statement” on page 488 for more information about the precise syntax rules for a LINEQS statement.

RAM Model Specification

The RAM model allows a path diagram to be transcribed into a RAM statement in list form. The displayed output from the RAM statement is in matrix or list form.

The following statement defines the structural model of the alienation example as a RAM model:

```
Ram
  1   1   7   1.           ,
  1   2   7   .833        ,
  1   3   8   1.           ,
  1   4   8   .833        ,
  1   5   9   1.           ,
  1   6   9   .5   Lamb   ,
  1   7   9   -.5   Gam1   ,
  1   8   7   .5   Beta    ,
  1   8   9   -.5   Gam2    ,
  2   1   1   3.   The1     ,
  2   2   2   3.   The2     ,
  2   3   3   3.   The1     ,
  2   4   4   3.   The2     ,
  2   5   5   3.   The3     ,
  2   6   6   3.   The4     ,
  2   1   3   .2   The5     ,
  2   2   4   .2   The5     ,
  2   7   7   4.   Psi1     ,
  2   8   8   4.   Psi2     ,
  2   9   9   6.   Phi ;
```

You must assign numbers to the nodes in the path diagram. In the path diagram of Figure 19.1, the boxes corresponding to the six manifest variables $V1, \dots, V6$ are assigned the number of the variable in the covariance matrix (1, \dots , 6); the circles corresponding to the three latent variables $F1, F2$, and $F3$ are given the numbers 7, 8, and 9. The path diagram contains 20 paths between the nine nodes; nine of the paths are one-headed arrows and eleven are two-headed arrows.

The RAM statement contains a list of items separated by commas. Each item corresponds to an arrow in the path diagram. The first entry in each item is the number of arrow heads (matrix number), the second entry shows where the arrow points to

(row number), the third entry shows where the arrow comes from (column number), the fourth entry gives the (initial) value of the coefficient, and the fifth entry assigns a name if the path represents a parameter rather than a constant. If you specify the fifth entry as a parameter name, then the fourth list entry can be omitted, since PROC CALIS tries to assign an initial value to this parameter.

See the section “RAM Model Statement” on page 484 for more information about the RAM statement.

FACTOR Model Specification

You can specify the FACTOR statement to compute factor loadings \mathbf{F} and unique variances \mathbf{U} of an exploratory or confirmatory first-order factor (or component) analysis. By default, the factor correlation matrix \mathbf{P} is an identity matrix.

$$\mathbf{C} = \mathbf{F}\mathbf{F}' + \mathbf{U}, \quad \mathbf{U} = \text{diag}$$

For a first-order confirmatory factor analysis, you can use MATRIX statements to define elements in the matrices \mathbf{F} , \mathbf{P} , and \mathbf{U} of the more general model

$$\mathbf{C} = \mathbf{F}\mathbf{P}\mathbf{F}' + \mathbf{U}, \quad \mathbf{P} = \mathbf{P}', \quad \mathbf{U} = \text{diag}$$

To perform a component analysis, specify the COMPONENT option to constrain the matrix \mathbf{U} to a zero matrix; that is, the model is replaced by

$$\mathbf{C} = \mathbf{F}\mathbf{F}'$$

Note that the rank of $\mathbf{F}\mathbf{F}'$ is equal to the number m of components in \mathbf{F} , and if m is smaller than the number of variables in the moment matrix \mathbf{C} , the matrix of predicted model values is singular and maximum likelihood estimates for \mathbf{F} cannot be computed. You should compute ULS estimates in this case.

The HEYWOOD option constrains the diagonal elements of \mathbf{U} to be nonnegative; that is, the model is replaced by

$$\mathbf{C} = \mathbf{F}\mathbf{F}' + \mathbf{U}^2, \quad \mathbf{U} = \text{diag}$$

If the factor loadings are unconstrained, they can be orthogonally rotated by one of the following methods:

- principal axes rotation
- quartimax
- varimax
- equamax
- parsimax

The most common approach to factor analysis consists of two steps:

1. Obtain estimates for factor loadings and unique variances.
2. Apply an orthogonal or oblique rotation method.

Most programs of factor analysis do not provide standard errors for the rotated factor loadings. PROC CALIS enables you to specify general linear and nonlinear equality and inequality constraints using the LINCON and NLINCON statements. You can specify the NLINCON statement to estimate orthogonal or oblique rotated factor loadings; refer to Browne and Du Toit (1992).

For default (exploratory) factor analysis, PROC CALIS computes initial estimates. If you use a MATRIX statement together with a FACTOR model specification, initial values are generally computed by McDonald's (McDonald and Hartmann 1992) method or are set by the START= option. See the section "FACTOR Model Statement" on page 493 and Example 19.3 for more information about the FACTOR statement.

Constrained Estimation

- Simple equality constraints, $x_i = c_i$, $c_i = \text{const}$, and $x_i = x_j$, can be defined in each model by specifying constants or using the same name for parameters constrained to be equal.
- BOUNDS statement: You can specify boundary constraints, $l_i \leq x_i \leq u_i$, $l_i, u_i = \text{const}$, with the BOUNDS statement for the COSAN, LINEQS, and RAM models and in connection with an INRAM= data set. There may be serious convergence problems if negative values appear in the diagonal locations (variances) of the central model matrices during the minimization process. You can use the BOUNDS statement to constrain these parameters to have nonnegative values.
- LINCON statement: You can specify general linear equality and inequality constraints of the parameter estimates with the LINCON statement or by using an INEST= data set. The variables listed in the LINCON statements must be (a subset of) the model parameters. All optimization methods can be used with linear constraints.
- NLINCON statement: You can specify general nonlinear equality and inequality constraints of the parameter estimates with the NLINCON statement. The syntax of the NLINCON statement is almost the same as that for the BOUNDS statement with the exception that the BOUNDS statement can contain only names of the model parameters. However, the variables listed in the NLINCON statement can be defined by program statements. Only the quasi-Newton optimization method can be used when there are nonlinear constraints.
- Reparameterizing the Model: Complex linear equality and inequality constraints can be defined by means of program statements similar to those used in the DATA step. In this case, some of the parameters x_i are not elements of the matrices \mathbf{G}_{kj} and \mathbf{Q}_k but are instead defined in a PARAMETERS statement. Elements of the model matrices can then be computed by program statements as functions of parameters in the PARAMETERS statement. This approach is

similar to the classical COSAN program of R. McDonald, implemented by C. Fraser (McDonald 1978, 1980). One advantage of the CALIS procedure is that you need not supply code for the derivatives of the specified functions. The analytic derivatives of the user-written functions are computed automatically by PROC CALIS. The specified functions must be continuous and have continuous first-order partial derivatives. See the “SAS Program Statements” section on page 514 and the “Constrained Estimation Using Program Code” section on page 561 for more information about imposing linear and nonlinear restrictions on parameters by using program statements.

Although much effort has been made to implement reliable and numerically stable optimization methods, no practical algorithm exists that can always find the global optimum of a nonlinear function, especially when there are nonlinear constraints.

Syntax

```
PROC CALIS < options > ;
  COSAN matrix model ;
  MATRIX matrix elements ;
  VARNAMES variables ;
  LINEQS model equations ;
  STD variance pattern ;
  COV covariance pattern ;
  RAM model list ;
  VARNAMES variables ;
  FACTOR < options > ;
  MATRIX matrix elements ;
  VARNAMES variables ;
  BOUNDS boundary constraints ;
  BY variables ;
  FREQ variable ;
  LINCON linear constraints ;
  NLINCON nonlinear constraints ;
  NLOPTIONS optimization options ;
  PARAMETERS parameters ;
  PARTIAL variables ;
  STRUCTEQ variables ;
  VAR variables ;
  WEIGHT variable ;
  program statements
```

- If no INRAM= data set is specified, one of the four statements that defines the input form of the analysis model, COSAN, LINEQS, RAM, or FACTOR, must be used.
- The MATRIX statement can be used multiple times for the same or different matrices along with a COSAN or FACTOR statement. If the MATRIX

statement is used multiple times for the same matrix, later definitions override earlier ones.

- The STD and COV statements can be used only with the LINEQS model statement.
- You can formulate a generalized COSAN model using a COSAN statement. MATRIX statements can be used to define the elements of a matrix used in the COSAN statement. The input notation resembles the COSAN program of R. McDonald and C. Fraser (McDonald 1978, 1980).
- The RAM statement uses a simple list input that is especially suitable for describing J. McArdle's RAM analysis model (McArdle 1980, McArdle and McDonald 1984) for causal and path analysis problems.
- The LINEQS statement formulates the analysis model by means of a system of linear equations similar to P. Bentler's (1989) EQS program notation. The STD and COV statements can be used to define the variances and covariances corresponding to elements of matrix Φ in the LINEQS model.
- A FACTOR statement can be used to compute a first-order exploratory or confirmatory factor (or component) analysis. The analysis of a simple exploratory factor analysis model performed by PROC CALIS is not as efficient as one performed by the FACTOR procedure. The CALIS procedure is designed for more general structural problems, and it needs significantly more computation time for a simple unrestricted factor or component analysis than does PROC FACTOR.
- You can add program statements to impose linear or nonlinear constraints on the parameters if you specify the model by means of a COSAN, LINEQS, or RAM statement. The PARAMETERS statement defines additional parameters that are needed as independent variables in your program code and that belong to the set of parameters to be estimated. Variable names used in the program code should differ from the preceding statement names. The code should respect the syntax rules of SAS statements usually used in the DATA step. See the "SAS Program Statements" section on page 514 for more information.
- The BOUNDS statement can be used to specify simple lower and upper boundary constraints for the parameters.
- You can specify general linear equality and inequality constraints with the LINCON statement (or via an INEST= data set). The NLINCON statement can be used to specify general nonlinear equality and inequality constraints by referring to nonlinear functions defined by program statements.
- The VAR, PARTIAL, WEIGHT, FREQ, and BY statements can be used in the same way as in other procedures, for example, the FACTOR or PRINCOMP procedure. You can select a subset of the input variables to analyze with the VAR statement. The PARTIAL statement defines a set of input variables that are chosen as partial variables for the analysis of a matrix of partial correlations or covariances. The BY statement specifies groups in which separate covariance structure analyses are performed.

PROC CALIS Statement

PROC CALIS < options > ;

This statement invokes the procedure. The options available with the PROC CALIS statement are summarized in Table 19.1 and discussed in the following six sections.

Table 19.1. PROC CALIS Statement Options

Data Set Options	Short Description
DATA=	input data set
INEST=	input initial values, constraints
INRAM=	input model
INWGT=	input weight matrix
OUTEST=	covariance matrix of estimates
OUTJAC	Jacobian into OUTEST= data set
OUTRAM=	output model
OUTSTAT=	output statistic
OUTWGT=	output weight matrix

Data Processing	Short Description
AUGMENT	analyzes augmented moment matrix
COVARIANCE	analyzes covariance matrix
EDF=	defines nobis by number error df
NOBS=	defines number of observations nobis
NOINT	analyzes uncorrected moments
RDF=	defines nobis by number regression df
RIDGE	specifies ridge factor for moment matrix
UCORR	analyzes uncorrected CORR matrix
UCOV	analyzes uncorrected COV matrix
VARDEF=	specifies variance divisor

Estimation Methods	Short Description
METHOD=	estimation method
ASYCOV=	formula of asymptotic covariances
DFREDUCE=	reduces degrees of freedom
G4=	algorithm for STDERR
NODIAG	excludes diagonal elements from fit
WPENALTY=	penalty weight to fit correlations
WRIDGE=	ridge factor for weight matrix

Optimization Techniques	Short Description
TECHNIQUE=	minimization method
UPDATE=	update technique
LINESEARCH=	line-search method
FCONV=	function convergence criterion

Table 19.1. (continued)

Optimization Techniques	Short Description
GCONV=	gradient convergence criterion
INSTEP=	initial step length (RADIUS=, SALPHA=)
LSPRECISION=	line-search precision (SPRECISION=)
MAXFUNC=	max number function calls
MAXITER=	max number iterations

Displayed Output Options	Short Description
KURTOSIS	compute and display kurtosis
MODIFICATION	modification indices
NOMOD	no modification indices
NOPRINT	suppresses the displayed output
PALL	all displayed output (ALL)
PCORR	analyzed and estimated moment matrix
PCOVES	covariance matrix of estimates
PDETERM	determination coefficients
PESTIM	parameter estimates
PINITIAL	pattern and initial values
PJACPAT	displays structure of variable and constant elements of the Jacobian matrix
PLATCOV	latent variable covariances, scores
PREDDET	displays predetermined moment matrix
PRIMAT	displays output in matrix form
PRINT	adds default displayed output
PRIVEC	displays output in vector form
PSHORT	reduces default output (SHORT)
PSUMMARY	displays only fit summary (SUMMARY)
PWEIGHT	weight matrix
RESIDUAL=	residual matrix and distribution
SIMPLE	univariate statistics
STDERR	standard errors
NOSTDERR	computes no standard errors
TOTEFF	displays total and indirect effects

Miscellaneous Options	Short Description
ALPHAECV=	probability Browne & Cudeck ECV
ALPHARMS=	probability Steiger & Lind RMSEA
BIASKUR	biased skewness and kurtosis
DEMPHAS=	emphasizes diagonal entries
FDCODE	uses numeric derivatives for code
HESSALG=	algorithm for Hessian
NOADJDF	no adjustment of df for active constraints
RANDOM=	randomly generated initial values
SINGULAR=	singularity criterion
ASINGULAR=	absolute singularity information matrix

Table 19.1. (continued)

Miscellaneous Options	Short Description
COVSING=	singularity tolerance of information matrix
MSINGULAR=	relative M singularity of information matrix
VSINGULAR=	relative V singularity of information matrix
SLMW=	probability limit for Wald test
START=	constant initial values

Data Set Options**DATA=SAS-data-set**

specifies an input data set that can be an ordinary SAS data set or a specially structured TYPE=CORR, TYPE=COV, TYPE=UCORR, TYPE=UCOV, TYPE=SSCP, or TYPE=FACTOR SAS data set, as described in the section “Input Data Sets” on page 517. If the DATA= option is omitted, the most recently created SAS data set is used.

INEST | INVAR | ESTDATA=SAS-data-set

specifies an input data set that contains initial estimates for the parameters used in the optimization process and can also contain boundary and general linear constraints on the parameters. If the model did not change too much, you can specify an OUTEST= data set from a previous PROC CALIS analysis. The initial estimates are taken from the values of the PARMS observation.

INRAM=SAS-data-set

specifies an input data set that contains in RAM list form all information needed to specify an analysis model. The INRAM= data set is described in the section “Input Data Sets” on page 517. Typically, this input data set is an OUTRAM= data set (possibly modified) from a previous PROC CALIS analysis. If you use an INRAM= data set to specify the analysis model, you cannot use the model specification statements COSAN, MATRIX, RAM, LINEQS, STD, COV, FACTOR, or VARNAMES, but you can use the BOUNDS and PARAMETERS statements and program statements. If the INRAM= option is omitted, you must define the analysis model with a COSAN, RAM, LINEQS, or FACTOR statement.

INWGT=SAS-data-set

specifies an input data set that contains the weight matrix **W** used in generalized least-squares (GLS), weighted least-squares (WLS, ADF), or diagonally weighted least-squares (DWLS) estimation. If the weight matrix **W** defined by an INWGT= data set is not positive definite, it can be ridged using the WRIDGE= option. See the section “Estimation Criteria” on page 531 for more information. If no INWGT= data set is specified, default settings for the weight matrices are used in the estimation process. The INWGT= data set is described in the section “Input Data Sets” on page 517. Typically, this input data set is an OUTWGT= data set from a previous PROC CALIS analysis.

OUTEST | OUTVAR=SAS-data-set

creates an output data set containing the parameter estimates, their gradient, Hessian matrix, and boundary and linear constraints. For METHOD=ML, METHOD=GLS, and METHOD=WLS, the OUTEST= data set also contains the information matrix,

the approximate covariance matrix of the parameter estimates ((generalized) inverse of information matrix), and approximate standard errors. If linear or nonlinear equality or active inequality constraints are present, the Lagrange multiplier estimates of the active constraints, the projected Hessian, and the Hessian of the Lagrange function are written to the data set. The OUTEST= data set also contains the Jacobian if the OUTJAC option is used.

The OUTEST= data set is described in the section “OUTEST= SAS-data-set” on page 521. If you want to create a permanent SAS data set, you must specify a two-level name. Refer to the chapter titled “SAS Data Files” in *SAS Language Reference: Concepts* for more information on permanent data sets.

OUTJAC

writes the Jacobian matrix, if it has been computed, to the OUTEST= data set. This is useful when the information and Jacobian matrices need to be computed for other analyses.

OUTSTAT=SAS-data-set

creates an output data set containing the BY group variables, the analyzed covariance or correlation matrices, and the predicted and residual covariance or correlation matrices of the analysis. You can specify the correlation or covariance matrix in an OUTSTAT= data set as an input DATA= data set in a subsequent analysis by PROC CALIS. The OUTSTAT= data set is described in the section “OUTSTAT= SAS-data-set” on page 528. If the model contains latent variables, this data set also contains the predicted covariances between latent and manifest variables and the latent variables scores regression coefficients (see the PLATCOV option on page 473). If the FACTOR statement is used, the OUTSTAT= data set also contains the rotated and unrotated factor loadings, the unique variances, the matrix of factor correlations, the transformation matrix of the rotation, and the matrix of standardized factor loadings.

You can specify the latent variable score regression coefficients with PROC SCORE to compute factor scores.

If you want to create a permanent SAS data set, you must specify a two-level name. Refer to the chapter titled “SAS Data Files” in *SAS Language Reference: Concepts* for more information on permanent data sets.

OUTRAM=SAS-data-set

creates an output data set containing the model information for the analysis, the parameter estimates, and their standard errors. An OUTRAM= data set can be used as an input INRAM= data set in a subsequent analysis by PROC CALIS. The OUTRAM= data set also contains a set of fit indices; it is described in more detail in the section “OUTRAM= SAS-data-set” on page 525. If you want to create a permanent SAS data set, you must specify a two-level name. Refer to the chapter titled “SAS Data Files” in *SAS Language Reference: Concepts* for more information on permanent data sets.

OUTWGT=SAS-data-set

creates an output data set containing the weight matrix **W** used in the estimation process. You cannot create an OUTWGT= data set with an unweighted least-squares or maximum likelihood estimation. The fit function in GLS, WLS (ADF), and DWLS

estimation contain the inverse of the (Cholesky factor of the) weight matrix **W** written in the OUTWGT= data set. The OUTWGT= data set contains the weight matrix on which the WRIDGE= and the WPENALTY= options are applied. An OUTWGT= data set can be used as an input INWGT= data set in a subsequent analysis by PROC CALIS. The OUTWGT= data set is described in the section “OUTWGT= SAS-data-set” on page 530. If you want to create a permanent SAS data set, you must specify a two-level name. Refer to the chapter titled “SAS Data Files” in *SAS Language Reference: Concepts* for more information on permanent data sets.

Data Processing Options

AUGMENT | AUG

analyzes the augmented correlation or covariance matrix. Using the AUG option is equivalent to specifying UCORR (NOINT but not COV) or UCOV (NOINT and COV) for a data set that is augmented by an intercept variable INTERCEPT that has constant values equal to 1. The variable INTERCEP can be used instead of the default INTERCEPT only if you specify the SAS option OPTIONS VALIDVAR-NAME=V6. The dimension of an augmented matrix is one higher than that of the corresponding correlation or covariance matrix. The AUGMENT option is effective only if the data set does not contain a variable called INTERCEPT and if you specify the UCOV, UCORR, or NOINT option.

Caution: The INTERCEPT variable is included in the moment matrix as the variable with number $n + 1$. Using the RAM model statement assumes that the first n variable numbers correspond to the n manifest variables in the input data set. Therefore, specifying the AUGMENT option assumes that the numbers of the latent variables used in the RAM or path model have to start with number $n + 2$.

COVARIANCE | COV

analyzes the covariance matrix instead of the correlation matrix. By default, PROC CALIS (like the FACTOR procedure) analyzes a correlation matrix. If the DATA= input data set is a valid TYPE=CORR data set (containing a correlation matrix and standard deviations), using the COV option means that the covariance matrix is computed and analyzed.

DFE | EDF= n

makes the effective number of observations $n + i$, where i is 0 if the NOINT, UCORR, or UCOV option is specified without the AUGMENT option or where i is 1 otherwise. You can also use the NOBS= option to specify the number of observations.

DFR | RDF= n

makes the effective number of observations the actual number of observations minus the RDF= value. The degree of freedom for the intercept should not be included in the RDF= option. If you use PROC CALIS to compute a regression model, you can specify RDF= *number-of-regressor-variables* to get approximate standard errors equal to those computed by PROC REG.

NOBS= *nobs*

specifies the number of observations. If the DATA= input data set is a raw data set, *nobs* is defined by default to be the number of observations in the raw data set. The NOBS= and EDF= options override this default definition. You can use the

RDF= option to modify the *nobs* specification. If the DATA= input data set contains a covariance, correlation, or scalar product matrix, you can specify the number of observations either by using the NOBS=, EDF=, and RDF= options in the PROC CALIS statement or by including a `_TYPE_='N'` observation in the DATA= input data set.

NOINT

specifies that no intercept be used in computing covariances and correlations; that is, covariances or correlations are not corrected for the mean. You can specify this option (or UCOV or UCORR) to analyze mean structures in an uncorrected moment matrix, that is, to compute intercepts in systems of structured linear equations (see Example 19.2). The term NOINT is misleading in this case because an uncorrected covariance or correlation matrix is analyzed containing a constant (intercept) variable that is used in the analysis model. The degrees of freedom used in the variance divisor (specified by the VARDEF= option) and some of the assessment of the fit function (see the section “Assessment of Fit” on page 536) depend on whether an intercept variable is included in the model (the intercept is used in computing the corrected covariance or correlation matrix or is used as a variable in the uncorrected covariance or correlation matrix to estimate mean structures) or not included (an uncorrected covariance or correlation matrix is used that does not contain a constant variable).

RIDGE<=*r*>

defines a ridge factor *r* for the diagonal of the moment matrix **S** that is analyzed. The matrix **S** is transformed to

$$\mathbf{S} \longrightarrow \tilde{\mathbf{S}} = \mathbf{S} + r(\text{diag}(\mathbf{S}))$$

If you do not specify *r* in the RIDGE option, PROC CALIS tries to ridge the moment matrix **S** so that the smallest eigenvalue is about 10^{-3} .

Caution: The moment matrix in the OUTSTAT= output data set does not contain the ridged diagonal.

UCORR

analyzes the uncorrected correlation matrix instead of the correlation matrix corrected for the mean. Using the UCORR option is equivalent to specifying the NOINT option but not the COV option.

UCOV

analyzes the uncorrected covariance matrix instead of the covariance matrix corrected for the mean. Using the UCOV option is equivalent to specifying both the COV and NOINT options. You can specify this option to analyze mean structures in an uncorrected covariance matrix, that is, to compute intercepts in systems of linear structural equations (see Example 19.2).

VARDEF= DF | N | WDF | WEIGHT | WGT

specifies the divisor used in the calculation of covariances and standard deviations. The default value is VARDEF=DF. The values and associated divisors are displayed in the following table, where *i* = 0 if the NOINT option is used and *i* = 1 otherwise and where *k* is the number of partial variables specified in the PARTIAL statement. Using an intercept variable in a mean structure analysis, by specifying the AUG-

MENT option, includes the intercept variable in the analysis. In this case, $i = 1$. When a WEIGHT statement is used, w_j is the value of the WEIGHT variable in the j th observation, and the summation is performed only over observations with positive weight.

Value	Description	Divisor
DF	degrees of freedom	$N - k - i$
N	number of observations	N
WDF	sum of weights DF	$\sum_j^N w_j - k - i$
WEIGHT WGT	sum of weights	$\sum_j^N w_j$

Estimation Methods

The default estimation method is maximum likelihood (METHOD=ML), assuming a multivariate normal distribution of the observed variables. The two-stage estimation methods METHOD=LSML, METHOD=LSGLS, METHOD=LSWLS, and METHOD=LSDWLS first compute unweighted least-squares estimates of the model parameters and their residuals. Afterward, these estimates are used as initial values for the optimization process to compute maximum likelihood, generalized least-squares, weighted least-squares, or diagonally weighted least-squares parameter estimates. You can do the same thing by using an OUTRAM= data set with least-squares estimates as an INRAM= data set for a further analysis to obtain the second set of parameter estimates. This strategy is also discussed in the section “Use of Optimization Techniques” on page 551. For more details, see the “Estimation Criteria” section on page 531.

METHOD | MET=name

specifies the method of parameter estimation. The default is METHOD=ML. Valid values for *name* are as follows:

ML M MAX	performs normal-theory maximum likelihood parameter estimation. The ML method requires a nonsingular covariance or correlation matrix.
GLS G	performs generalized least-squares parameter estimation. If no INWGT= data set is specified, the GLS method uses the inverse sample covariance or correlation matrix as weight matrix W . Therefore, METHOD=GLS requires a nonsingular covariance or correlation matrix.
WLS W ADF	performs weighted least-squares parameter estimation. If no INWGT= data set is specified, the WLS method uses the inverse matrix of estimated asymptotic covariances of the sample covariance or correlation matrix as the weight matrix W . In this case, the WLS estimation method is equivalent to Browne’s (1982, 1984) asymptotically distribution-free estimation. The WLS method requires a nonsingular weight matrix.
DWLS D	performs diagonally weighted least-squares parameter estimation. If no INWGT= data set is specified, the DWLS method uses the inverse diagonal matrix of asymptotic

	variances of the input sample covariance or correlation matrix as the weight matrix W . The DWLS method requires a nonsingular diagonal weight matrix.
ULS LS U	performs unweighted least-squares parameter estimation.
LSML LSM LSMAX	performs unweighted least-squares followed by normal-theory maximum likelihood parameter estimation.
LSGLS LSG	performs unweighted least-squares followed by generalized least-squares parameter estimation.
LSWLS LSW LSADF	performs unweighted least-squares followed by weighted least-squares parameter estimation.
LSDWLS LSD	performs unweighted least-squares followed by diagonally weighted least-squares parameter estimation.
NONE NO	uses no estimation method. This option is suitable for checking the validity of the input information and for displaying the model matrices and initial values.

ASYCOV | ASC=name

specifies the formula for asymptotic covariances used in the weight matrix **W** for WLS and DWLS estimation. The ASYCOV option is effective only if METHOD=WLS or METHOD=DWLS and no INWGT=input data set is specified. The following formulas are implemented:

BIASED:	Browne's (1984) formula (3.4) biased asymptotic covariance estimates; the resulting weight matrix is at least positive semidefinite. This is the default for analyzing a covariance matrix.
UNBIASED:	Browne's (1984) formula (3.8) asymptotic covariance estimates corrected for bias; the resulting weight matrix can be indefinite (that is, can have negative eigenvalues), especially for small N .
CORR:	Browne and Shapiro's (1986) formula (3.2) (identical to DeLeeuw's (1983) formulas (2,3,4)) the asymptotic variances of the diagonal elements are set to the reciprocal of the value r specified by the WPENALTY= option (default: $r = 100$). This formula is the default for analyzing a correlation matrix.

Caution: Using the WLS and DWLS methods with the ASYCOV=CORR option means that you are fitting a correlation (rather than a covariance) structure. Since the fixed diagonal of a correlation matrix for some models does not contribute to the model's degrees of freedom, you can specify the DFREDUCE= i option to reduce the degrees of freedom by the number of manifest variables used in the model. See the section "Counting the Degrees of Freedom" on page 563 for more information.

DFREDUCE | DFRED=*i*

reduces the degrees of freedom of the χ^2 test by i . In general, the number of degrees of freedom is the number of elements of the lower triangle of the predicted model matrix \mathbf{C} , $n(n+1)/2$, minus the number of parameters, t . If the NODIAG option is used, the number of degrees of freedom is additionally reduced by n . Because negative values of i are allowed, you can also increase the number of degrees of freedom by using this option. If the DFREDUCE= or NODIAG option is used in a correlation structure analysis, PROC CALIS does not additionally reduce the degrees of freedom by the number of constant elements in the diagonal of the predicted model matrix, which is otherwise done automatically. See the section “Counting the Degrees of Freedom” on page 563 for more information.

G4=*i*

specifies the algorithm to compute the approximate covariance matrix of parameter estimates used for computing the approximate standard errors and modification indices when the information matrix is singular. If the number of parameters t used in the model you analyze is smaller than the value of i , the time-expensive Moore-Penrose (G4) inverse of the singular information matrix is computed by eigenvalue decomposition. Otherwise, an inexpensive pseudo (G1) inverse is computed by sweeping. By default, $i = 60$. For more details, see the section “Estimation Criteria” on page 531.

NODIAG | NODI

omits the diagonal elements of the analyzed correlation or covariance matrix from the fit function. This option is useful only for special models with constant error variables. The NODIAG option does not allow fitting those parameters that contribute to the diagonal of the estimated moment matrix. The degrees of freedom are automatically reduced by n . A simple example for the usefulness of the NODIAG option is the fit of the first-order factor model, $\mathbf{S} = \mathbf{F}\mathbf{F}' + \mathbf{U}^2$. In this case, you do not have to estimate the diagonal matrix of unique variances \mathbf{U}^2 that are fully determined by $\text{diag}(\mathbf{S} - \mathbf{F}\mathbf{F}')$.

WPENALTY | WPEN=*r*

specifies the penalty weight $r \geq 0$ for the WLS and DWLS fit of the diagonal elements of a correlation matrix (constant 1s). The criterion for weighted least-squares estimation of a correlation structure is

$$\mathbf{F}_{WLS} = \sum_{i=2}^n \sum_{j=1}^{i-1} \sum_{k=2}^n \sum_{l=1}^{k-1} w^{ij,kl} (s_{ij} - c_{ij})(s_{kl} - c_{kl}) + r \sum_i^n (s_{ii} - c_{ii})^2$$

where r is the penalty weight specified by the WPENALTY= r option and the $w^{ij,kl}$ are the elements of the inverse of the reduced $(n(n-1)/2) \times (n(n-1)/2)$ weight matrix that contains only the nonzero rows and columns of the full weight matrix \mathbf{W} . The second term is a penalty term to fit the diagonal elements of the correlation matrix. The default value is 100. The reciprocal of this value replaces the asymptotic variance corresponding to the diagonal elements of a correlation matrix in the weight matrix \mathbf{W} , and it is effective only with the ASYCOV=CORR option. The often used value $r = 1$ seems to be too small in many cases to fit the diagonal elements of a correlation matrix properly. The default WPENALTY= value emphasizes the impor-

tance of the fit of the diagonal elements in the correlation matrix. You can decrease or increase the value of r if you want to decrease or increase the importance of the diagonal elements fit. This option is effective only with the WLS or DWLS estimation method and the analysis of a correlation matrix. See the section “Estimation Criteria” on page 531 for more details.

WRIDGE= r

defines a ridge factor r for the diagonal of the weight matrix \mathbf{W} used in GLS, WLS, or DWLS estimation. The weight matrix \mathbf{W} is transformed to

$$\mathbf{W} \longrightarrow \tilde{\mathbf{W}} = \mathbf{W} + r(\text{diag}(\mathbf{W}))$$

The WRIDGE= option is applied on the weight matrix

- before the WPENALTY= option is applied on it
- before the weight matrix is written to the OUTWGT= data set
- before the weight matrix is displayed

Optimization Techniques

Since there is no single nonlinear optimization algorithm available that is clearly superior (in terms of stability, speed, and memory) for all applications, different types of optimization techniques are provided in the CALIS procedure. Each technique can be modified in various ways. The default optimization technique for less than 40 parameters ($t < 40$) is TECHNIQUE=LEVMAR. For $40 \leq t < 400$, TECHNIQUE=QUANEW is the default method, and for $t \geq 400$, TECHNIQUE=CONGRA is the default method. For more details, see the section “Use of Optimization Techniques” on page 551. You can specify the following set of options in the PROC CALIS statement or in the NLOPTIONS statement.

TECHNIQUE | TECH=*name*

OMETHOD | OM=*name*

specifies the optimization technique. Valid values for *name* are as follows:

- | | |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CONGRA CG | chooses one of four different conjugate-gradient optimization algorithms, which can be more precisely defined with the UPDATE= option and modified with the LINESEARCH= option. The conjugate-gradient techniques need only $O(t)$ memory compared to the $O(t^2)$ memory for the other three techniques, where t is the number of parameters. On the other hand, the conjugate-gradient techniques are significantly slower than other optimization techniques and should be used only when memory is insufficient for more efficient techniques. When you choose this option, UPDATE=PB by default. This is the default optimization technique if there are more than 400 parameters to estimate. |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- DBLDOG | DD performs a version of double dogleg optimization, which uses the gradient to update an approximation of the Cholesky factor of the Hessian. This technique is, in many aspects, very similar to the dual quasi-Newton method, but it does not use line search. The implementation is based on Dennis and Mei (1979) and Gay (1983).
- LEVMar | LM | MARQUARDT performs a highly stable but, for large problems, memory- and time-consuming Levenberg-Marquardt optimization technique, a slightly improved variant of the Moré (1978) implementation. This is the default optimization technique if there are fewer than 40 parameters to estimate.
- NEWRAP | NR | NEWTON performs a usually stable but, for large problems, memory- and time-consuming Newton-Raphson optimization technique. The algorithm combines a line-search algorithm with ridging, and it can be modified with the LINESEARCH= option. In releases prior to Release 6.11, this option invokes the NRRIDG option.
- NRRIDG | NRR | NR performs a usually stable but, for large problems, memory- and time-consuming Newton-Raphson optimization technique. This algorithm does not perform a line search. Since TECH=NRRIDG uses an orthogonal decomposition of the approximate Hessian, each iteration of TECH=NRRIDG can be slower than that of TECH=NEWRAP, which works with Cholesky decomposition. However, usually TECH=NRRIDG needs less iterations than TECH=NEWRAP.
- QUANEW | QN chooses one of four different quasi-Newton optimization algorithms that can be more precisely defined with the UPDATE= option and modified with the LINESEARCH= option. If boundary constraints are used, these techniques sometimes converge slowly. When you choose this option, UPDATE=DBFGS by default. If nonlinear constraints are specified in the NLINCON statement, a modification of Powell's (1982a, 1982b) VMCWD algorithm is used, which is a sequential quadratic programming (SQP) method. This algorithm can be modified by specifying VERSION=1, which replaces the update of the Lagrange multiplier estimate vector μ to the original update of Powell (1978a, 1978b) that is used in the VF02AD algorithm. This can be helpful for applications with linearly dependent active constraints. The QUANEW technique is the default optimization technique if there are nonlinear constraints specified or if there are more than 40 and fewer than 400 parameters to estimate. The QUANEW algorithm uses only first-order derivatives of the objective function and, if available, of the nonlinear constraint functions.

TRUREG TR	performs a usually very stable but, for large problems, memory- and time-consuming trust region optimization technique. The algorithm is implemented similar to Gay (1983) and Moré and Sorensen (1983).
NONE NO	does not perform any optimization. This option is similar to METHOD=NONE, but TECH=NONE also computes and displays residuals and goodness-of-fit statistics. If you specify METHOD=ML, METHOD=LSML, METHOD=GLS, METHOD=LSGLS, METHOD=WLS, or METHOD=LSWLS, this option allows computing and displaying (if the display options are specified) of the standard error estimates and modification indices corresponding to the input parameter estimates.

UPDATE | UPD=*name*

specifies the update method for the quasi-Newton or conjugate-gradient optimization technique.

For TECHNIQUE=CONGRA, the following updates can be used:

PB	performs the automatic restart update method of Powell (1977) and Beale (1972). This is the default.
FR	performs the Fletcher-Reeves update (Fletcher 1980, p. 63).
PR	performs the Polak-Ribiere update (Fletcher 1980, p. 66).
CD	performs a conjugate-descent update of Fletcher (1987).

For TECHNIQUE=DBLDOG, the following updates (Fletcher 1987) can be used:

DBFGS	performs the dual Broyden, Fletcher, Goldfarb, and Shanno (BFGS) update of the Cholesky factor of the Hessian matrix. This is the default.
DDFP	performs the dual Davidon, Fletcher, and Powell (DFP) update of the Cholesky factor of the Hessian matrix.

For TECHNIQUE=QUANEW, the following updates (Fletcher 1987) can be used:

BFGS	performs original BFGS update of the inverse Hessian matrix. This is the default for earlier releases.
DFP	performs the original DFP update of the inverse Hessian matrix.
DBFGS	performs the dual BFGS update of the Cholesky factor of the Hessian matrix. This is the default.
DDFP	performs the dual DFP update of the Cholesky factor of the Hessian matrix.

LINESEARCH | LIS | SMETHOD | SM=*i*

specifies the line-search method for the CONGRA, QUANEW, and NEWRAP optimization techniques. Refer to Fletcher (1980) for an introduction to line-search techniques. The value of *i* can be 1, . . . , 8; the default is *i* = 2.

- LIS=1 specifies a line-search method that needs the same number of function and gradient calls for cubic interpolation and cubic extrapolation; this method is similar to one used by the Harwell subroutine library.
- LIS=2 specifies a line-search method that needs more function calls than gradient calls for quadratic and cubic interpolation and cubic extrapolation; this method is implemented as shown in Fletcher (1987) and can be modified to an exact line search by using the LSPRECISION= option.
- LIS=3 specifies a line-search method that needs the same number of function and gradient calls for cubic interpolation and cubic extrapolation; this method is implemented as shown in Fletcher (1987) and can be modified to an exact line search by using the LSPRECISION= option.
- LIS=4 specifies a line-search method that needs the same number of function and gradient calls for stepwise extrapolation and cubic interpolation.
- LIS=5 specifies a line-search method that is a modified version of LIS=4.
- LIS=6 specifies golden section line search (Polak 1971), which uses only function values for linear approximation.
- LIS=7 specifies bisection line search (Polak 1971), which uses only function values for linear approximation.
- LIS=8 specifies Armijo line-search technique (Polak 1971), which uses only function values for linear approximation.

FCONV | FTOL=*r*

specifies the relative function convergence criterion. The optimization process is terminated when the relative difference of the function values of two consecutive iterations is smaller than the specified value of *r*, that is

$$\frac{|f(x^{(k)}) - f(x^{(k-1)})|}{\max(|f(x^{(k-1)})|, FSIZE)} \leq r$$

where *FSIZE* can be defined by the FSIZE= option in the NLOPTIONS statement. The default value is $r = 10^{-FDIGITS}$, where *FDIGITS* either can be specified in the NLOPTIONS statement or is set by default to $-\log_{10}(\epsilon)$, where ϵ is the machine precision.

GCONV | GTOL=*r*

specifies the relative gradient convergence criterion (see the ABSGCONV= option on page 504 for the absolute gradient convergence criterion).

Termination of all techniques (except the CONGRA technique) requires the normalized predicted function reduction to be small,

$$\frac{[g(x^{(k)})]'[\mathbf{G}^{(k)}]^{-1}g(x^{(k)})}{\max(|f(x^{(k)})|, FSIZE)} \leq r$$

where *FSIZE* can be defined by the FSIZE= option in the NLOPTIONS statement. For the CONGRA technique (where a reliable Hessian estimate \mathbf{G} is not available),

$$\frac{\|g(x^{(k)})\|_2^2 \|s(x^{(k)})\|_2}{\|g(x^{(k)}) - g(x^{(k-1)})\|_2 \max(|f(x^{(k)})|, FSIZE)} \leq r$$

is used. The default value is $r = 10^{-8}$.

Note that for releases prior to Release 6.11, the GCONV= option specified the absolute gradient convergence criterion.

INSTEP=*r*

For highly nonlinear objective functions, such as the EXP function, the default initial radius of the trust-region algorithms TRUREG, DBLDOG, and LEVMAR or the default step length of the line-search algorithms can produce arithmetic overflows. If this occurs, specify decreasing values of $0 < r < 1$ such as INSTEP=1E-1, INSTEP=1E-2, INSTEP=1E-4, . . ., until the iteration starts successfully.

- For trust-region algorithms (TRUREG, DBLDOG, and LEVMAR), the INSTEP option specifies a positive factor for the initial radius of the trust region. The default initial trust-region radius is the length of the scaled gradient, and it corresponds to the default radius factor of $r = 1$.
- For line-search algorithms (NEWRAP, CONGRA, and QUANEW), INSTEP specifies an upper bound for the initial step length for the line search during the first five iterations. The default initial step length is $r = 1$.

For releases prior to Release 6.11, specify the SALPHA= and RADIUS= options. For more details, see the section “Computational Problems” on page 564.

LSPRECISION | LSP=*r***SPPRECISION | SP=*r***

specifies the degree of accuracy that should be obtained by the line-search algorithms LIS=2 and LIS=3. Usually an imprecise line search is inexpensive and successful. For more difficult optimization problems, a more precise and more expensive line search may be necessary (Fletcher 1980, p.22). The second (default for NEWRAP, QUANEW, and CONGRA) and third line-search methods approach exact line search for small LSPRECISION= values. If you have numerical problems, you should decrease the LSPRECISION= value to obtain a more precise line search. The default LSPRECISION= values are displayed in the following table.

TECH=	UPDATE=	LSP default
QUANEW	DBFGS, BFGS	$r = 0.4$
QUANEW	DDFP, DFP	$r = 0.06$
CONGRA	all	$r = 0.1$
NEWRAP	no update	$r = 0.9$

For more details, refer to Fletcher (1980, pp. 25–29).

MAXFUNC | MAXFU=*i*

specifies the maximum number *i* of function calls in the optimization process. The default values are displayed in the following table.

TECH=	MAXFUNC default
LEVMAR, NEWRAP, NRRIDG, TRUREG	$i=125$
DBLDOG, QUANEW	$i=500$
CONGRA	$i=1000$

The default is used if you specify MAXFUNC=0. The optimization can be terminated only after completing a full iteration. Therefore, the number of function calls that is actually performed can exceed the number that is specified by the MAXFUNC= option.

MAXITER | MAXIT=*i* <*n*>

specifies the maximum number *i* of iterations in the optimization process. The default values are displayed in the following table.

TECH=	MAXITER default
LEVMAR, NEWRAP, NRRIDG, TRUREG	$i=50$
DBLDOG, QUANEW	$i=200$
CONGRA	$i=400$

The default is used if you specify MAXITER=0 or if you omit the MAXITER option.

The optional second value *n* is valid only for TECH=QUANEW with nonlinear constraints. It specifies an upper bound *n* for the number of iterations of an algorithm and reduces the violation of nonlinear constraints at a starting point. The default is $n=20$. For example, specifying

```
maxiter= . 0
```

means that you do not want to exceed the default number of iterations during the main optimization process and that you want to suppress the feasible point algorithm for nonlinear constraints.

RADIUS=*r*

is an alias for the INSTEP= option for Levenberg-Marquardt minimization.

SALPHA=*r*

is an alias for the INSTEP= option for line-search algorithms.

SPPRECISION | SP=*r*

is an alias for the LSPRECISION= option.

Displayed Output Options

There are three kinds of options to control the displayed output:

- The PCORR, KURTOSIS, MODIFICATION, NOMOD, PCOVES, PDETERM, PESTIM, PINITIAL, PJACPAT, PLATCOV, PREDET, PWEIGHT, RESIDUAL, SIMPLE, STDERR, and TOTEFF options refer to specific parts of displayed output.
- The PALL, PRINT, PSHORT, PSUMMARY, and NOPRINT options refer to special subsets of the displayed output options mentioned in the first item. If the NOPRINT option is not specified, a default set of output is displayed. The PRINT and PALL options add other output options to the default output, and the PSHORT and PSUMMARY options reduce the default displayed output.
- The PRIMAT and PRIVEC options describe the form in which some of the output is displayed (the only nonredundant information displayed by PRIVEC is the gradient).

Output Options	PALL	PRINT	default	PSHORT	PSUMMARY
fit indices	*	*	*	*	*
linear dependencies	*	*	*	*	*
PREDET	*	(*)	(*)	(*)	
model matrices	*	*	*	*	
PESTIM	*	*	*	*	
iteration history	*	*	*	*	
PINITIAL	*	*	*		
SIMPLE	*	*	*		
STDERR	*	*	*		
RESIDUAL	*	*			
KURTOSIS	*	*			
PLATCOV	*	*			
TOTEFF	*	*			
PCORR	*				
MODIFICATION	*				
PWEIGHT	*				
PCOVES					
PDETERM					
PJACPAT					
PRIMAT					
PRIVEC					

KURTOSIS | KU

computes and displays univariate kurtosis and skewness, various coefficients of multivariate kurtosis, and the numbers of observations that contribute most to the normalized multivariate kurtosis. See the section “Measures of Multivariate Kurtosis” on page 544 for more information. Using the KURTOSIS option implies the SIMPLE display option. This information is computed only if the DATA= data set is a

raw data set, and it is displayed by default if the PRINT option is specified. The multivariate LS kappa and the multivariate mean kappa are displayed only if you specify METHOD=WLS and the weight matrix is computed from an input raw data set. All measures of skewness and kurtosis are corrected for the mean. If an intercept variable is included in the analysis, the measures of multivariate kurtosis do not include the intercept variable in the corrected covariance matrix, as indicated by a displayed message. Using the BIASKUR option displays the biased values of univariate skewness and kurtosis.

MODIFICATION | MOD

computes and displays Lagrange multiplier test indices for constant parameter constraints, equality parameter constraints, and active boundary constraints, as well as univariate and multivariate Wald test indices. The modification indices are not computed in the case of unweighted or diagonally weighted least-squares estimation.

The Lagrange multiplier test (Bentler 1986; Lee 1985; Buse 1982) provides an estimate of the χ^2 reduction that results from dropping the constraint. For constant parameter constraints and active boundary constraints, the approximate change of the parameter value is displayed also. You can use this value to obtain an initial value if the parameter is allowed to vary in a modified model. For more information, see the section “Modification Indices” on page 560.

NOMOD

does not compute modification indices. The NOMOD option is useful in connection with the PALL option because it saves computing time.

NOPRINT | NOP

suppresses the displayed output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 15, “Using the Output Delivery System.”

PALL | ALL

displays all optional output except the output generated by the PCOVES, PDETERM, PJACPAT, and PRIVEC options.

Caution: The PALL option includes the very expensive computation of the modification indices. If you do not really need modification indices, you can save computing time by specifying the NOMOD option in addition to the PALL option.

PCORR | CORR

displays the (corrected or uncorrected) covariance or correlation matrix that is analyzed and the predicted model covariance or correlation matrix.

PCOVES | PCE

displays the following:

- the information matrix (crossproduct Jacobian)
- the approximate covariance matrix of the parameter estimates (generalized inverse of the information matrix)
- the approximate correlation matrix of the parameter estimates

The covariance matrix of the parameter estimates is not computed for estimation methods ULS and DWLS. This displayed output is not included in the output generated by the PALL option.

PDETERM | PDE

displays three coefficients of determination: the determination of all equations (DETAE), the determination of the structural equations (DETSE), and the determination of the manifest variable equations (DETMV). These determination coefficients are intended to be global means of the squared multiple correlations for different subsets of model equations and variables. The coefficients are displayed only when you specify a RAM or LINEQS model, but they are displayed for all five estimation methods: ULS, GLS, ML, WLS, and DWLS.

You can use the STRUCTEQ statement to define which equations are structural equations. If you don't use the STRUCTEQ statement, PROC CALIS uses its own default definition to identify structural equations.

The term “structural equation” is not defined in a unique way. The LISREL program defines the structural equations by the user-defined BETA matrix. In PROC CALIS, the default definition of a structural equation is an equation that has a dependent left side variable that appears at least once on the right side of another equation, or an equation that has at least one right side variable that is the left side variable of another equation. Therefore, PROC CALIS sometimes identifies more equations as structural equations than the LISREL program does.

If the model contains structural equations, PROC CALIS also displays the “Stability Coefficient of Reciprocal Causation,” that is, the largest eigenvalue of the \mathbf{BB}' matrix, where \mathbf{B} is the causal coefficient matrix of the structural equations. These coefficients are computed as in the LISREL VI program of Jöreskog and Sörbom (1985). This displayed output is not included in the output generated by the PALL option.

PESTIM | PES

displays the parameter estimates. In some cases, this includes displaying the standard errors and t values.

PINITIAL | PIN

displays the input model matrices and the vector of initial values.

PJACPAT | PJP

displays the structure of variable and constant elements of the Jacobian matrix. This displayed output is not included in the output generated by the PALL option.

PLATCOV | PLC

displays the following:

- the estimates of the covariances among the latent variables
- the estimates of the covariances between latent and manifest variables
- the latent variable score regression coefficients

The estimated covariances between latent and manifest variables and the latent variable score regression coefficients are written to the OUTSTAT= data set. You can use the score coefficients with PROC SCORE to compute factor scores.

PREDET | PRE

displays the pattern of variable and constant elements of the predicted moment matrix that is predetermined by the analysis model. It is especially helpful in finding manifest variables that are not used or that are used as exogenous variables in a complex model specified in the COSAN statement. Those entries of the predicted moment matrix for which the model generates variable (rather than constant) elements are displayed as missing values. This output is displayed even without specifying the PREDET option if the model generates constant elements in the predicted model matrix different from those in the analysis moment matrix and if you specify at least the PSHORT amount of displayed output.

If the analyzed matrix is a correlation matrix (containing constant elements of 1s in the diagonal) and the model generates a predicted model matrix with q constant (rather than variable) elements in the diagonal, the degrees of freedom are automatically reduced by q . The output generated by the PREDET option displays those constant diagonal positions. If you specify the DFREDUCE= or NODIAG option, this automatic reduction of the degrees of freedom is suppressed. See the section “Counting the Degrees of Freedom” on page 563 for more information.

PRIMAT | PMAT

displays parameter estimates, approximate standard errors, and t values in matrix form if you specify the analysis model in the RAM or LINEQS statement. When a COSAN statement is used, this occurs by default.

PRINT | PRI

adds the options KURTOSIS, RESIDUAL, PLATCOV, and TOTEFF to the default output.

PRIVEC | PVEC

displays parameter estimates, approximate standard errors, the gradient, and t values in vector form. The values are displayed with more decimal places. This displayed output is not included in the output generated by the PALL option.

PSHORT | SHORT | PSH

excludes the output produced by the PINITIAL, SIMPLE, and STDERR options from the default output.

PSUMMARY | SUMMARY | PSUM

displays the fit assessment table and the ERROR, WARNING, and NOTE messages.

PWEIGHT | PW

displays the weight matrix **W** used in the estimation. The weight matrix is displayed after the WRIDGE= and the WPENALTY= options are applied to it.

RESIDUAL | RES < = NORM | VARSTAND | ASYSTAND >

displays the absolute and normalized residual covariance matrix, the rank order of the largest residuals, and a bar chart of the residuals. This information is displayed by default when you specify the PRINT option.

Three types of normalized or standardized residual matrices can be chosen with the RESIDUAL= specification.

RESIDUAL= NORM Normalized Residuals

RESIDUAL= VARSTAND Variance Standardized Residuals

RESIDUAL= ASYSTAND Asymptotically Standardized Residuals

For more details, see the section “Assessment of Fit” on page 536.

SIMPLE | S

displays means, standard deviations, skewness, and univariate kurtosis if available. This information is displayed when you specify the PRINT option. If you specify the UCOV, UCORR, or NOINT option, the standard deviations are not corrected for the mean. If the KURTOSIS option is specified, the SIMPLE option is set by default.

STDERR | SE

displays approximate standard errors if estimation methods other than unweighted least squares (ULS) or diagonally weighted least squares (DWLS) are used (and the NOSTDERR option is not specified). If you specify neither the STDERR nor the NOSTDERR option, the standard errors are computed for the OUTRAM= data set. This information is displayed by default when you specify the PRINT option.

NOSTDERR | NOSE

specifies that standard errors should not be computed. Standard errors are not computed for unweighted least-squares (ULS) or diagonally weighted least-squares (DWLS) estimation. In general, standard errors are computed even if the STDERR display option is not used (for file output).

TOTEFF | TE

computes and displays total effects and indirect effects.

Miscellaneous Options

ALPHAECV= α

specifies the significance level for a $1 - \alpha$ confidence interval, $0 \leq \alpha \leq 1$, for the Browne & Cudeck (1993) expected cross validation index (ECVI). The default value is $\alpha = 0.1$, which corresponds to a 90% confidence interval for the ECVI.

ALPHARMS= α

specifies the significance level for a $1 - \alpha$ confidence interval, $0 \leq \alpha \leq 1$, for the Steiger & Lind (1980) root mean squared error of approximation (RMSEA) coefficient (refer to Browne and Du Toit 1992). The default value is $\alpha = 0.1$, which corresponds to a 90% confidence interval for the RMSEA.

ASINGULAR | ASING= r

specifies an absolute singularity criterion r , $r > 0$, for the inversion of the information matrix, which is needed to compute the covariance matrix. The following singularity criterion is used:

$$|d_{j,j}| \leq \max(ASING, VSING * |H_{j,j}|, MSING * \max(|H_{1,1}|, \dots, |H_{n,n}|))$$

In the preceding criterion, $d_{j,j}$ is the diagonal pivot of the matrix, and *VSING* and *MSING* are the specified values of the *VSINGULAR=* and *MSINGULAR=* options. The default value for *ASING* is the square root of the smallest positive double precision value. Note that, in many cases, a normalized matrix $\mathbf{D}^{-1}\mathbf{H}\mathbf{D}^{-1}$ is decomposed, and the singularity criteria are modified correspondingly.

BIASKUR

computes univariate skewness and kurtosis by formulas uncorrected for bias. See the section “Measures of Multivariate Kurtosis” on page 544 for more information.

COVSING=*r*

specifies a nonnegative threshold r , which determines whether the eigenvalues of the information matrix are considered to be zero. If the inverse of the information matrix is found to be singular (depending on the *VSINGULAR=*, *MSINGULAR=*, *ASINGULAR=*, or *SINGULAR=* option), a generalized inverse is computed using the eigenvalue decomposition of the singular matrix. Those eigenvalues smaller than r are considered to be zero. If a generalized inverse is computed and you do not specify the *NOPRINT* option, the distribution of eigenvalues is displayed.

DEMPHAS | DE=*r*

changes the initial values of all parameters that are located on the diagonals of the central model matrices by the relationship

$$diag_{new} = r(|diag_{old}| + 1)$$

The initial values of the diagonal elements of the central matrices should always be nonnegative to generate positive definite predicted model matrices in the first iteration. By using values of $r > 1$, for example, $r = 2$, $r = 10$, ..., you can increase these initial values to produce predicted model matrices with high positive eigenvalues in the first iteration. The *DEMPHAS=* option is effective independent of the way the initial values are set; that is, it changes the initial values set in the model specification as well as those set by an *INRAM=* data set and those automatically generated for *RAM*, *LINEQS*, or *FACTOR* model statements. It also affects the initial values set by the *START=* option, which uses, by default, *DEMPHAS=100* if a covariance matrix is analyzed and *DEMPHAS=10* for a correlation matrix.

FDCODE

replaces the analytic derivatives of the program statements by numeric derivatives (finite difference approximations). In general, this option is needed only when you have program statements that are too difficult for the built-in function compiler to differentiate analytically. For example, if the program code for the nonlinear constraints contains many arrays and many *DO* loops with array processing, the built-in function compiler can require too much time and memory to compute derivatives of the constraints with respect to the parameters. In this case, the Jacobian matrix of constraints is computed numerically by using finite difference approximations. The *FDCODE* option does not modify the kind of derivatives specified with the *HES-SALG=* option.

HESSALG | HA = 1 | 2 | 3 | 4 | 5 | 6 | 11

specifies the algorithm used to compute the (approximate) Hessian matrix when **TECHNIQUE=LEVMar** and **NEWMar**, to compute approximate standard errors of the parameter estimates, and to compute Lagrange multipliers. There are different groups of algorithms available.

- analytic formulas: HA=1,2,3,4,11
- finite difference approximation: HA=5,6
- dense storage: HA=1,2,3,4,5,6
- sparse storage: HA=11

If the Jacobian is more than 25% dense, the dense analytic algorithm, HA= 1, is used by default. The HA= 1 algorithm is faster than the other dense algorithms, but it needs considerably more memory for large problems than HA= 2,3,4. If the Jacobian is more than 75% sparse, the sparse analytic algorithm, HA= 11, is used by default. The dense analytic algorithm HA= 4 corresponds to the original COSAN algorithm; you are advised not to specify HA= 4 due to its very slow performance. If there is not enough memory available for the dense analytic algorithm HA= 1 and you must specify HA= 2 or HA= 3, it may be more efficient to use one of the quasi-Newton or conjugate-gradient optimization techniques since Levenberg-Marquardt and Newton-Raphson optimization techniques need to compute the Hessian matrix in each iteration. For approximate standard errors and modification indices, the Hessian matrix has to be computed at least once, regardless of the optimization technique.

The algorithms HA= 5 and HA= 6 compute approximate derivatives by using forward difference formulas. The HA= 5 algorithm corresponds to the analytic HA= 1: it is faster than HA= 6, however it needs much more memory. The HA= 6 algorithm corresponds to the analytic HA= 2: it is slower than HA= 5, however it needs much less memory.

Test computations of large sparse problems show that the sparse algorithm HA= 11 can be up to ten times faster than HA= 1 (and needs much less memory).

MSINGULAR | MSING=*r*

specifies a relative singularity criterion r , $r > 0$, for the inversion of the information matrix, which is needed to compute the covariance matrix. The following singularity criterion is used:

$$|d_{j,j}| \leq \max(ASING, VSING * |H_{j,j}|, MSING * \max(|H_{1,1}|, \dots, |H_{n,n}|))$$

where $d_{j,j}$ is the diagonal pivot of the matrix, and *ASING* and *VSING* are the specified values of the **ASINGULAR=** and **VSINGULAR=** options. If you do not specify the **SINGULAR=** option, the default value for *MSING* is 1E–12; otherwise, the default value is 1E–4 * **SINGULAR**. Note that, in many cases, a normalized matrix $\mathbf{D}^{-1}\mathbf{H}\mathbf{D}^{-1}$ is decomposed, and the singularity criteria are modified correspondingly.

NOADJDF

turns off the automatic adjustment of degrees of freedom when there are active constraints in the analysis. When the adjustment is in effect, most fit statistics and the associated probability levels will be affected. This option should be used when the researcher believes that the active constraints observed in the current sample will have little chance to occur in repeated sampling.

RANDOM =*i*

specifies a positive integer as a seed value for the pseudo-random number generator to generate initial values for the parameter estimates for which no other initial value assignments in the model definitions are made. Except for the parameters in the diagonal locations of the central matrices in the model, the initial values are set to random numbers in the range $0 \leq r \leq 1$. The values for parameters in the diagonals of the central matrices are random numbers multiplied by 10 or 100. For more information, see the section “Initial Estimates” on page 547.

SINGULAR | SING =*r*

specifies the singularity criterion r , $0 < r < 1$, used, for example, for matrix inversion. The default value is the square root of the relative machine precision or, equivalently, the square root of the largest double precision value that, when added to 1, results in 1.

SLMW=*r*

specifies the probability limit used for computing the stepwise multivariate Wald test. The process stops when the univariate probability is smaller than r . The default value is $r = 0.05$.

START =*r*

In general, this option is needed only in connection with the COSAN model statement, and it specifies a constant r as an initial value for all the parameter estimates for which no other initial value assignments in the pattern definitions are made. Start values in the diagonal locations of the central matrices are set to $100|r|$ if a COV or UCOV matrix is analyzed and $10|r|$ if a CORR or UCORR matrix is analyzed. The default value is $r = .5$. Unspecified initial values in a FACTOR, RAM, or LINEQS model are usually computed by PROC CALIS. If none of the initialization methods are able to compute all starting values for a model specified by a FACTOR, RAM, or LINEQS statement, then the start values of parameters that could not be computed are set to r , $10|r|$, or $100|r|$. If the DEMPHAS= option is used, the initial values of the diagonal elements of the central model matrices are multiplied by the value specified in the DEMPHAS= option. For more information, see the section “Initial Estimates” on page 547.

VSINGULAR | VSING=*r*

specifies a relative singularity criterion r , $r > 0$, for the inversion of the information matrix, which is needed to compute the covariance matrix. The following singularity criterion is used:

$$|d_{j,j}| \leq \max(ASING, VSING * |H_{j,j}|, MSING * \max(|H_{1,1}|, \dots, |H_{n,n}|))$$

where $d_{j,j}$ is the diagonal pivot of the matrix, and $ASING$ and $MSING$ are the specified

values of the ASINGULAR= and MSINGULAR= options. If you do not specify the SINGULAR= option, the default value for VSING is 1E-8; otherwise, the default value is SINGULAR. Note that in many cases a normalized matrix $\mathbf{D}^{-1}\mathbf{H}\mathbf{D}^{-1}$ is decomposed, and the singularity criteria are modified correspondingly.

COSAN Model Statement

COSAN *matrix_term* < + *matrix_term* . . > ;

where *matrix_term* represents

matrix_definition < * *matrix_definition* ... >

and *matrix_definition* represents

matrix_name (*column_number* < ,*general_form* < ,*transformation* >>)

The COSAN statement constructs the symmetric matrix model for the covariance analysis mentioned earlier (see the section “The Generalized COSAN Model” on page 440):

$$\mathbf{C} = \mathbf{F}_1\mathbf{P}_1\mathbf{F}_1' + \cdots + \mathbf{F}_m\mathbf{P}_m\mathbf{F}_m'$$

$$\mathbf{F}_k = \mathbf{F}_{k_1} \cdots \mathbf{F}_{k_{n(k)}}, \quad \text{and} \quad \mathbf{P}_k = \mathbf{P}_k', \quad k = 1, \dots, m$$

$$\mathbf{F}_{k_j} = \begin{cases} \mathbf{G}_{k_j} & \\ \mathbf{G}_{k_j}^{-1} & \\ (\mathbf{I} - \mathbf{G}_{k_j})^{-1} & \end{cases} \quad j = 1, \dots, n(k), \quad \text{and} \quad \mathbf{P}_k = \begin{cases} \mathbf{Q}_k & \\ \mathbf{Q}_k^{-1} & \end{cases}$$

You can specify only one COSAN statement with each PROC CALIS statement. The COSAN statement contains *m* *matrix_terms* corresponding to the generalized COSAN formula. The *matrix_terms* are separated by plus signs (+) according to the addition of the terms within the model.

Each *matrix_term* of the COSAN statement contains the definitions of the first $n(k)+1$ matrices, \mathbf{F}_{k_j} and \mathbf{P}_k , separated by asterisks (*) according to the multiplication of the matrices within the term. The matrices \mathbf{F}_k' of the right-hand-side product are redundant and are not specified within the COSAN statement.

Each *matrix_definition* consists of the name of the matrix (*matrix_name*), followed in parentheses by the number of columns of the matrix (*column_number*) and, optionally, one or two matrix properties, separated by commas, describing the form of the matrix.

The number of rows of the first matrix in each term is defined by the input correlation or covariance matrix. You can reorder and reduce the variables in the input moment matrix using the VAR statement. The number of rows of the other matrices within the term is defined by the number of columns of the preceding matrix.

The first matrix property describes the general form of the matrix in the model. You can choose one of the following specifications of the first matrix property. The default first matrix property is GEN.

Code	Description
IDE	specifies an identity matrix; if the matrix is not square, this specification describes an identity submatrix followed by a rectangular zero submatrix.
ZID	specifies an identity matrix; if the matrix is not square, this specification describes a rectangular zero submatrix followed by an identity submatrix.
DIA	specifies a diagonal matrix; if the matrix is not square, this specification describes a diagonal submatrix followed by a rectangular zero submatrix.
ZDI	specifies a diagonal matrix; if the matrix is not square, this specification describes a rectangular zero submatrix followed by a diagonal submatrix.
LOW	specifies a lower triangular matrix; the matrix can be rectangular.
UPP	specifies an upper triangular matrix; the matrix can be rectangular.
SYM	specifies a symmetric matrix; the matrix cannot be rectangular.
GEN	specifies a general rectangular matrix (default).

The second matrix property describes the kind of inverse matrix transformation. If the second matrix property is omitted, no transformation is applied to the matrix.

Code	Description
INV	uses the inverse of the matrix.
IMI	uses the inverse of the difference between the identity and the matrix.

You cannot specify a nonsquare parameter matrix as an INV or IMI model matrix. Specifying a matrix of type DIA, ZDI, UPP, LOW, or GEN is not necessary if you do not use the *unspecified location* list in the corresponding MATRIX statements. After PROC CALIS processes the corresponding MATRIX statements, the matrix type DIA, ZDI, UPP, LOW, or GEN is recognized from the pattern of possibly nonzero elements. If you do not specify the first matrix property and you use the *unspecified location* list in a corresponding MATRIX statement, the matrix is recognized as a GEN matrix. You can also generate an IDE or ZID matrix by specifying a DIA, ZDI, or IMI matrix and by using MATRIX statements that define the pattern structure. However, PROC CALIS would be unable to take advantage of the fast algorithms that are available for IDE and ZID matrices in this case.

For example, to specify a second-order factor analysis model

$$\mathbf{S} = \mathbf{F}_1 \mathbf{F}_2 \mathbf{P}_2 \mathbf{F}_2' \mathbf{F}_1' + \mathbf{F}_1 \mathbf{U}_2^2 \mathbf{F}_1' + \mathbf{U}_1^2$$

with $m_1 = 3$ first-order factors and $m_2 = 2$ second-order factors and with $n = 9$ variables, you can use the following COSAN statement:

```
cosan F1(3) * F2(2) * P2(2,SYM)+F1(3) * U2(3,DIA) * I1(3,IDE)
      +U1(9,DIA) * I2(9,IDE)
```

MATRIX Statement

MATRIX *matrix-name* < *location* > = *list* < , *location* = *list* ... > ;

You can specify one or more MATRIX statements with a COSAN or FACTOR statement. A MATRIX statement specifies which elements of the matrix are constant and which are parameters. You can also assign values to the constant elements and initial values for the parameters. The input notation resembles that used in the COSAN program of R. McDonald and C. Fraser (personal communication), except that in PROC CALIS, parameters are distinguished from constants by giving parameters names instead of by using positive and negative integers.

A MATRIX statement cannot be used for an IDE or ZID matrix. For all other types of matrices, each element is assumed to be a constant of 0 unless a MATRIX statement specifies otherwise. Hence, there must be at least one MATRIX statement for each matrix mentioned in the COSAN statement except for IDE and ZID matrices. There can be more than one MATRIX statement for a given matrix. If the same matrix element is given different definitions, later definitions override earlier definitions.

At the start, all elements of each model matrix, except IDE or ZID matrices, are set equal to 0.

Description of *location*:

There are several ways to specify the starting *location* and continuation direction of a *list* with $n + 1$, $n \geq 0$, elements within the parameter matrix.

- [*i* , *j*] The *list* elements correspond to the diagonally continued matrix elements [*i*,*j*] , [*i*+1,*j*+1] , ... , [*i*+*n*,*j*+*n*]. The number of elements is defined by the length of the list and eventually terminated by the matrix boundaries. If the list contains just one element (constant or variable), then it is assigned to the matrix element [*i*,*j*].
- [*i* ,] The *list* elements correspond to the horizontally continued matrix elements [*i*,*j*] , [*i*,*j*+1] , ... , [*i*,*j*+*n*], where the starting column *j* is the diagonal position for a DIA, ZDI, or UPP matrix and is the first column for all other matrix types. For a SYM matrix, the list elements refer only to the matrix elements in the lower triangle. For a DIA or ZDI matrix, only one list element is accepted.
- [, *j*] The *list* elements correspond to the vertically continued matrix elements [*i*,*j*] , [*i*+1,*j*] , ... , [*i*+*n*,*j*], where the starting row *i* is equal to the diagonal position for a DIA, ZDI, SYM, or LOW matrix and is the first row for each other matrix type. For a SYM matrix, the list elements refer only to the matrix elements in the lower triangle. For a DIA or ZDI matrix, only one list element is accepted.
- [,] unspecified location: The *list* is allocated to all valid matrix positions (except for a ZDI matrix) starting at the element [1,1] and continuing rowwise. The only valid matrix positions for a DIA or ZDI matrix are the diagonal

elements; for an UPP or LOW matrix, the valid positions are the elements above or below the diagonal; and for a symmetric matrix, the valid positions are the elements in the lower triangle since the other triangle receives the symmetric allocation automatically. This *location* definition differs from the definitions with specified pattern locations in one important respect: if the number of elements in the *list* is smaller than the number of valid matrix elements, the list is repeated in the allocation process until all valid matrix elements are filled.

Omitting the left-hand-side term is equivalent to using [,] for an *unspecified location*.

Description of *list*:

The *list* contains numeric values or parameter names, or both, that are assigned to a list of matrix elements starting at a specified position and proceeding in a specified direction. A real number r in the list defines the corresponding matrix element as a constant element with this value. The notation $n * r$ generates n values of r in the list. A name in the list defines the corresponding matrix element as a parameter to be estimated. You can use numbered name lists (X1-X10) or the asterisk notation ($5 * X$ means five occurrences of the parameter X). If a sublist of n_1 names inside a *list* is followed by a list of $n_2 \leq n_1$ real values inside parentheses, the last n_2 parameters in the name sublist are given the initial values mentioned inside the parenthesis. For example, the following *list*

```
0. 1. A2-A5 (1.4 1.9 2.5) 5.
```

specifies that the first two matrix elements (specified by the *location* to the left of the equal sign) are constants with values 0 and 1. The next element is parameter A2 with no specified initial value. The next three matrix elements are the variable parameters A3, A4, and A5 with initial values 1.4, 1.9, and 2.5, respectively. The next matrix element is specified by the seventh list element to be the constant 5.

If your model contains many unconstrained parameters and it is too cumbersome to find different parameter names, you can specify all those parameters by the same prefix name. A prefix is a short name followed by a colon. The CALIS procedure generates a parameter name by appending an integer suffix to this prefix name. The prefix name should have no more than five or six characters so that the generated parameter name is not longer than eight characters. For example, if the prefix A (the parameter A1) is already used once in a *list*, the previous example would be identical to

```
0. 1. 4 * A: (1.4 1.9 2.5) 5.
```

To avoid unintentional equality constraints, the prefix names should not coincide with explicitly defined parameter names.

If you do not assign initial values to the parameters (listed in parentheses following a name sublist within the pattern list), PROC CALIS assigns initial values as follows:

- If the PROC CALIS statement contains a START= r option, each uninitialized parameter is given the initial value r . The uninitialized parameters in the diag-

onals of the central model matrices are given the initial value $10|r|$, $100|r|$, or $|r|$ multiplied by the value specified in the DEMPHAS= option.

- If the PROC CALIS statement contains a RANDOM=*i* option, each uninitialized parameter is given a random initial value $0 \leq r \leq 1$. The uninitialized parameters in the diagonals of the central model matrices are given the random values multiplied by 10, 100, or the value specified in the DEMPHAS= option.
- Otherwise, the initial value is set corresponding to START=0.5.

For example, to specify a confirmatory second-order factor analysis model

$$\mathbf{S} = \mathbf{F}_1 \mathbf{F}_2 \mathbf{P}_2 \mathbf{F}_2' \mathbf{F}_1' + \mathbf{F}_1 \mathbf{U}_2^2 \mathbf{F}_1' + \mathbf{U}_1^2$$

with $m_1 = 3$ first-order factors, $m_2 = 2$ second-order factors, and $n = 9$ variables and the following matrix pattern,

$$\mathbf{F}_1 = \begin{pmatrix} X_1 & 0 & 0 \\ X_2 & 0 & 0 \\ X_3 & 0 & 0 \\ 0 & X_4 & 0 \\ 0 & X_5 & 0 \\ 0 & X_6 & 0 \\ 0 & 0 & X_7 \\ 0 & 0 & X_8 \\ 0 & 0 & X_9 \end{pmatrix}, \quad \mathbf{U}_1 = \begin{pmatrix} U_1 & & & & & & & & \\ & U_2 & & & & & & & \\ & & U_3 & & & & & & \\ & & & U_4 & & & & & \\ & & & & U_5 & & & & \\ & & & & & U_6 & & & \\ & & & & & & U_7 & & \\ & & & & & & & U_8 & \\ & & & & & & & & U_9 \end{pmatrix}$$

$$\mathbf{F}_2 = \begin{pmatrix} Y_1 & 0 \\ Y_1 & Y_2 \\ 0 & Y_2 \end{pmatrix}, \quad \mathbf{P}_2 = \begin{pmatrix} P & 0 \\ 0 & P \end{pmatrix}, \quad \mathbf{U}_2 = \begin{pmatrix} V_1 & & \\ & V_2 & \\ & & V_3 \end{pmatrix}$$

you can specify the following COSAN and MATRIX statements:

```

cosan f1(3) * f2(2) * p2(2,dia) + f1(3) * u2(3,dia) * i1(3,ide)
      + u1(9,dia) * i2(9,ide);
matrix f1
      [ ,1]= x1-x3,
      [ ,2]= 3 * 0 x4-x6,
      [ ,3]= 6 * 0 x7-x9;
matrix u1
      [1,1]=u1-u9;

matrix f2
      [ ,1]= 2 * y1,
      [ ,2]= 0. 2 * y2;
matrix u2 = 3 * v;;
matrix p2 = 2 * p;
run;
```

The matrix pattern includes several equality constraints. Two loadings in the first and second factor of \mathbf{F}_2 (parameter names Y1 and Y2) and the two factor correlations in the diagonal of matrix \mathbf{P}_2 (parameter name P) are constrained to be equal. There are many other ways to specify the same model. See Figure 19.2 for the path diagram of this model.

The MATRIX statement can also be used with the FACTOR model statement. See “Using the FACTOR and MATRIX Statements” on page 494 for the usage.

RAM Model Statement

RAM *list-entry* < , *list-entry* ... > ;

where *list-entry* represents

matrix-number row-number column-number <value> <parameter-name>

The RAM statement defines the elements of the symmetric RAM matrix model

$$\mathbf{v} = \mathbf{A}\mathbf{v} + \mathbf{u}$$

in the form of a list type input (McArdle and McDonald 1984).

The covariance structure is given by

$$\mathbf{C} = \mathbf{J}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{P}((\mathbf{I} - \mathbf{A})^{-1})'\mathbf{J}'$$

with selection matrix \mathbf{J} and

$$\mathbf{C} = \mathcal{E}\{\mathbf{J}\mathbf{v}\mathbf{v}'\mathbf{J}'\}, \quad \mathbf{P} = \mathcal{E}\{\mathbf{u}\mathbf{u}'\}$$

You can specify only one RAM statement with each PROC CALIS statement. Using the RAM statement requires that the first n variable numbers in the path diagram and in the vector \mathbf{v} correspond to the numbers of the n manifest variables of the given covariance or correlation matrix. If you are not sure what the order of the manifest variables in the DATA= data set is, use a VAR statement to specify the order of these observed variables. Using the AUGMENT option includes the INTERCEPT variable as a manifest variable with number $n + 1$ in the RAM model. In this case, latent variables have to start with $n + 2$. The box of each manifest variable in the path diagram is assigned the number of the variable in the covariance or correlation matrix.

The selection matrix \mathbf{J} is always a rectangular identity (IDE) matrix, and it does not have to be specified in the RAM statement. A constant matrix element is defined in a RAM statement by a *list-entry* with four numbers. You define a parameter element by three or four numbers followed by a name for the parameter. Separate the list entries with a comma. Each *list-entry* in the RAM statement corresponds to a path in the diagram, as follows:

- The first number in each list entry (*matrix-number*) is the number of arrow heads of the path, which is the same as the number of the matrix in the RAM model (1 := \mathbf{A} , 2 := \mathbf{P}).

- The second number in each list entry (*row-number*) is the number of the node in the diagram to which the path points, which is the same as the row number of the matrix element.
- The third number in each list entry (*column-number*) is the number of the node in the diagram from which the path originates, which is the same as the column number of the matrix element.
- The fourth number (*value*) gives the (initial) value of the path coefficient. If you do not specify a fifth *list-entry*, this number specifies a constant coefficient; otherwise, this number specifies the initial value of this parameter. It is not necessary to specify the fourth item. If you specify neither the fourth nor the fifth item, the constant is set to 1 by default. If the fourth item (*value*) is not specified for a parameter, PROC CALIS tries to compute an initial value for this parameter.
- If the path coefficient is a parameter rather than a constant, then a fifth item in the list entry (*parameter-name*) is required to assign a name to the parameter. Using the same name for different paths constrains the corresponding coefficients to be equal.

If the initial value of a parameter is not specified in the list, the initial value is chosen in one of the following ways:

- If the PROC CALIS statement contains a `RANDOM=i` option, then the parameter obtains a randomly generated initial value r , such that $0 \leq r \leq 1$. The uninitialized parameters in the diagonals of the central model matrices are given the random values r multiplied by 10, 100, or the value specified in the `DEMPHAS=` option.
- If the `RANDOM=` option is not used, PROC CALIS tries to estimate the initial values.
- If the initial values cannot be estimated, the value of the `START=` option is used as an initial value.

If your model contains many unconstrained parameters and it is too cumbersome to find different parameter names, you can specify all those parameters by the same prefix name. A prefix is a short name followed by a colon. The CALIS procedure then generates a parameter name by appending an integer suffix to this prefix name. The prefix name should have no more than five or six characters so that the generated parameter name is not longer than eight characters. To avoid unintentional equality constraints, the prefix names should not coincide with explicitly defined parameter names.

For example, you can specify the confirmatory second-order factor analysis model (mentioned on page 483)

$$\mathbf{S} = \mathbf{F}_1 \mathbf{F}_2 \mathbf{P}_2 \mathbf{F}_2' \mathbf{F}_1' + \mathbf{F}_1 \mathbf{U}_2^2 \mathbf{F}_1' + \mathbf{U}_1^2$$

using the following RAM model statement.

```

ram
  1  1 10    x1,
  1  2 10    x2,
  1  3 10    x3,
  1  4 11    x4,
  1  5 11    x5,
  1  6 11    x6,
  1  7 12    x7,
  1  8 12    x8,
  1  9 12    x9,
  1 10 13    y1,
  1 11 13    y1,
  1 11 14    y2,
  1 12 14    y2,
  2  1  1    u:,
  2  2  2    u:,
  2  3  3    u:,
  2  4  4    u:,
  2  5  5    u:,
  2  6  6    u:,
  2  7  7    u:,
  2  8  8    u:,
  2  9  9    u:,
  2 10 10    v:,
  2 11 11    v:,
  2 12 12    v:,
  2 13 13    p ,
  2 14 14    p ;
run;

```

The confirmatory second-order factor analysis model corresponds to the path diagram displayed in Figure 19.2.

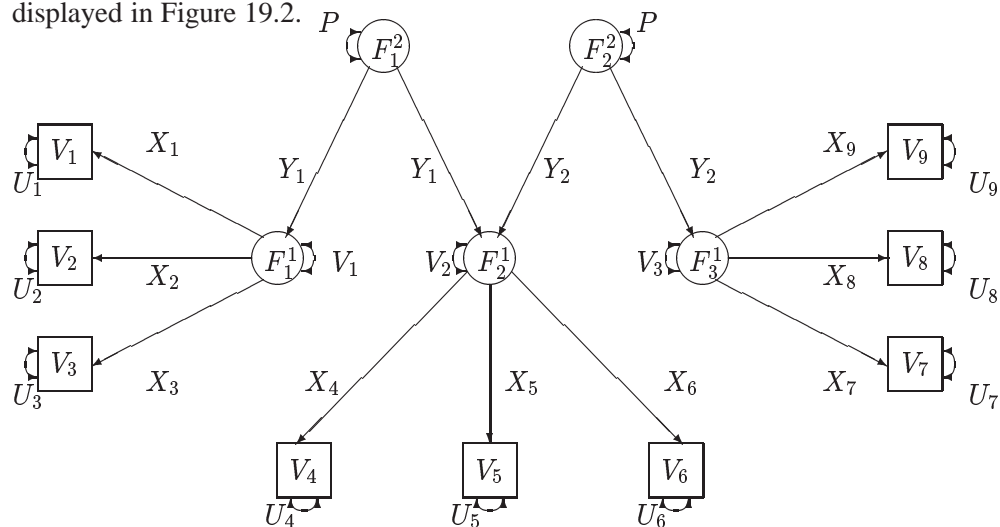
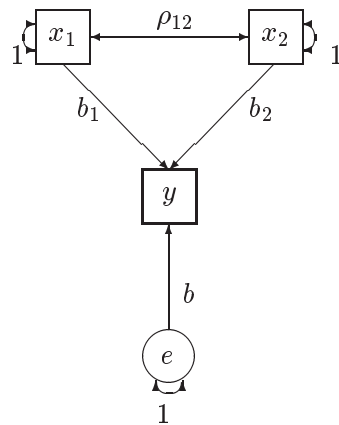
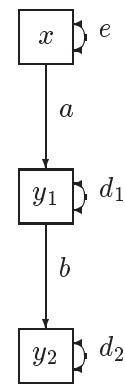


Figure 19.2. Path Diagram of Second-Order Factor Analysis Model

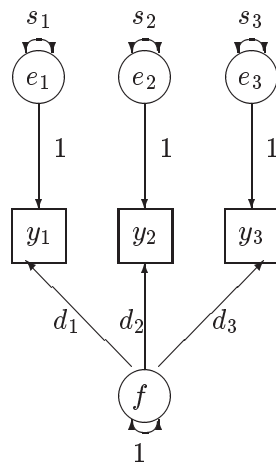
There is a very close relationship between the RAM model algebra and the specification of structural linear models by path diagrams. See Figure 19.3 for an example.



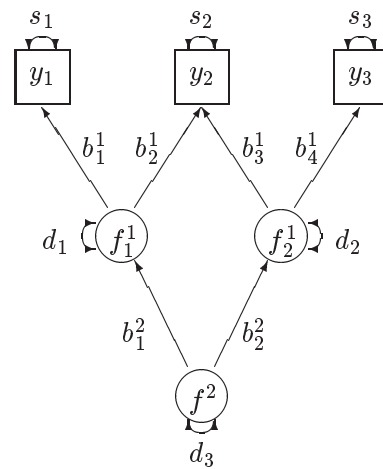
1. Multiple Regression



2. Chain Simplex



3. First-Order Factor Analysis



4. Second-Order Factor Analysis

Figure 19.3. Examples of RAM Nomography

Refer to McArdle (1980) for the interpretation of the models displayed in Figure 19.3.

LINEQS Model Statement

LINEQS *equation* < , *equation* ... > ;

where *equation* represents *dependent* = *term* < + *term* ... >

and where *term* represents one of the following:

- *coefficient-name* < (*number*) > *variable-name*
- *prefix-name* < (*number*) > *variable-name*
- < *number* > *variable-name*

The LINEQS statement defines the LINEQS model

$$\begin{aligned}\eta &= \beta\eta + \gamma\xi \\ \mathbf{C} &= \mathbf{J}(\mathbf{I} - \mathbf{B})^{-1}\mathbf{\Gamma}\mathbf{\Phi}\mathbf{\Gamma}'((\mathbf{I} - \mathbf{B})^{-1})'\mathbf{J}'\end{aligned}$$

You can specify only one LINEQS statement with each PROC CALIS statement. There are some differences from Bentler's notation in choosing the variable names. The length of each variable name is restricted to eight characters. The names of the manifest variables are defined in the DATA= input data set. The VAR statement can be used to select a subset of manifest variables in the DATA= input data set to analyze. You do not need to use a V prefix for manifest variables in the LINEQS statement nor do you need to use a numerical suffix in any variable name. The names of the latent variables must start with the prefix letter F (for Factor); the names of the residuals must start with the prefix letters E (for Error) or D (for Disturbance). The trailing part of the variable name can contain letters or digits. The prefix letter E is used for the errors of the manifest variables, and the prefix letter D is used for the disturbances of the latent variables. The names of the manifest variables in the DATA= input data set can start with F, E, or D, but these names should not coincide with the names of latent or error variables used in the model. The left-hand side (that is, endogenous *dependent* variable) of each equation should be either a manifest variable of the data set or a latent variable with prefix letter F. The left-hand-side variable should not appear on the right-hand side of the same equation; this means that matrix β should not have a nonzero diagonal element. Each equation should contain, at most, one E or D variable.

The equations must be separated by a comma. The order of the equations is arbitrary. The displayed output generally contains equations and terms in an order different from the input.

Coefficients to estimate are indicated in the equations by a name preceding the independent variable's name. The coefficient's name can be followed by a number inside parentheses indicating the initial value for this coefficient. A number preceding the independent variable's name indicates a constant coefficient. If neither a coefficient name nor a number precedes the independent variable's name, a constant coefficient of 1 is assumed.

If the initial value of a parameter is not specified in the equation, the initial value is chosen in one of the following ways:

- If you specify the **RANDOM=** option in the **PROC CALIS** statement, the variable obtains a randomly generated initial value r , such that $0 \leq r \leq 1$. The uninitialized parameters in the diagonals of the central model matrices are given the nonnegative random values r multiplied by 10, 100, or the value specified in the **DEMPHAS=** option.
- If the **RANDOM=** option is not used, **PROC CALIS** tries to estimate the initial values.
- If the initial values cannot be estimated, the value of the **START=** option is used as an initial value.

In Bentler's notation, estimated coefficients are indicated by asterisks. Referring to a parameter in Bentler's notation requires the specification of two variable names that correspond to the row and column of the position of the parameter in the matrix. Specifying the estimated coefficients by parameter names makes it easier to impose additional constraints with code. You do not need any additional statements to express equality constraints. Simply specify the same name for parameters that should have equal values.

If your model contains many unconstrained parameters and it is too cumbersome to find different parameter names, you can specify all those parameters by the same prefix name. A prefix is a short name followed by a colon. The **CALIS** procedure then generates a parameter name by appending an integer suffix to this prefix name. The prefix name should have no more than five or six characters so that the generated parameter name is not longer than eight characters. To avoid unintentional equality constraints, the prefix names should not coincide with explicitly defined parameter names.

For example, you can specify confirmatory second-order factor analysis model (mentioned on page 483)

$$\mathbf{S} = \mathbf{F}_1 \mathbf{F}_2 \mathbf{P}_2 \mathbf{F}_2' \mathbf{F}_1' + \mathbf{F}_1 \mathbf{U}_2^2 \mathbf{F}_1' + \mathbf{U}_1^2$$

by using the **LINEQS** and **STD** statements:

```
lineqs
      V1 = X1 F1 + E1,
      V2 = X2 F1 + E2,
      V3 = X3 F1 + E3,
      V4 = X4 F2 + E4,
      V5 = X5 F2 + E5,
      V6 = X6 F2 + E6,
      V7 = X7 F3 + E7,
      V8 = X8 F3 + E8,
      V9 = X9 F3 + E9,
      F1 = Y1 F4 + D1,
      F2 = Y1 F4 + Y2 F5 + D2,
      F3 = Y2 F5 + D3;
```

```

std
    E1-E9 = 9 * U:,
    D1-D3 = 3 * V:,
    F4 F5 = 2 * P;
run;

```

STD Statement

STD *assignment* < , *assignment* ... > ;

where *assignment* represents *variables* = *pattern-definition*

The STD statement tells which variances are parameters to estimate and which are fixed. The STD statement can be used only with the LINEQS statement. You can specify only one STD statement with each LINEQS model statement. The STD statement defines the diagonal elements of the central model matrix Φ . These elements correspond to the variances of the exogenous variables and to the error variances of the endogenous variables. Elements that are not defined are assumed to be 0.

Each *assignment* consists of a variable list (*variables*) on the left-hand side and a pattern list (*pattern-definition*) on the right-hand side of an equal sign. The *assignments* in the STD statement must be separated by commas. The *variables* list on the left-hand side of the equal sign should contain only names of variables that do not appear on the left-hand side of an equation in the LINEQS statement, that is, exogenous, error, and disturbance variables.

The *pattern-definition* on the right-hand side is similar to that used in the MATRIX statement. Each list element on the right-hand side defines the variance of the variable on the left-hand side in the same list position. A name on the right-hand side means that the corresponding variance is a parameter to estimate. A name on the right-hand side can be followed by a number inside parentheses that gives the initial value. A number on the right-hand side means that the corresponding variance of the variable on the left-hand side is fixed. If the right-hand-side list is longer than the left-hand-side variable list, the right-hand-side list is shortened to the length of the variable list. If the right-hand-side list is shorter than the variable list, the right-hand-side list is filled with repetitions of the last item in the list.

The right-hand side can also contain prefixes. A prefix is a short name followed by a colon. The CALIS procedure then generates a parameter name by appending an integer suffix to this prefix name. The prefix name should have no more than five or six characters so that the generated parameter name is not longer than eight characters. To avoid unintentional equality constraints, the prefix names should not coincide with explicitly defined parameter names. For example, if the prefix A is not used in any previous statement, this STD statement

```
std E1-E6=6 * A: (6 * 3.) ;
```

defines the six error variances as free parameters A_1, \dots, A_6 , all with starting values of 3.

COV Statement

COV *assignment* < , *assignment* ... > ;

where *assignment* represents *variables* < * *variables2* > = *pattern-definition*

The COV statement tells which covariances are parameters to estimate and which are fixed. The COV statement can be used only with the LINEQS statement. The COV statement differs from the STD statement only in the meaning of the left-hand-side *variables* list. You can specify only one COV statement with each LINEQS statement. The COV statement defines the off-diagonal elements of the central model matrix Φ . These elements correspond to the covariances of the exogenous variables and to the error covariances of the endogenous variables. Elements that are not defined are assumed to be 0. The *assignments* in the COV statement must be separated by commas.

The *variables* list on the left-hand side of the equal sign should contain only names of variables that do not appear on the left-hand side of an equation in the LINEQS statement, that is, exogenous, error, and disturbance variables.

The *pattern-definition* on the right-hand side is similar to that used in the MATRIX statement. Each list element on the right-hand side defines the covariance of a pair of variables in the list on the left-hand side. A name on the right-hand side can be followed by a number inside parentheses that gives the initial value. A number on the right-hand side means that the corresponding covariance of the variable on the left-hand side is fixed. If the right-hand-side list is longer than the left-hand-side variable list, the right-hand-side list is shortened to the length of the variable list. If the right-hand-side list is shorter than the variable list, the right-hand-side list is filled with repetitions of the last item in the list.

You can use one of two alternatives to refer to parts of Φ . The first alternative uses only one variable list and refers to all distinct pairs of variables within the list. The second alternative uses two variable lists separated by an asterisk and refers to all pairs of variables among the two lists.

Within-List Covariances

Using k variable names in the *variables* list on the left-hand side of an equal sign in a COV statement means that the parameter list (*pattern-definition*) on the right-hand side refers to all $k(k - 1)/2$ distinct variable pairs in the below-diagonal part of the Φ matrix. Order is very important. The order relation between the left-hand-side variable pairs and the right-hand-side parameter list is illustrated by the following example:

```
COV E1-E4 = PHI1-PHI6 ;
```

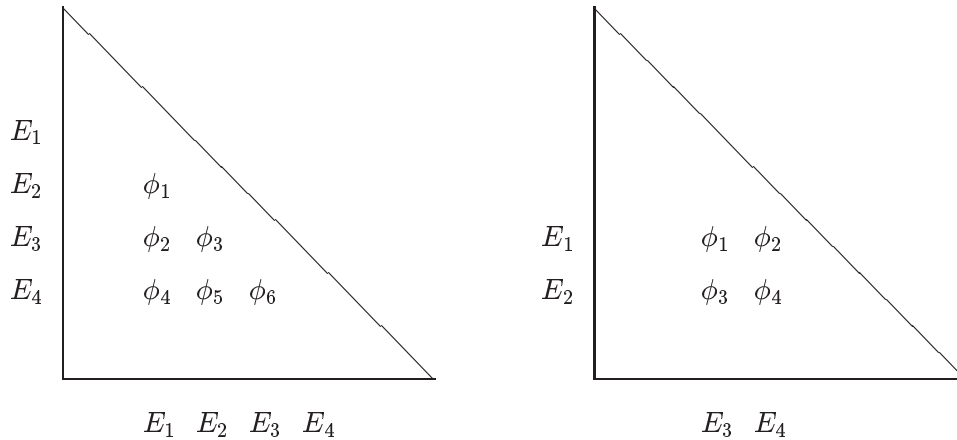
This is equivalent to the following specification:

```
COV E2 E1 = PHI1,
    E3 E1 = PHI2, E3 E2 = PHI3,
    E4 E1 = PHI4, E4 E2 = PHI5, E4 E3 = PHI6;
```

The symmetric elements are generated automatically. When you use prefix names on the right-hand sides, you do not have to count the exact number of parameters. For example,

```
COV E1-E4 = PHI: ;
```

generates the same list of parameter names if the prefix PHI is not used in a previous statement.



Within List Covariances

Between List Covariances

Figure 19.4. Within-List and Between-List Covariances

Between-List Covariances

Using k_1 and k_2 variable names in the two lists (separated by an asterisk) on the left-hand side of an equal sign in a COV statement means that the parameter list on the right-hand side refers to all $k_1 \times k_2$ distinct variable pairs in the Φ matrix. Order is very important. The order relation between the left-hand-side variable pairs and the right-hand-side parameter list is illustrated by the following example:

```
COV E1 E2 * E3 E4 = PHI1-PHI4 ;
```

This is equivalent to the following specification:

```
COV  E1 E3 = PHI1, E1 E4 = PHI2,
     E2 E3 = PHI3, E2 E4 = PHI4;
```

The symmetric elements are generated automatically.

Using prefix names on the right-hand sides lets you achieve the same purpose without counting the number of parameters. That is,

```
COV  E1 E2 * E3 E4 = PHI: ;
```

FACTOR Model Statement

FACTOR < options > ;

You can use the FACTOR statement to specify an exploratory or confirmatory first-order factor analysis of the given covariance or correlation matrix **C**,

$$\mathbf{C} = \mathbf{F}\mathbf{F}' + \mathbf{U}, \quad \mathbf{U} = \text{diag}$$

or

$$\mathbf{C} = \mathbf{F}\mathbf{P}\mathbf{F}' + \mathbf{U}, \quad \mathbf{P} = \mathbf{P}'$$

where **U** is a diagonal matrix and **P** is symmetric. Within this section, n denotes the number of manifest variables corresponding to the rows and columns of matrix **C**, and m denotes the number of latent variables (factors or components) corresponding to the columns of the loading matrix **F**.

You can specify only one FACTOR statement with each PROC CALIS statement. You can specify higher-order factor analysis problems using a COSAN model specification. PROC CALIS requires more computing time and memory than PROC FACTOR because it is designed for more general structural estimation problems and is unable to exploit the special properties of the unconstrained factor analysis model.

For default (exploratory) factor analysis, PROC CALIS computes initial estimates for factor loadings and unique variances by an algebraic method of approximate factor analysis. If you use a MATRIX statement together with a FACTOR model specification, initial values are computed by McDonald's (McDonald and Hartmann 1992) method (if possible). For details, see "Using the FACTOR and MATRIX Statements" on page 494. If neither of the two methods are appropriate, the initial values are set by the START= option.

The unrestricted factor analysis model is not identified because any orthogonal rotated factor loading matrix $\tilde{\mathbf{F}} = \mathbf{F}\boldsymbol{\Theta}$ is equivalent to the result **F**,

$$\mathbf{C} = \tilde{\mathbf{F}}\tilde{\mathbf{F}}' + \mathbf{U}, \quad \tilde{\mathbf{F}} = \mathbf{F}\boldsymbol{\Theta}, \quad \text{where} \quad \boldsymbol{\Theta}'\boldsymbol{\Theta} = \boldsymbol{\Theta}\boldsymbol{\Theta}' = \mathbf{I}$$

To obtain an identified factor solution, the FACTOR statement imposes zero constraints on the $m(m-1)/2$ elements in the upper triangle of **F** by default.

The following options are available in the FACTOR statement.

COMPONENT | COMP

computes a component analysis instead of a factor analysis (the diagonal matrix **U** in the model is set to 0). Note that the rank of $\mathbf{F}\mathbf{F}'$ is equal to the number m of components in **F**. If m is smaller than the number of variables in the moment matrix **C**, the matrix of predicted model values is singular and maximum likelihood estimates for **F** cannot be computed. You should compute ULS estimates in this case.

HEYWOOD | HEY

constrains the diagonal elements of **U** to be nonnegative; in other words, the model is replaced by

$$\mathbf{C} = \mathbf{F}\mathbf{F}' + \mathbf{U}^2, \quad \mathbf{U} = \text{diag}$$

N = m

specifies the number of first-order factors or components. The number m of factors should not exceed the number n of variables in the covariance or correlation matrix analyzed. For the saturated model, $m = n$, the COMP option should generally be specified for $\mathbf{U} = 0$; otherwise, $df < 0$. For $m = 0$ no factor loadings are estimated, and the model is $\mathbf{C} = \mathbf{U}$, with $\mathbf{U} = \text{diag}$. By default, $m = 1$.

NORM

normalizes the rows of the factor pattern for rotation using Kaiser's normalization.

ROTATE | R = name

specifies an orthogonal rotation. By default, ROTATE=NONE. The possible values for *name* are as follows:

PRINCIPAL | PC specifies a principal axis rotation. If ROTATE=PRINCIPAL is used with a factor rather than a component model, the following rotation is performed:

$$\mathbf{F}_{new} = \mathbf{F}_{old}\mathbf{T}, \quad \text{with} \quad \mathbf{F}_{old}'\mathbf{F}_{old} = \mathbf{T}\mathbf{\Lambda}\mathbf{T}'$$

where the columns of matrix **T** contain the eigenvectors of $\mathbf{F}_{old}'\mathbf{F}_{old}$.

QUARTIMAX | Q specifies quartimax rotation.

VARIMAX | V specifies varimax rotation.

EQUAMAX | E specifies equamax rotation.

PARSIMAX | P specifies parsimax rotation.

NONE performs no rotation (default).

Using the FACTOR and MATRIX Statements

You can specify the MATRIX statement and the FACTOR statement to compute a confirmatory first-order factor or component analysis. You can define the elements of the matrices **F**, **P**, and **U** of the oblique model,

$$\mathbf{C} = \mathbf{F}\mathbf{P}\mathbf{F}' + \mathbf{U}^2, \quad \mathbf{P} = \mathbf{P}', \quad \mathbf{U} = \text{diag}$$

To specify the structure for matrix **F**, **P**, or **U**, you have to refer to the matrix **_F_**, **_P_**, or **_U_** in the MATRIX statement. Matrix names automatically set by PROC CALIS always start with an underscore. As you name your own matrices or variables, you should avoid leading underscores.

The default matrix forms are as follows.

- _F_ lower triangular matrix (0 upper triangle for problem identification, removing rotational invariance)
- _P_ identity matrix (constant)
- _U_ diagonal matrix

For details about specifying the elements in matrices, see the section “MATRIX Statement” on page 481. If you are using at least one MATRIX statement in connection with a FACTOR model statement, you can also use the BOUNDS or PARAMETERS statement and program statements to constrain the parameters named in the MATRIX statement. Initial estimates are computed by McDonald’s (McDonald and Hartmann 1992) method. McDonald’s method of computing initial values works better if you scale the factors by setting the factor variances to 1 rather than by setting the loadings of the reference variables equal to 1.

BOUNDS Statement

BOUNDS *constraint* < , *constraint* ... > ;

where *constraint* represents

< *number operator* > *parameter-list* < *operator number* >

You can use the BOUNDS statement to define boundary constraints for any parameter that has its name specified in a MATRIX, LINEQS, STD, COV, or RAM statement or that is used in the model of an INRAM= data set. Valid operators are <=, <, >=, >, and = or, equivalently, LE, LT, GE, GT, and EQ. The following is an example of the BOUNDS statement:

```

bounds          0.    <= a1-a9 x    <= 1. ,
                  -1.    <= c2-c5      ,
                   b1-b10 y    >= 0. ;

```

You must separate boundary constraints with a comma, and you can specify more than one BOUNDS statement. The feasible region for a parameter is the intersection of all boundary constraints specified for that parameter; if a parameter has a maximum lower boundary constraint larger than its minimum upper bound, the parameter is set equal to the minimum of the upper bounds.

If you need to compute the values of the upper or lower bounds, create a TYPE=EST data set containing _TYPE_='UPPERBD' or _TYPE_='LOWERBD' observations and use it as an INEST= or INVAR= input data set in a later PROC CALIS run.

The BOUNDS statement can contain only parameter names and numerical constants. You cannot use the names of variables created in program statements.

The active set strategies made available in PROC CALIS cannot realize the strict inequality constraints < or >. For example, you cannot specify BOUNDS x > 0; to prevent infinite values for $y = \log(x)$. Use BOUNDS x > 1E-8; instead.

If the CALIS procedure encounters negative diagonal elements in the central model matrices during the minimization process, serious convergence problems can occur.

You can use the BOUNDS statement to constrain these parameters to nonnegative values. Using negative values in these locations can lead to a smaller χ^2 value but uninterpretable estimates.

LINCON Statement

LINCON *constraint* < , *constraint* ... > ;

where *constraint* represents

number operator linear-term or

linear-term operator number ,

and *linear-term* is

<+/-><coefficient* > *parameter* <<+/-><coefficient* > *parameter*... >

The LINCON statement specifies a set of linear equality or inequality constraints of the form

$$\sum_{j=1}^n a_{ij}x_j \leq b_i, \quad i = 1, \dots, m$$

The constraints must be separated by commas. Each linear constraint i in the statement consists of a linear combination $\sum_j a_{ij}x_j$ of a subset of the n parameters $x_j, j = 1, \dots, n$, and a constant value b_i separated by a comparison operator. Valid operators are <=, <, >=, >, and = or, equivalently, LE, LT, GE, GT, and EQ. PROC CALIS cannot enforce the strict inequalities < or >. Note that the coefficients a_{ij} in the linear combination must be constant numbers and must be followed by an asterisk and the name of a parameter (for example, listed in the PARMS, STD or COV statement). The following is an example of the LINCON statement that sets a linear constraint on parameters x1 and x2:

```
lincon      x1 + 3 * x2 <= 1;
```

Although you can easily express boundary constraints in LINCON statements, for many applications it is much more convenient to specify both the BOUNDS and the LINCON statements in the same PROC CALIS call.

The LINCON statement can contain only parameter names, operators, and numerical constants. If you need to compute the values of the coefficients a_{ij} or right-hand sides b_i , you can run a preliminary DATA step and create a TYPE=EST data set containing _TYPE_='LE', _TYPE_='GE', or _TYPE_='EQ' observations, then specify this data set as an INEST= or INVAR= data set in a following PROC CALIS run.

NLINCON Statement

NLINCON | NLC *constraint* < , *constraint* ... > ;

where *constraint* represents

number operator variable-list number operator or
variable-list operator number or
number operator variable-list

You can specify nonlinear equality and inequality constraints with the NLINCON or NLC statement. The QUANEW optimization subroutine is used when you specify nonlinear constraints using the NLINCON statement.

The syntax of the NLINCON statement is similar to that of the BOUNDS statement, except that the NLINCON statement must contain the names of variables that are defined in the program statements and are defined as continuous functions of parameters in the model. They must not be confused with the variables in the data set.

As with the BOUNDS statement, one- or two-sided constraints are allowed in the NLINCON statement; equality constraints must be one sided. Valid operators are <=, <, >=, >, and = or, equivalently, LE, LT, GE, GT, and EQ.

PROC CALIS cannot enforce the strict inequalities < or > but instead treats them as <= and >=, respectively. The listed nonlinear constraints must be separated by commas. The following is an example of the NLINCON statement that constrains the nonlinear parametric function $x_1 * x_1 + u_1$, which is defined below in a program statement, to a fixed value of 1:

```
nlincon    xx = 1;
xx = x1 * x1 + u1;
```

Note that x1 and u1 are parameters defined in the model. The following three NLINCON statements, which require xx1, xx2, and xx3 to be between zero and ten, are equivalent:

```
nlincon 0. <= xx1-xx3,
          xx1-xx3 <= 10;
nlincon 0. <= xx1-xx3 <= 10.;
nlincon 10. >= xx1-xx3 >= 0.;
```

NLOPTIONS Statement

NLOPTIONS *option(s)* ;

Many options that are available in PROC NLP can now be specified for the optimization subroutines in PROC CALIS using the NLOPTIONS statement. The NLOPTIONS statement provides more displayed and file output on the results of the optimization process, and it permits the same set of termination criteria as in PROC NLP. These are more technical options that you may not need to specify in most cases. The available options are summarized in Table 19.2 through Table 19.4, and the options are described in detail in the following three sections.

Table 19.2. Options Documented in the PROC CALIS Statement

Option	Short Description
Estimation Methods	
G4= <i>i</i>	algorithm for computing STDERR
Optimization Techniques	
TECHNIQUE= <i>name</i>	minimization method
UPDATE= <i>name</i>	update technique
LINESEARCH= <i>i</i>	line-search method
FCONV= <i>r</i>	relative change function convergence criterion
GCONV= <i>r</i>	relative gradient convergence criterion
INSTEP= <i>r</i>	initial step length (SALPHA=, RADIUS=)
LSPRECISION= <i>r</i>	line-search precision
MAXFUNC= <i>i</i>	maximum number of function calls
MAXITER= <i>i</i> < <i>n</i> >	maximum number of iterations
Miscellaneous Options	
ASINGULAR= <i>r</i>	absolute singularity criterion for inversion of the information matrix
COVSING= <i>r</i>	singularity tolerance of the information matrix
MSINGULAR= <i>r</i>	relative M singularity criterion for inversion of the information matrix
SINGULAR= <i>r</i>	singularity criterion for inversion of the Hessian
VSINGULAR= <i>r</i>	relative V singularity criterion for inversion of the information matrix

Table 19.3. Termination Criteria Options

Option	Short Description
Options Used by All Techniques	
ABSCONV= <i>r</i>	absolute function convergence criterion
MAXFUNC= <i>i</i>	maximum number of function calls
MAXITER= <i>i</i> < <i>n</i> >	maximum number of iterations
MAXTIME= <i>r</i>	maximum CPU time
MINITER= <i>i</i>	minimum number of iterations
Options for Unconstrained and Linearly Constrained Techniques	
ABSFCONV= <i>r</i> < <i>n</i> >	absolute change function convergence criterion
ABSGCONV= <i>r</i> < <i>n</i> >	absolute gradient convergence criterion
ABSXCONV= <i>r</i> < <i>n</i> >	absolute change parameter convergence criterion
FCONV= <i>r</i> < <i>n</i> >	relative change function convergence criterion
FCONV2= <i>r</i> < <i>n</i> >	function convergence criterion
FDIGITS= <i>r</i>	precision in computation of the objective function
FSIZE= <i>r</i>	parameter for FCONV= and GCONV=
GCONV= <i>r</i> < <i>n</i> >	relative gradient convergence criterion
GCONV2= <i>r</i> < <i>n</i> >	relative gradient convergence criterion
XCONV= <i>r</i> < <i>n</i> >	relative change parameter convergence criterion
XSIZE= <i>r</i>	parameter for XCONV=
Options for Nonlinearly Constrained Techniques	
ABSGCONV= <i>r</i> < <i>n</i> >	maximum absolute gradient of Lagrange function criterion
FCONV2= <i>r</i> < <i>n</i> >	predicted objective function reduction criterion
GCONV= <i>r</i> < <i>n</i> >	normalized predicted objective function reduction criterion

Table 19.4. Miscellaneous Options

Option	Short Description
Options for the Approximate Covariance Matrix of Parameter Estimates	
CFACTOR= <i>r</i>	scalar factor for STDERR
NOHLF	use Hessian of the objective function for STDERR
Options for Additional Displayed Output	
PALL	display initial and final optimization values
PCRPJAC	display approximate Hessian matrix
PHessian	display Hessian matrix
PHISTORY	display optimization history
PINIT	display initial values and derivatives (PALL)
PNLCJAC	display Jacobian matrix of nonlinear constraints (PALL)
PRINT	display results of the optimization process
Additional Options for Optimization Techniques	
DAMPSTEP<= <i>r</i> >	controls initial line-search step size
HESCAL= <i>n</i>	scaling version of Hessian or Jacobian
LCDEACT= <i>r</i>	Lagrange multiplier threshold of constraint
LCEPSILON= <i>r</i>	range for boundary and linear constraints
LCSINGULAR= <i>r</i>	QR decomposition linear dependence criterion
NOEIGNUM	suppress computation of matrices
RESTART= <i>i</i>	restart algorithm with a steepest descent direction
VERSION=1 2	quasi-Newton optimization technique version

Options Documented in the PROC CALIS Statement

The following options are the same as in the PROC CALIS statement and are documented in the section “PROC CALIS Statement” on page 456.

Estimation Method Option

G4=*i*

specifies the method for computing the generalized (G2 or G4) inverse of a singular matrix needed for the approximate covariance matrix of parameter estimates. This option is valid only for applications where the approximate covariance matrix of parameter estimates is found to be singular.

Optimization Technique Options**TECHNIQUE | TECH=***name***OMETHOD | OM=***name*

specifies the optimization technique.

UPDATE | UPD=*name*

specifies the update method for the quasi-Newton or conjugate-gradient optimization technique.

LINESEARCH | LIS=*i*

specifies the line-search method for the CONGRA, QUANEW, and NEWRAP optimization techniques.

FCNV | FTOL=*r*

specifies the relative function convergence criterion. For more details, see the section “Termination Criteria Options” on page 502.

GCONV | GTOL=*r*

specifies the relative gradient convergence criterion. For more details, see the section “Termination Criteria Options” on page 502.

INSTEP | SALPHA | RADIUS=*r*

restricts the step length of an optimization algorithm during the first iterations.

LSPRECISION | LSP=*r*

specifies the degree of accuracy that should be obtained by the line-search algorithms LIS=2 and LIS=3.

MAXFUNC | MAXFU=*i*specifies the maximum number *i* of function calls in the optimization process. For more details, see the section “Termination Criteria Options” on page 502.**MAXITER | MAXIT=***i* < *n* >specifies the maximum number *i* of iterations in the optimization process. For more details, see the section “Termination Criteria Options” on page 502.**Miscellaneous Options****ASINGULAR | ASING=***r*specifies an absolute singularity criterion *r*, $r > 0$, for the inversion of the information matrix, which is needed to compute the approximate covariance matrix of parameter estimates.**COVSING=***r*specifies a nonnegative threshold *r*, $r > 0$, that decides whether the eigenvalues of the information matrix are considered to be zero. This option is valid only for applications where the approximate covariance matrix of parameter estimates is found to be singular.**MSINGULAR | MSING=***r*specifies a relative singularity criterion *r*, $r > 0$, for the inversion of the information matrix, which is needed to compute the approximate covariance matrix of parameter estimates.

SINGULAR | SING =*r*

specifies the singularity criterion r , $0 \leq r \leq 1$, that is used for the inversion of the Hessian matrix. The default value is 1E-8.

VSINGULAR | VSING=*r*

specifies a relative singularity criterion r , $r > 0$, for the inversion of the information matrix, which is needed to compute the approximate covariance matrix of parameter estimates.

Termination Criteria Options

Let x^* be the point at which the objective function $f(\cdot)$ is optimized, and let $x^{(k)}$ be the parameter values attained at the k th iteration. All optimization techniques stop at the k th iteration if at least one of a set of termination criteria is satisfied. The specified termination criteria should allow termination in an area of sufficient size around x^* . You can avoid termination respective to any of the following function, gradient, or parameter criteria by setting the corresponding option to zero. There is a default set of termination criteria for each optimization technique; most of these default settings make the criteria ineffective for termination. PROC CALIS may have problems due to rounding errors (especially in derivative evaluations) that prevent an optimizer from satisfying strong termination criteria.

Note that PROC CALIS also terminates if the point $x^{(k)}$ is fully constrained by linearly independent active linear or boundary constraints, and all Lagrange multiplier estimates of active inequality constraints are greater than a small negative tolerance.

The following options are available only in the NLOPTIONS statement (except for FCONV, GCONV, MAXFUNC, and MAXITER), and they affect the termination criteria.

Options Used by All Techniques

The following five criteria are used by all optimization techniques.

ABSCONV | ABSTOL=*r*

specifies an absolute function convergence criterion.

- For minimization, termination requires

$$f^{(k)} = f(x^{(k)}) \leq ABSCONV$$

- For maximization, termination requires

$$f^{(k)} = f(x^{(k)}) \geq ABSCONV$$

The default value of ABSCONV is

- for minimization, the negative square root of the largest double precision value
- for maximization, the positive square root of the largest double precision value

MAXFUNC | MAXFU=*i*

requires the number of function calls to be no larger than i . The default values are listed in the following table.

TECH=	MAXFUNC default
LEVMar, NEWRAP, NRRIDG, TRUREG	$i=125$
DBLDOG, QUANEW	$i=500$
CONGRA	$i=1000$

The default is used if you specify MAXFUNC=0. The optimization can be terminated only after completing a full iteration. Therefore, the number of function calls that is actually performed can exceed the number that is specified by the MAXFUNC= option.

MAXITER | MAXIT= $i < n >$

requires the number of iterations to be no larger than i . The default values are listed in the following table.

TECH=	MAXITER default
LEVMar, NEWRAP, NRRIDG, TRUREG	$i=50$
DBLDOG, QUANEW	$i=200$
CONGRA	$i=400$

The default is used if you specify MAXITER=0 or you omit the MAXITER option.

The optional second value n is valid only for TECH=QUANEW with nonlinear constraints. It specifies an upper bound n for the number of iterations of an algorithm and reduces the violation of nonlinear constraints at a starting point. The default value is $n=20$. For example, specifying MAXITER= . 0 means that you do not want to exceed the default number of iterations during the main optimization process and that you want to suppress the feasible point algorithm for nonlinear constraints.

MAXTIME= r

requires the CPU time to be no larger than r . The default value of the MAXTIME= option is the largest double floating point number on your computer.

MINITER | MINIT= i

specifies the minimum number of iterations. The default value is $i = 0$.

The ABSCONV=, MAXITER=, MAXFUNC=, and MAXTIME= options are useful for dividing a time-consuming optimization problem into a series of smaller problems by using the OUTEST= and INEST= data sets.

Options for Unconstrained and Linearly Constrained Techniques

This section contains additional termination criteria for all unconstrained, boundary, or linearly constrained optimization techniques.

ABSFCNV | ABSFTOL= $r < n >$

specifies the absolute function convergence criterion. Termination requires a small change of the function value in successive iterations,

$$|f(x^{(k-1)}) - f(x^{(k)})| \leq r$$

The default value is $r = 0$. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

ABSGCONV | ABSGTOL= $r < n >$

specifies the absolute gradient convergence criterion. Termination requires the maximum absolute gradient element to be small,

$$\max_j |g_j^{(k)}| \leq r$$

The default value is $r=1\text{E}-5$. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

Note: In some applications, the small default value of the ABSGCONV= criterion is too difficult to satisfy for some of the optimization techniques.

ABSXCONV | ABSXTOL= $r < n >$

specifies the absolute parameter convergence criterion. Termination requires a small Euclidean distance between successive parameter vectors,

$$\|x^{(k)} - x^{(k-1)}\|_2 \leq r$$

The default value is $r = 0$. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

FCONV | FTOL= $r < n >$

specifies the relative function convergence criterion. Termination requires a small relative change of the function value in successive iterations,

$$\frac{|f(x^{(k)}) - f(x^{(k-1)})|}{\max(|f(x^{(k-1)})|, FSIZE)} \leq r$$

where $FSIZE$ is defined by the $FSIZE=$ option. The default value is $r = 10^{-FDIGITS}$, where $FDIGITS$ either is specified or is set by default to $-\log_{10}(\epsilon)$, where ϵ is the machine precision. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

FCONV2 | FTOL2= $r < n >$

specifies another function convergence criterion. For least-squares problems, termination requires a small predicted reduction

$$df^{(k)} \approx f(x^{(k)}) - f(x^{(k)} + s^{(k)})$$

of the objective function.

The predicted reduction

$$\begin{aligned} df^{(k)} &= -g^{(k)'} s^{(k)} - \frac{1}{2} s^{(k)'} G^{(k)} s^{(k)} \\ &= -\frac{1}{2} s^{(k)'} g^{(k)} \\ &\leq r \end{aligned}$$

is computed by approximating the objective function f by the first two terms of the Taylor series and substituting the Newton step

$$s^{(k)} = -G^{(k)-1} g^{(k)}$$

The FCONV2 criterion is the unscaled version of the GCONV criterion. The default value is $r = 0$. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

FDIGITS= r

specifies the number of accurate digits in evaluations of the objective function. Fractional values such as FDIGITS=4.7 are allowed. The default value is $r = -\log_{10} \epsilon$, where ϵ is the machine precision. The value of r is used for the specification of the default value of the FCONV= option.

FSIZE= r

specifies the *FSIZE* parameter of the relative function and relative gradient termination criteria. The default value is $r = 0$. See the FCONV= and GCONV= options.

GCONV | GTOL= $r < n >$

specifies the relative gradient convergence criterion. For all techniques except the CONGRA technique, termination requires that the normalized predicted function reduction is small,

$$\frac{[g^{(k)}]' [G^{(k)}]^{-1} g^{(k)}}{\max(|f(x^{(k)})|, FSIZE)} \leq r$$

where *FSIZE* is defined by the FSIZE= option. For the CONGRA technique (where a reliable Hessian estimate \mathbf{G} is not available),

$$\frac{\|g^{(k)}\|_2^2 \|s^{(k)}\|_2}{\|g^{(k)} - g^{(k-1)}\|_2 \max(|f(x^{(k)})|, FSIZE)} \leq r$$

is used. The default value is $r=1\text{E}-8$. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

Note: The default setting for the GCONV= option sometimes leads to early termination far from the location of the optimum. This is especially true for the special form of this criterion used in the CONGRA optimization.

GCONV2 | GTOL2= $r < n >$

specifies another relative gradient convergence criterion. For least-squares problems and the TRUREG, LEVMAR, NRRIDG, and NEWRAP techniques, the criterion of Browne (1982) is used,

$$\max_j \frac{|g_j^{(k)}|}{\sqrt{f(x^{(k)})G_{j,j}^{(k)}}} \leq r$$

This criterion is not used by the other techniques. The default value is $r = 0$. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

XCONV | XTOL= $r < n >$

specifies the relative parameter convergence criterion. Termination requires a small relative parameter change in subsequent iterations,

$$\frac{\max_j |x_j^{(k)} - x_j^{(k-1)}|}{\max(|x_j^{(k)}|, |x_j^{(k-1)}|, XSIZE)} \leq r$$

The default value is $r = 0$. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

XSIZE= r

specifies the *XSIZE* parameter of the relative function and relative gradient termination criteria. The default value is $r = 0$. See the XCONV= option.

Options for Nonlinearly Constrained Techniques

The non-NMSIMP algorithms available for nonlinearly constrained optimization (currently only TECH=QUANEW) do not monotonically reduce either the value of the objective function or some kind of merit function that combines objective and constraint functions. Furthermore, the algorithm uses the watchdog technique with backtracking (Chamberlain et al., 1982). Therefore, no termination criteria are implemented that are based on the values (x or f) of successive iterations. In addition to the criteria used by all optimization techniques, only three more termination criteria are currently available, and they are based on the Lagrange function

$$L(x, \lambda) = f(x) - \sum_{i=1}^m \lambda_i c_i(x)$$

and its gradient

$$\nabla_x L(x, \lambda) = g(x) - \sum_{i=1}^m \lambda_i \nabla_x c_i(x)$$

Here, m denotes the total number of constraints, $g = g(x)$ denotes the gradient of the objective function, and λ denotes the m vector of Lagrange multipliers. The Kuhn-

Tucker conditions require that the gradient of the Lagrange function is zero at the optimal point (x^*, λ^*) :

$$\nabla_x L(x^*, \lambda^*) = 0$$

The termination criteria available for nonlinearly constrained optimization follow.

ABSGCONV | ABSGTOL= $r < n$ >

specifies that termination requires the maximum absolute gradient element of the Lagrange function to be small,

$$\max_j |\{\nabla_x L(x^{(k)}, \lambda^{(k)})\}_j| \leq r$$

The default value is $r=1\text{E}-5$. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

FCONV2 | FTOL2= $r < n$ >

specifies that termination requires the predicted objective function reduction to be small:

$$|g(x^{(k)})s(x^{(k)})| + \sum_{i=1}^m |\lambda_i c_i| \leq r$$

The default value is $r=1\text{E}-6$. This is the criterion used by the programs VMCWD and VF02AD (Powell 1982b). The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

GCONV | GTOL= $r < n$ >

specifies that termination requires the normalized predicted objective function reduction to be small:

$$\frac{|g(x^{(k)})s(x^{(k)})| + \sum_{i=1}^m |\lambda_i c_i(x^{(k)})|}{\max(|f(x^{(k)})|, FSIZE)} \leq r$$

where $FSIZE$ is defined by the $FSIZE=$ option. The default value is $r=1\text{E}-8$. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

Miscellaneous Options

Options for the Approximate Covariance Matrix of Parameter Estimates

You can specify the following options to modify the approximate covariance matrix of parameter estimates.

CFACTOR= r

specifies the scalar factor for the covariance matrix of parameter estimates. The scalar $r \geq 0$ replaces the default value c/NM . For more details, see the section “Approximate Standard Errors” on page 535.

NOHLF

specifies that the Hessian matrix of the objective function (rather than the Hessian matrix of the Lagrange function) is used for computing the approximate covariance matrix of parameter estimates and, therefore, the approximate standard errors.

It is theoretically not correct to use the NOHLF option. However, since most implementations use the Hessian matrix of the objective function and not the Hessian matrix of the Lagrange function for computing approximate standard errors, the NOHLF option can be used to compare the results.

Options for Additional Displayed Output

You can specify the following options to obtain additional displayed output.

PALL | ALL

displays information on the starting values and final values of the optimization process.

PCRPJAC | PJTJ

displays the approximate Hessian matrix. If general linear or nonlinear constraints are active at the solution, the projected approximate Hessian matrix is also displayed.

PHESIAN | PHES

displays the Hessian matrix. If general linear or nonlinear constraints are active at the solution, the projected Hessian matrix is also displayed.

PHISTORY | PHIS

displays the optimization history. The PHISTORY option is set automatically if the PALL or PRINT option is set.

PINIT | PIN

displays the initial values and derivatives (if available). The PINIT option is set automatically if the PALL option is set.

PNLCJAC

displays the Jacobian matrix of nonlinear constraints specified by the NLINCON statement. The PNLCJAC option is set automatically if the PALL option is set.

PRINT | PRI

displays the results of the optimization process, such as parameter estimates and constraints.

More Options for Optimization Techniques

You can specify the following options, in addition to the options already listed, to fine-tune the optimization process. These options should not be necessary in most applications of PROC CALIS.

DAMPSTEP | DS $\leq r$

specifies that the initial step-size value $\alpha^{(0)}$ for each line search (used by the QUANEW, CONGRA, or NEWRAP techniques) cannot be larger than r times the step-size value used in the former iteration. If the factor r is not specified, the default value is $r = 2$. The DAMPSTEP option can prevent the line-search algorithm from repeatedly stepping into regions where some objective functions are difficult to compute or where they can lead to floating point overflows during the computation of

objective functions and their derivatives. The DAMPSTEP<=*r*> option can prevent time-costly function calls during line searches with very small step sizes α of objective functions. For more information on setting the start values of each line search, see the section “Restricting the Step Length” on page 558.

HESCAL | HS = 0 | 1 | 2 | 3

specifies the scaling version of the Hessian or crossproduct Jacobian matrix used in NRRIDG, TRUREG, LEVMAR, NEWRAP, or DBLDOG optimization. If HS is not equal to zero, the first iteration and each restart iteration sets the diagonal scaling matrix $\mathbf{D}^{(0)} = \text{diag}(d_i^{(0)})$:

$$d_i^{(0)} = \sqrt{\max(|\mathbf{G}_{i,i}^{(0)}|, \epsilon)}$$

where $\mathbf{G}_{i,i}^{(0)}$ are the diagonal elements of the Hessian or crossproduct Jacobian matrix. In every other iteration, the diagonal scaling matrix $\mathbf{D}^{(0)} = \text{diag}(d_i^{(0)})$ is updated depending on the HS option:

HS=0 specifies that no scaling is done.

HS=1 specifies the Moré (1978) scaling update:

$$d_i^{(k+1)} = \max(d_i^{(k)}, \sqrt{\max(|\mathbf{G}_{i,i}^{(k)}|, \epsilon)})$$

HS=2 specifies the Dennis, Gay, and Welsch (1981) scaling update:

$$d_i^{(k+1)} = \max(0.6 * d_i^{(k)}, \sqrt{\max(|\mathbf{G}_{i,i}^{(k)}|, \epsilon)})$$

HS=3 specifies that d_i is reset in each iteration:

$$d_i^{(k+1)} = \sqrt{\max(|\mathbf{G}_{i,i}^{(k)}|, \epsilon)}$$

In the preceding equations, ϵ is the relative machine precision. The default is HS=1 for LEVMAR minimization and HS=0 otherwise. Scaling of the Hessian or crossproduct Jacobian can be time-consuming in the case where general linear constraints are active.

LCDEACT | LCD = *r*

specifies a threshold r for the Lagrange multiplier that decides whether an active inequality constraint remains active or can be deactivated. For maximization, r must be greater than zero; for minimization, r must be smaller than zero. The default is

$$r = \pm \min(0.01, \max(0.1 * ABSGCONV, 0.001 * gmax^{(k)}))$$

where “+” stands for maximization, “−” stands for minimization, $ABSGCONV$ is the value of the absolute gradient criterion, and $gmax^{(k)}$ is the maximum absolute element of the (projected) gradient $g^{(k)}$ or $Z'g^{(k)}$.

LCEPSILON | LCEPS | LCE = r

specifies the range r , $r \geq 0$, for active and violated boundary and linear constraints. If the point $x^{(k)}$ satisfies the condition

$$\left| \sum_{j=1}^n a_{ij} x_j^{(k)} - b_i \right| \leq r * (|b_i| + 1)$$

the constraint i is recognized as an active constraint. Otherwise, the constraint i is either an inactive inequality or a violated inequality or equality constraint. The default value is $r=1\text{E}-8$. During the optimization process, the introduction of rounding errors can force PROC NLP to increase the value of r by factors of 10. If this happens, it is indicated by a message displayed in the log.

LCSINGULAR | LCSING | LCS = r

specifies a criterion r , $r \geq 0$, used in the update of the QR decomposition that decides whether an active constraint is linearly dependent on a set of other active constraints. The default is $r=1\text{E}-8$. The larger r becomes, the more the active constraints are recognized as being linearly dependent.

NOEIGNUM

suppresses the computation and displayed output of the determinant and the inertia of the Hessian, crossproduct Jacobian, and covariance matrices. The inertia of a symmetric matrix are the numbers of negative, positive, and zero eigenvalues. For large applications, the NOEIGNUM option can save computer time.

RESTART | REST = i

specifies that the QUANEW or CONGRA algorithm is restarted with a steepest descent/ascent search direction after at most i iterations, $i > 0$. Default values are as follows:

- CONGRA: UPDATE=PB: restart is done automatically so specification of i is not used.
- CONGRA: UPDATE≠PB: $i = \min(10n, 80)$, where n is the number of parameters.
- QUANEW: i is the largest integer available.

VERSION | VS = 1 | 2

specifies the version of the quasi-Newton optimization technique with nonlinear constraints.

- | | |
|------|--------------------------------------------------------------------------------------------|
| VS=1 | specifies the update of the μ vector as in Powell (1978a, 1978b) (update like VF02AD). |
| VS=2 | specifies the update of the μ vector as in Powell (1982a, 1982b) (update like VMCWD). |

The default is VS=2.

PARAMETERS Statement

PARAMETERS | PARMS *parameter(s)* << = > *number(s)* >
 << , > *parameter(s)* << = > *number(s)* > ... > ;

The PARAMETERS statement defines additional parameters that are not elements of a model matrix to use in your own program statements. You can specify more than one PARAMETERS statement with each PROC CALIS statement. The *parameters* can be followed by an equal sign and a number list. The values of the *numbers* list are assigned as initial values to the preceding parameters in the *parameters* list. For example, each of the following statements assigns the initial values ALPHA=.5 and BETA=-.5 for the parameters used in program statements:

```
parameters alfa beta=.5 -.5;
parameters alfa beta (.5 -.5);
parameters alfa beta .5 -.5;
parameters alfa=.5 beta (-.5);
```

The number of parameters and the number of values does not have to match. When there are fewer values than parameter names, either the RANDOM= or START= option is used. When there are more values than parameter names, the extra values are dropped. Parameters listed in the PARAMETERS statement can be assigned initial values by program statements or by the START= or RANDOM= option in the PROC CALIS statement.

Caution: The OUTRAM= and INRAM= data sets do not contain any information about the PARAMETERS statement or additional program statements.

STRUCTEQ Statement

STRUCTEQ *variable* < *variable* ... > ;

The STRUCTEQ statement is used to list the dependent variables of the structural equations. This statement is ignored if you omit the PDETERM option. This statement is useful because the term *structural equation* is not defined in a unique way, and PROC CALIS has difficulty identifying the structural equations.

If LINEQS statements are used, the names of the left-hand-side (dependent) variables of those equations to be treated as structural equations should be listed in the STRUCTEQ statement.

If the RAM statement is used, variable names in the STRUCTEQ statements depend on the VARNAMES statement:

- If the VARNAMES statement is used, variable names must correspond to those in the VARNAMES statement.

- If the VARNAMEs statement is not used, variable names must correspond to the names of manifest variables or latent (F) variables.

The STRUcTEQ statement also defines the names of variables used in the causal coefficient matrix of the structural equations, **B**, for computing the *Stability Coefficient of Reciprocal Causation* (the largest eigenvalue of the **BB'** matrix). If the PROC CALIS option PDETERM is used without the STRUcTEQ statement, the structural equations are defined as described in the PDETERM option. See the PROC CALIS option PDETERM on page 473 for more details.

VARNAMEs Statement

VARNAMEs | **VNAMEs** *assignment* < , *assignment* ... > ;

where *assignment* represents

matrix-id variable-names or *matrix-name* = *matrix-name*

Use the VARNAMEs statement in connection with the RAM, COSAN, or FACTOR model statement to allocate names to latent variables including error and disturbance terms. This statement is not needed if you are using the LINEQS statement.

In connection with the RAM model statement, the *matrix-id* must be specified by the integer number as it is used in the RAM list input (1 for matrix **A**, 2 for matrix **P**). Because the first variables of matrix **A** correspond to the manifest variables in the input data set, you can specify names only for the latent variables following the manifest variables in the rows of **A**. For example, in the RAM notation of the alienation example, you can specify the latent variables by names F1, F2, F3 and the error variables by names E1, . . . , E6, D1, D2, D3 with the following statement:

```
vnames 1  F1-F3,
        2  E1-E6 D1-D3;
```

If the RAM model statement is not accompanied by a VNAMEs statement, default variable names are assigned using the prefixes F, E, and D with numerical suffixes: latent variables are F1, F2, . . . , and error variables are E1, E2,

The *matrix-id* must be specified by its name when used with the COSAN or FACTOR statement. The *variable-names* following the matrix name correspond to the columns of this matrix. The variable names corresponding to the rows of this matrix are set automatically by

- the names of the manifest variables for the first matrix in each term
- the column variable names of the same matrix for the central symmetric matrix in each term
- the column variable names of the preceding matrix for each other matrix

You also can use the second kind of name assignment in connection with a COSAN statement. Two matrix names separated by an equal sign allocate the column names of one matrix to the column names of the other matrix. This assignment assumes that the column names of at least one of the two matrices are already allocated. For example, in the COSAN notation of the alienation example, you can specify the variable names by using the following statements to allocate names to the columns of **J**, **A**, and **P**:

```

vnames  J  V1-V6 F1-F3 ,
        A  =J  ,
        P  E1-E6 D1-D3 ;

```

BY Statement

BY *variables* ;

You can specify a BY statement with PROC CALIS to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the CALIS procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

VAR Statement

VAR *variables* ;

The VAR statement lists the numeric variables to be analyzed. If the VAR statement is omitted, all numeric variables not mentioned in other statements are analyzed. You can use the VAR statement to ensure that the manifest variables appear in correct order for use in the RAM statement. Only one VAR statement can be used with each PROC CALIS statement. If you do not use all manifest variables when you specify

the model with a RAM or LINEQS statement, PROC CALIS does automatic variable selection. For more information, see the section “Automatic Variable Selection” on page 548.

PARTIAL Statement

PARTIAL *variables* ;

If you want the analysis to be based on a partial correlation or covariance matrix, use the PARTIAL statement to list the variables used to partial out the variables in the analysis. You can specify only one PARTIAL statement with each PROC CALIS statement.

FREQ Statement

FREQ *variable* ;

If one variable in your data set represents the frequency of occurrence for the other values in the observation, specify the variable's name in a FREQ statement. PROC CALIS then treats the data set as if each observation appears n_i times, where n_i is the value of the FREQ variable for observation i . Only the integer portion of the value is used. If the value of the FREQ variable is less than 1 or is missing, that observation is not included in the analysis. The total number of observations is considered to be the sum of the FREQ values when the procedure computes significance probabilities. You can use only one FREQ statement with each PROC CALIS statement.

WEIGHT Statement

WEIGHT *variable* ;

To compute weighted covariances or correlations, specify the name of the weighting variable in a WEIGHT statement. This is often done when the error variance associated with each observation is different and the values of the weight variable are proportional to the reciprocals of the variances. You can use only one WEIGHT statement with each PROC CALIS statement. The WEIGHT and FREQ statements have a similar effect, except the WEIGHT statement does not alter the number of observations unless VARDEF=WGT or VARDEF=WDF. An observation is used in the analysis only if the WEIGHT variable is greater than 0 and is not missing.

SAS Program Statements

This section lists the program statements used to express the linear and nonlinear constraints on the parameters and documents the differences between program statements in PROC CALIS and program statements in the DATA step. The very different

use of the ARRAY statement by PROC CALIS is also discussed. Most of the program statements that can be used in the SAS DATA step also can be used in PROC CALIS. Refer to *SAS Language Reference: Dictionary* for a description of the SAS program statements. You can specify the following SAS program statements to compute parameter constraints with the CALIS procedure:

```

ABORT ;
CALL name < ( expression < , expression ... > ) > ;
DELETE;
DO < variable = expression < TO expression> < BY expression>
    < , expression < TO expression> < BY expression> ... > >
    < WHILE expression>
    < UNTIL expression> ;
END;
GOTO statement-label ;
IF expression;
IF expression THEN program-statement ;
    ELSE program-statement ;
variable = expression ;
variable+expression ;
LINK statement-label ;
PUT <variable> <=> < ... > ;
RETURN ;
SELECT < ( expression ) > ;
STOP;
SUBSTR ( variable, index, length ) = expression ;
WHEN (expression) program-statement ;
    OTHERWISE program-statement ;

```

For the most part, the SAS program statements work the same as they do in the SAS DATA step as documented in *SAS Language Reference: Concepts*. However, there are several differences that should be noted.

- The ABORT statement does not allow any arguments.
- The DO statement does not allow a character index variable. Thus,

```
do I=1,2,3;
```

is supported; however,

```
do I='A','B','C';
```

is not valid in PROC CALIS, although it is supported in the DATA step.

- The PUT statement, used mostly for program debugging in PROC CALIS, supports only some of the features of the DATA step PUT statement, and it has some new features that the DATA step PUT statement does not have:
 - The CALIS procedure PUT statement does not support line pointers, factored lists, iteration factors, overprinting, _INFILE_, the colon (:) format modifier, or \$.

- The CALIS procedure PUT statement does support expressions enclosed in parentheses. For example, the following statement displays the square root of x:

```
put (sqrt(x));
```

- The CALIS procedure PUT statement supports the print item `_PDV_` to display a formatted listing of all variables in the program. For example, the following statement displays a much more readable listing of the variables than the `_ALL_` print item:

```
put _pdv_ ;
```

- The WHEN and OTHERWISE statements allow more than one target statement. That is, DO/END groups are not necessary for multiple WHEN statements. For example, the following syntax is valid:

```
select;
  when ( expression1 ) statement1;
                           statement2;
  when ( expression2 ) statement3;
                           statement4;
end;
```

You can specify one or more PARMS statements to define parameters used in the program statements that are not defined in the model matrices (MATRIX, RAM, LINEQS, STD, or COV statement).

Parameters that are used only on the right-hand side of your program statements are called independent, and parameters that are used at least once on the left-hand side of an equation in the program code are called dependent parameters. The dependent parameters are used only indirectly in the minimization process. They should be fully defined as functions of the independent parameters. The independent parameters are included in the set **X** of parameters used in the minimization. Be sure that all independent parameters used in your program statements are somehow connected to elements of the model matrices. Otherwise the minimization function does not depend on those independent parameters, and the parameters vary without control (since the corresponding derivative is the constant 0). You also can specify the PARMS statement to set the initial values of all independent parameters used in the program statements that are not defined as elements of model matrices.

ARRAY Statement

ARRAY *arrayname* < (*dimensions*) > < \$ > < *variables and constants* > ;

The ARRAY statement is similar to, but not the same as, the ARRAY statement in the DATA step. The ARRAY statement is used to associate a name with a list of variables and constants. The array name can then be used with subscripts in the program to refer to the items in the list.

The ARRAY statement supported by PROC CALIS does not support all the features of the DATA step ARRAY statement. With PROC CALIS, the ARRAY statement cannot be used to give initial values to array elements. Implicit indexing variables cannot be used; all array references must have explicit subscript expressions. Only exact array dimensions are allowed; lower-bound specifications are not supported. A maximum of six dimensions is allowed.

On the other hand, the ARRAY statement supported by PROC CALIS does allow both variables and constants to be used as array elements. Constant array elements cannot be changed. Both the dimension specification and the list of elements are optional, but at least one must be given. When the list of elements is not given or fewer elements than the size of the array are listed, array variables are created by suffixing element numbers to the array name to complete the element list.

Details

Input Data Sets

You can use four different kinds of input data sets in the CALIS procedure, and you can use them simultaneously. The DATA= data set contains the data to be analyzed, and it can be an ordinary SAS data set containing raw data or a special TYPE=COV, TYPE=UCOV, TYPE=CORR, TYPE=UCORR, TYPE=SYMATRIX, TYPE=SSCP, or TYPE=FACTOR data set containing previously computed statistics. The INEST= data set specifies an input data set that contains initial estimates for the parameters used in the optimization process, and it can also contain boundary and general linear constraints on the parameters. If the model does not change too much, you can use an OUTEST= data set from a previous PROC CALIS analysis; the initial estimates are taken from the values of the PARMS observation. The INRAM= data set names a third input data set that contains all information needed to specify the analysis model in RAM list form (except for user-written program statements). Often the INRAM= data set can be the OUTRAM= data set from a previous PROC CALIS analysis. See the section “OUTRAM= SAS-data-set” on page 525 for the structure of both OUTRAM= and INRAM= data sets. Using the INWGT= data set enables you to read in the weight matrix **W** that can be used in generalized least-squares, weighted least-squares, or diagonally weighted least-squares estimation.

DATA= SAS-data-set

A TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR data set can be created by the CORR procedure or various other procedures. It contains means, standard deviations, the sample size, the covariance or correlation matrix, and possibly other statistics depending on which procedure is used.

If your data set has many observations and you plan to run PROC CALIS several times, you can save computer time by first creating a TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR data set and using it as input to PROC CALIS. For example, assuming that PROC CALIS is first run with an OUTRAM=MOD option, you can run

```
* create TYPE=COV data set;
proc corr cov nocorr data=raw outp=cov(type=cov);
run;
* analysis using correlations;
proc calis data=cov inram=mod;
run;
* analysis using covariances;
proc calis cov data=cov inram=mod;
run;
```

Most procedures automatically set the TYPE= option of an output data set appropriately. However, the CORR procedure sets TYPE=CORR unless an explicit TYPE= option is used. Thus, (TYPE=COV) is needed in the preceding PROC CORR request, since the output data set is a covariance matrix. If you use a DATA step with a SET statement to modify this data set, you must declare the TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR attribute in the new data set.

You can use a VAR statement with PROC CALIS when reading a TYPE=COV, TYPE=UCOV, TYPE=CORR, TYPE=UCORR, or TYPE=SSCP data set to select a subset of the variables or change the order of the variables.

Caution: Problems can arise from using the CORR procedure when there are missing data. By default, PROC CORR computes each covariance or correlation from all observations that have values present for the pair of variables involved (“pairwise deletion”). The resulting covariance or correlation matrix can have negative eigenvalues. A correlation or covariance matrix with negative eigenvalues is recognized as a singular matrix in PROC CALIS, and you cannot compute (default) generalized least-squares or maximum likelihood estimates. You can specify the RIDGE option to ridge the diagonal of such a matrix to obtain a positive definite data matrix. If the NOMISS option is used with the CORR procedure, observations with any missing values are completely omitted from the calculations (“listwise deletion”), and there is no possibility of negative eigenvalues (but still a chance for a singular matrix).

PROC CALIS can also create a TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR data set that includes all the information needed for repeated analyses. If the data set DATA=RAW does not contain missing values, the following statements should give the same PROC CALIS results as the previous example.

```

* using correlations;
proc calis data=raw outstat=cov inram=mod;
run;
* using covariances;
proc calis cov data=cov inram=mod;
run;

```

You can create a TYPE=COV, TYPE=UCOV, TYPE=CORR, TYPE=UCORR, or TYPE=SSCP data set in a DATA step. Be sure to specify the TYPE= option in parentheses after the data set name in the DATA statement, and include the _TYPE_ and _NAME_ variables. If you want to analyze the covariance matrix but your DATA= data set is a TYPE=CORR or TYPE=UCORR data set, you should include an observation with _TYPE_=STD giving the standard deviation of each variable. If you specify the COV option, PROC CALIS analyzes the recomputed covariance matrix:

```

data correl(type=corr);
  input _type_ $ _name_ $ X1-X3;
  datalines;
std    .    4.    2.    8.
corr  X1    1.0    .    .
corr  X2     .7    1.0    .
corr  X3     .5     .4    1.0
;
proc calis cov inram=model;
run;

```

If you want to analyze the UCOV or UCORR matrix but your DATA= data set is a TYPE=COV or TYPE=CORR data set, you should include observations with _TYPE_=STD and _TYPE_=MEAN giving the standard deviation and mean of each variable.

INEST= SAS-data-set

You can use the INEST= (or INVAR= or ESTDATA=) input data set to specify the initial values of the parameters used in the optimization and to specify boundary constraints and the more general linear constraints that can be imposed on these parameters.

The variables of the INEST= data set must correspond to

- a character variable _TYPE_ that indicates the type of the observation
- *n* numeric variables with the parameter names used in the specified PROC CALIS model
- the BY variables that are used in a DATA= input data set
- a numeric variable _RHS_ (right-hand side) (needed only if linear constraints are used)
- additional variables with names corresponding to constants used in the program statements

The content of the `_TYPE_` variable defines the meaning of the observation of the `INEST=` data set. PROC CALIS recognizes observations with the following `_TYPE_` specifications.

PARMS	specifies initial values for parameters that are defined in the model statements of PROC CALIS. The <code>_RHS_</code> variable is not used. Additional variables can contain the values of constants that are referred to in program statements. At the beginning of each run of PROC CALIS, the values of the constants are read from the PARMS observation initializing the constants in the program statements.
UPPERBD UB	specifies upper bounds with nonmissing values. The use of a missing value indicates that no upper bound is specified for the parameter. The <code>_RHS_</code> variable is not used.
LOWERBD LB	specifies lower bounds with nonmissing values. The use of a missing value indicates that no lower bound is specified for the parameter. The <code>_RHS_</code> variable is not used.
LE <= <	specifies the linear constraint $\sum_j a_{ij}x_j \leq b_i$. The n parameter values contain the coefficients a_{ij} , and the <code>_RHS_</code> variable contains the right-hand-side b_i . The use of a missing value indicates a zero coefficient a_{ij} .
GE >= >	specifies the linear constraint $\sum_j a_{ij}x_j \geq b_i$. The n parameter values contain the coefficients a_{ij} , and the <code>_RHS_</code> variable contains the right-hand-side b_i . The use of a missing value indicates a zero coefficient a_{ij} .
EQ =	specifies the linear constraint $\sum_j a_{ij}x_j = b_i$. The n parameter values contain the coefficients a_{ij} , and the <code>_RHS_</code> variable contains the right-hand-side b_i . The use of a missing value indicates a zero coefficient a_{ij} .

The constraints specified in the `INEST=`, `INVAR=`, or `ESTDATA=` data set are added to the constraints specified in `BOUNDS` and `LINCON` statements.

You can use an `OUTEST=` data set from a PROC CALIS run as an `INEST=` data set in a new run. However, be aware that the `OUTEST=` data set also contains the boundary and general linear constraints specified in the previous run of PROC CALIS. When you are using this `OUTEST=` data set without changes as an `INEST=` data set, PROC CALIS adds the constraints from the data set to the constraints specified by a `BOUNDS` and `LINCON` statement. Although PROC CALIS automatically eliminates multiple identical constraints, you should avoid specifying the same constraint a second time.

INRAM= SAS-data-set

This data set is usually created in a previous run of PROC CALIS. It is useful if you want to reanalyze a problem in a different way such as using a different estimation method. You can alter an existing `OUTRAM=` data set, either in the `DATA` step or using the `FSEDIT` procedure, to create the `INRAM=` data set describing a modified

model. For more details on the INRAM= data set, see the section “OUTRAM= SAS-data-set” on page 525.

In the case of a RAM or LINEQS analysis of linear structural equations, the OUTRAM= data set always contains the variable names of the model specified. These variable names and the model specified in the INRAM= data set are the basis of the automatic variable selection algorithm performed after reading the INRAM= data set.

INWGT= SAS-data-set

This data set enables you to specify a weight matrix other than the default matrix for the generalized, weighted, and diagonally weighted least-squares estimation methods. The specification of any INWGT= data set for unweighted least-squares or maximum likelihood estimation is ignored. For generalized and diagonally weighted least-squares estimation, the INWGT= data set must contain a `_TYPE_` and a `_NAME_` variable as well as the manifest variables used in the analysis. The value of the `_NAME_` variable indicates the row index i of the weight w_{ij} . For weighted least squares, the INWGT= data set must contain `_TYPE_`, `_NAME_`, `_NAM2_`, and `_NAM3_` variables as well as the manifest variables used in the analysis. The values of the `_NAME_`, `_NAM2_`, and `_NAM3_` variables indicate the three indices i, j, k of the weight $w_{ij,kl}$. You can store information other than the weight matrix in the INWGT= data set, but only observations with `_TYPE_=WEIGHT` are used to specify the weight matrix **W**. This property enables you to store more than one weight matrix in the INWGT= data set. You can then run PROC CALIS with each of the weight matrices by changing only the `_TYPE_` observation in the INWGT= data set with an intermediate DATA step.

For more details on the INWGT= data set, see the section “OUTWGT= SAS-data-set” on page 530.

Output Data Sets

OUTEST= SAS-data-set

The OUTEST= (or OUTVAR=) data set is of `TYPE=EST` and contains the final parameter estimates, the gradient, the Hessian, and boundary and linear constraints. For `METHOD=ML`, `METHOD=GLS`, and `METHOD=WLS`, the OUTEST= data set also contains the approximate standard errors, the information matrix (crossproduct Jacobian), and the approximate covariance matrix of the parameter estimates ((generalized) inverse of the information matrix). If there are linear or nonlinear equality or active inequality constraints at the solution, the OUTEST= data set also contains Lagrange multipliers, the projected Hessian matrix, and the Hessian matrix of the Lagrange function.

The OUTEST= data set can be used to save the results of an optimization by PROC CALIS for another analysis with either PROC CALIS or another SAS procedure. Saving results to an OUTEST= data set is advised for expensive applications that cannot be repeated without considerable effort.

The OUTEST= data set contains the BY variables, two character variables `_TYPE_` and `_NAME_`, t numeric variables corresponding to the parameters used in the model, a numeric variable `_RHS_` (right-hand side) that is used for the right-hand-

side value b_i of a linear constraint or for the value $f = f(x)$ of the objective function at the final point x^* of the parameter space, and a numeric variable `_ITER_` that is set to zero for initial values, set to the iteration number for the OUTITER output, and set to missing for the result output.

The `_TYPE_` observations in Table 19.5 are available in the OUTEST= data set, depending on the request.

Table 19.5. `_TYPE_` Observations in the OUTEST= data set

<code>_TYPE_</code>	Description								
ACTBC	<p>If there are active boundary constraints at the solution x^*, three observations indicate which of the parameters are actively constrained, as follows.</p> <table> <tr> <th><code>_NAME_</code></th><th>Description</th></tr> <tr> <td>GE</td><td>indicates the active lower bounds</td></tr> <tr> <td>LE</td><td>indicates the active upper bounds</td></tr> <tr> <td>EQ</td><td>indicates the active masks</td></tr> </table>	<code>_NAME_</code>	Description	GE	indicates the active lower bounds	LE	indicates the active upper bounds	EQ	indicates the active masks
<code>_NAME_</code>	Description								
GE	indicates the active lower bounds								
LE	indicates the active upper bounds								
EQ	indicates the active masks								
COV	contains the approximate covariance matrix of the parameter estimates; used in computing the approximate standard errors.								
COVRANK	contains the rank of the covariance matrix of the parameter estimates.								
CRPJ_LF	contains the Hessian matrix of the Lagrange function (based on CRPJAC).								
CRPJAC	contains the approximate Hessian matrix used in the optimization process. This is the inverse of the information matrix.								
EQ	If linear constraints are used, this observation contains the i th linear constraint $\sum_j a_{ij}x_j = b_i$. The parameter variables contain the coefficients a_{ij} , $j = 1, \dots, n$, the <code>_RHS_</code> variable contains b_i , and <code>_NAME_=ACTLC</code> or <code>_NAME_=LDACTLC</code> .								
GE	If linear constraints are used, this observation contains the i th linear constraint $\sum_j a_{ij}x_j \geq b_i$. The parameter variables contain the coefficients a_{ij} , $j = 1, \dots, n$, and the <code>_RHS_</code> variable contains b_i . If the constraint i is active at the solution x^* , then <code>_NAME_=ACTLC</code> or <code>_NAME_=LDACTLC</code> .								
GRAD	contains the gradient of the estimates.								
GRAD_LF	contains the gradient of the Lagrange function. The <code>_RHS_</code> variable contains the value of the Lagrange function.								
HESSIAN	contains the Hessian matrix.								
HESS_LF	contains the Hessian matrix of the Lagrange function (based on HESSIAN).								

Table 19.5. _TYPE_ Observations in the OUTEST= data set (continued)

TYPE	Description										
INFORMAT	contains the information matrix of the parameter estimates (only for METHOD=ML, METHOD=GLS, or METHOD=WLS).										
INITIAL	contains the starting values of the parameter estimates.										
JACNLC	contains the Jacobian of the nonlinear constraints evaluated at the final estimates.										
JACOBIAN	contains the Jacobian matrix (only if the OUTJAC option is used).										
LAGM BC	contains Lagrange multipliers for masks and active boundary constraints.										
	<table> <tr> <th>_NAME_</th><th>Description</th></tr> <tr> <td>GE</td><td>indicates the active lower bounds</td></tr> <tr> <td>LE</td><td>indicates the active upper bounds</td></tr> <tr> <td>EQ</td><td>indicates the active masks</td></tr> </table>	_NAME_	Description	GE	indicates the active lower bounds	LE	indicates the active upper bounds	EQ	indicates the active masks		
NAME	Description										
GE	indicates the active lower bounds										
LE	indicates the active upper bounds										
EQ	indicates the active masks										
LAGM LC	contains Lagrange multipliers for linear equality and active inequality constraints in pairs of observations containing the constraint number and the value of the Lagrange multiplier.										
	<table> <tr> <th>_NAME_</th><th>Description</th></tr> <tr> <td>LEC_NUM</td><td>number of the linear equality constraint</td></tr> <tr> <td>LEC_VAL</td><td>corresponding Lagrange multiplier value</td></tr> <tr> <td>LIC_NUM</td><td>number of the linear inequality constraint</td></tr> <tr> <td>LIC_VAL</td><td>corresponding Lagrange multiplier value</td></tr> </table>	_NAME_	Description	LEC_NUM	number of the linear equality constraint	LEC_VAL	corresponding Lagrange multiplier value	LIC_NUM	number of the linear inequality constraint	LIC_VAL	corresponding Lagrange multiplier value
NAME	Description										
LEC_NUM	number of the linear equality constraint										
LEC_VAL	corresponding Lagrange multiplier value										
LIC_NUM	number of the linear inequality constraint										
LIC_VAL	corresponding Lagrange multiplier value										
LAGM NLC	contains Lagrange multipliers for nonlinear equality and active inequality constraints in pairs of observations containing the constraint number and the value of the Lagrange multiplier.										
	<table> <tr> <th>_NAME_</th><th>Description</th></tr> <tr> <td>NLEC_NUM</td><td>number of the nonlinear equality constraint</td></tr> <tr> <td>NLEC_VAL</td><td>corresponding Lagrange multiplier value</td></tr> <tr> <td>NLIC_NUM</td><td>number of the linear inequality constraint</td></tr> <tr> <td>NLIC_VAL</td><td>corresponding Lagrange multiplier value</td></tr> </table>	_NAME_	Description	NLEC_NUM	number of the nonlinear equality constraint	NLEC_VAL	corresponding Lagrange multiplier value	NLIC_NUM	number of the linear inequality constraint	NLIC_VAL	corresponding Lagrange multiplier value
NAME	Description										
NLEC_NUM	number of the nonlinear equality constraint										
NLEC_VAL	corresponding Lagrange multiplier value										
NLIC_NUM	number of the linear inequality constraint										
NLIC_VAL	corresponding Lagrange multiplier value										
LE	If linear constraints are used, this observation contains the i th linear constraint $\sum_j a_{ij}x_j \leq b_i$. The parameter variables contain the coefficients a_{ij} , $j = 1, \dots, n$, and the _RHS_ variable contains b_i . If the constraint i is active at the solution x^* , then _NAME_=ACTLC or _NAME_=LDACTLC.										
LOWERBD LB	If boundary constraints are used, this observation contains the lower bounds. Those parameters not subjected to lower bounds contain missing values. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.										

Table 19.5. _TYPE_ Observations in the OUTEST= data set (continued)

TYPE	Description								
NACTBC	All parameter variables contain the number n_{abc} of active boundary constraints at the solution x^* . The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.								
NACTLC	All parameter variables contain the number n_{alc} of active linear constraints at the solution x^* that are recognized as linearly independent. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.								
NLC_EQ	contains values and residuals of nonlinear constraints. The _NAME_ variable is described as follows.								
NLC_GE									
NLC_LE									
	<table> <tr> <th>_NAME_</th><th>Description</th></tr> <tr> <td>NLC</td><td>inactive nonlinear constraint</td></tr> <tr> <td>NLCACT</td><td>linear independent active nonlinear constr.</td></tr> <tr> <td>NLCACTLD</td><td>linear dependent active nonlinear constr.</td></tr> </table>	_NAME_	Description	NLC	inactive nonlinear constraint	NLCACT	linear independent active nonlinear constr.	NLCACTLD	linear dependent active nonlinear constr.
NAME	Description								
NLC	inactive nonlinear constraint								
NLCACT	linear independent active nonlinear constr.								
NLCACTLD	linear dependent active nonlinear constr.								
NLDACTBC	contains the number of active boundary constraints at the solution x^* that are recognized as linearly dependent. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.								
NLDACTLC	contains the number of active linear constraints at the solution x^* that are recognized as linearly dependent. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.								
NOBS	contains the number of observations.								
PARMS	contains the final parameter estimates. The _RHS_ variable contains the value of the objective function.								
PCRPJ_LF	contains the projected Hessian matrix of the Lagrange function (based on CRPJAC).								
PHESS_LF	contains the projected Hessian matrix of the Lagrange function (based on HESSIAN).								
PROJCRPJ	contains the projected Hessian matrix (based on CRPJAC).								
PROJGRAD	If linear constraints are used in the estimation, this observation contains the $n - n_{act}$ values of the projected gradient $g_Z = Z'g$ in the variables corresponding to the first $n - n_{act}$ parameters. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.								
PROJHESS	contains the projected Hessian matrix (based on HESSIAN).								
SIGSQ	contains the scalar factor of the covariance matrix of the parameter estimates.								
STDERR	contains approximate standard errors (only for METHOD=ML, METHOD=GLS, or METHOD=WLS).								

Table 19.5. _TYPE_ Observations in the OUTEST= data set (continued)

TYPE	Description
TERMINAT	The _NAME_ variable contains the name of the termination criterion.
UPPERBD UB	If boundary constraints are used, this observation contains the upper bounds. Those parameters not subjected to upper bounds contain missing values. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.

If the technique specified by the TECH= option cannot be performed (for example, no feasible initial values can be computed, or the function value or derivatives cannot be evaluated at the starting point), the OUTEST= data set may contain only some of the observations (usually only the PARMS and GRAD observations).

OUTRAM= SAS-data-set

The OUTRAM= data set is of TYPE=RAM and contains the model specification and the computed parameter estimates. This data set is intended to be reused as an INRAM= data set to specify good initial values in a subsequent analysis by PROC CALIS. For a structural equation model, after some alterations, this data set can be used for plotting a path diagram by the NETDRAW procedure, a SAS/OR procedure.

The OUTRAM= data set contains the following variables:

- the BY variables, if any
- the character variable _TYPE_, which takes the values MODEL, ESTIM, VARNAME, METHOD, and STAT
- six additional variables whose meaning depends on the _TYPE_ of the observation

Each observation with _TYPE_ =MODEL defines one matrix in the generalized COSAN model. The additional variables are as follows.

Table 19.6. Additional Variables when _TYPE_=MODEL

Variable	Contents
NAME	name of the matrix (character)
MATNR	number for the term and matrix in the model (numeric)
ROW	matrix row number (numeric)
COL	matrix column number (numeric)
ESTIM	first matrix type (numeric)
STDERR	second matrix type (numeric)

If the generalized COSAN model has only one matrix term, the _MATNR_ variable contains only the number of the matrix in the term. If there is more than one term, then it is the term number multiplied by 10,000 plus the matrix number (assuming that there are no more than 9,999 matrices specified in the COSAN model statement).

Each observation with `_TYPE_ = ESTIM` defines one element of a matrix in the generalized COSAN model. The variables are used as follows.

Table 19.7. Additional Variables when `_TYPE_ = ESTIM`

Variable	Contents
<code>_NAME_</code>	name of the parameter (character)
<code>_MATNR_</code>	term and matrix location of parameter (numeric)
<code>_ROW_</code>	row location of parameter (numeric)
<code>_COL_</code>	column location of parameter (numeric)
<code>_ESTIM_</code>	parameter estimate or constant value (numeric)
<code>_STDERR_</code>	standard error of estimate (numeric)

For constants rather than estimates, the `_STDERR_` variable is 0. The `_STDERR_` variable is missing for ULS and DWLS estimates if `NOSTDERR` is specified or if the approximate standard errors are not computed.

Each observation with `_TYPE_ = VARNAME` defines a column variable name of a matrix in the generalized COSAN model.

The observations with `_TYPE_ = METHOD` and `_TYPE_ = STAT` are not used to build the model. The `_TYPE_ = METHOD` observation contains the name of the estimation method used to compute the parameter estimates in the `_NAME_` variable. If `METHOD = NONE` is not specified, the `_ESTIM_` variable of the `_TYPE_ = STAT` observations contains the information summarized in Table 19.8 (described in the section “Assessment of Fit” on page 536).

Table 19.8. `_ESTIM_` Contents for `_TYPE_ = STAT`

<code>_NAME_</code>	<code>_ESTIM_</code>
N	sample size
NPARAM	number of parameters used in the model
DF	degrees of freedom
N_ACT	number of active boundary constraints for ML, GLS, and WLS estimation
FIT	fit function
GFI	goodness-of-fit index (GFI)
AGFI	adjusted GFI for degrees of freedom
RMR	root mean square residual
PGFI	parsimonious GFI of Mulaik et al. (1989)
CHISQUAR	overall χ^2
P_CHISQ	probability $> \chi^2$
CHISQNUL	null (baseline) model χ^2
RMSEAEST	Steiger & Lind’s (1980) RMSEA index estimate
RMSEALOB	lower range of RMSEA confidence interval
RMSEAUPB	upper range of RMSEA confidence interval
P_CLOSFT	Browne & Cudeck’s (1993) probability of close fit
ECVI_EST	Browne & Cudeck’s (1993) ECVI index estimate
ECVI_LOB	lower range of ECVI confidence interval
ECVI_UPB	upper range of ECVI confidence interval
COMPFITI	Bentler’s (1989) comparative fit index

Table 19.8. _ESTIM_ Contents for _TYPE_=STAT (continued)

NAME	_ESTIM_
ADJCHISQ	adjusted χ^2 for elliptic distribution
P_ACHISQ	probability corresponding adjusted χ^2
RLSCHISQ	reweighted least-squares χ^2 (only ML estimation)
AIC	Akaike's information criterion
CAIC	Bozdogan's consistent information criterion
SBC	Schwarz's Bayesian criterion
CENTRALI	McDonald's centrality criterion
PARSIMON	Parsimonious index of James, Mulaik, and Brett
ZTESTWH	z test of Wilson and Hilferty
BB_NONOR	Bentler-Bonett (1980) nonnormed index ρ
BB_NORMD	Bentler-Bonett (1980) normed index Δ
BOL_RHO1	Bollen's (1986) normed index ρ_1
BOL_DEL2	Bollen's (1989a) nonnormed index Δ_2
CNHOELT	Hoelter's critical N index

You can edit the OUTRAM= data set to use its contents for initial estimates in a subsequent analysis by PROC CALIS, perhaps with a slightly changed model. But you should be especially careful for _TYPE_=MODEL when changing matrix types. The codes for the two matrix types are listed in Table 19.9.

Table 19.9. Matrix Type Codes

Code	First Matrix Type	Description
1:	IDE	identity matrix
2:	ZID	zero:identity matrix
3:	DIA	diagonal matrix
4:	ZDI	zero:diagonal matrix
5:	LOW	lower triangular matrix
6:	UPP	upper triangular matrix
7:		temporarily not used
8:	SYM	symmetric matrix
9:	GEN	general-type matrix
10:	BET	identity minus general-type matrix
11:	PER	selection matrix
12:		first matrix (J) in LINEQS model statement
13:		second matrix (β) in LINEQS model statement
14:		third matrix (γ) in LINEQS model statement
Code	Second Matrix Type	Description
0:		noninverse model matrix
1:	INV	inverse model matrix
2:	IMI	'identity minus inverse' model matrix

OUTSTAT= SAS-data-set

The OUTSTAT= data set is similar to the TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR data set produced by the CORR procedure. The OUTSTAT= data set contains the following variables:

- the BY variables, if any
- two character variables, _TYPE_ and _NAME_
- the variables analyzed, that is, those in the VAR statement, or if there is no VAR statement, all numeric variables not listed in any other statement but used in the analysis. (**Caution:** Using the LINEQS or RAM model statements selects variables automatically.)

The OUTSTAT= data set contains the following information (when available):

- the mean and standard deviation
- the skewness and kurtosis (if the DATA= data set is a raw data set and the KURTOSIS option is specified)
- the number of observations
- if the WEIGHT statement is used, sum of the weights
- the correlation or covariance matrix to be analyzed
- the predicted correlation or covariance matrix
- the standardized or normalized residual correlation or covariance matrix
- if the model contains latent variables, the predicted covariances between latent and manifest variables, and the latent variable (or factor) score regression coefficients (see the PLATCOV display option on page 473)

In addition, if the FACTOR model statement is used, the OUTSTAT= data set contains:

- the unrotated factor loadings, the unique variances, and the matrix of factor correlations
- the rotated factor loadings and the transformation matrix of the rotation
- the matrix of standardized factor loadings

Each observation in the OUTSTAT= data set contains some type of statistic as indicated by the _TYPE_ variable. The values of the _TYPE_ variable are given in Table 19.10.

Table 19.10. _TYPE_ Observations in the OUTSTAT= data set

TYPE	Contents
MEAN	means
STD	standard deviations
USTD	uncorrected standard deviations
SKEWNESS	univariate skewness
KURTOSIS	univariate kurtosis
N	sample size
SUMWGT	sum of weights (if WEIGHT statement is used)
COV	covariances analyzed
CORR	correlations analyzed
UCOV	uncorrected covariances analyzed
UCORR	uncorrected correlations analyzed
ULSPRED	ULS predicted model values
GLSPRED	GLS predicted model values
MAXPRED	ML predicted model values
WLSPRED	WLS predicted model values
DWLSPRED	DWLS predicted model values
ULSNRES	ULS normalized residuals
GLSNRES	GLS normalized residuals
MAXNRES	ML normalized residuals
WLSNRES	WLS normalized residuals
DWLSNRES	DWLS normalized residuals
ULSSRES	ULS variance standardized residuals
GLSSRES	GLS variance standardized residuals
MAXSRES	ML variance standardized residuals
WLSSRES	WLS variance standardized residuals
DWLSSRES	DWLS variance standardized residuals
ULSASRES	ULS asymptotically standardized residuals
GLSASRES	GLS asymptotically standardized residuals
MAXASRES	ML asymptotically standardized residuals
WLSASRES	WLS asymptotically standardized residuals
DWLSASRS	DWLS asymptotically standardized residuals
UNROTATE	unrotated factor loadings
FCORR	matrix of factor correlations
UNIQUE_V	unique variances
TRANSFOR	transformation matrix of rotation
LOADINGS	rotated factor loadings
STD_LOAD	standardized factor loadings
LSSCORE	latent variable (or factor) score regression coefficients for ULS method
SCORE	latent variable (or factor) score regression coefficients other than ULS method

The `_NAME_` variable contains the name of the manifest variable corresponding to each row for the covariance, correlation, predicted, and residual matrices and contains the name of the latent variable in case of factor regression scores. For other observations, `_NAME_` is blank.

The unique variances and rotated loadings can be used as starting values in more difficult and constrained analyses.

If the model contains latent variables, the `OUTSTAT=` data set also contains the latent variable score regression coefficients and the predicted covariances between latent and manifest variables. You can use the latent variable score regression coefficients with `PROC SCORE` to compute factor scores.

If the analyzed matrix is a (corrected or uncorrected) covariance rather than a correlation matrix, the `_TYPE_=STD` or `_TYPE_=USTD` observation is not included in the `OUTSTAT=` data set. In this case, the standard deviations can be obtained from the diagonal elements of the covariance matrix. Dropping the `_TYPE_=STD` or `_TYPE_=USTD` observation prevents `PROC SCORE` from standardizing the observations before computing the factor scores.

OUTWGT= SAS-data-set

You can create an `OUTWGT=` data set that is of `TYPE=WEIGHT` and contains the weight matrix used in generalized, weighted, or diagonally weighted least-squares estimation. The *inverse* of the weight matrix is used in the corresponding fit function. The `OUTWGT=` data set contains the weight matrix on which the `WRIDGE=` and the `WPENALTY=` options are applied. For unweighted least-squares or maximum likelihood estimation, no `OUTWGT=` data set can be written. The last weight matrix used in maximum likelihood estimation is the predicted model matrix (observations with `_TYPE_ =MAXPRED`) that is included in the `OUTSTAT=` data set.

For generalized and diagonally weighted least-squares estimation, the weight matrices **W** of the `OUTWGT=` data set contain all elements w_{ij} , where the indices i and j correspond to all manifest variables used in the analysis. Let $varnam_i$ be the name of the i th variable in the analysis. In this case, the `OUTWGT=` data set contains n observations with variables as displayed in the following table.

Table 19.11. Contents of `OUTWGT=` data set for GLS and DWLS Estimation

Variable	Contents
<code>_TYPE_</code>	WEIGHT (character)
<code>_NAME_</code>	name of variable $varnam_i$ (character)
$varnam_1$	weight w_{i1} for variable $varnam_1$ (numeric)
\vdots	\vdots
$varnam_n$	weight w_{in} for variable $varnam_n$ (numeric)

For weighted least-squares estimation, the weight matrix **W** of the `OUTWGT=` data set contains only the nonredundant elements $w_{ij,kl}$. In this case, the `OUTWGT=` data set contains $n(n+1)(2n+1)/6$ observations with variables as follows.

Table 19.12. Contents of OUTWGT= data set for WLS Estimation

Variable	Contents
TYPE	WEIGHT (character)
NAME	name of variable $varnam_i$ (character)
NAM2	name of variable $varnam_j$ (character)
NAM3	name of variable $varnam_k$ (character)
$varnam_1$	weight $w_{ij,k1}$ for variable $varnam_1$ (numeric)
⋮	⋮
$varnam_n$	weight $w_{ij,kn}$ for variable $varnam_n$ (numeric)

Symmetric redundant elements are set to missing values.

Missing Values

If the DATA= data set contains raw data (rather than a covariance or correlation matrix), observations with missing values for any variables in the analysis are omitted from the computations. If a covariance or correlation matrix is read, missing values are allowed as long as every pair of variables has at least one nonmissing value.

Estimation Criteria

The following five estimation methods are available in PROC CALIS:

- unweighted least squares (ULS)
- generalized least squares (GLS)
- normal-theory maximum likelihood (ML)
- weighted least squares (WLS, ADF)
- diagonally weighted least squares (DWLS)

An INWGT= data set can be used to specify other than the default weight matrices **W** for GLS, WLS, and DWLS estimation.

In each case, the parameter vector is estimated iteratively by a nonlinear optimization algorithm that optimizes a goodness-of-fit function F . When n denotes the number of manifest variables, **S** denotes the given sample covariance or correlation matrix for a sample with size N , and **C** denotes the predicted moment matrix, then the fit function for unweighted least-squares estimation is

$$F_{ULS} = .5Tr[(\mathbf{S} - \mathbf{C})^2]$$

For normal-theory generalized least-squares estimation, the function is

$$F_{GLS} = .5Tr[(\mathbf{S}^{-1}(\mathbf{S} - \mathbf{C}))^2]$$

For normal-theory maximum likelihood estimation, the function is

$$F_{ML} = Tr(\mathbf{S}\mathbf{C}^{-1}) - n + \ln(\det(\mathbf{C})) - \ln(\det(\mathbf{S}))$$

The first three functions can be expressed by the generally weighted least-squares criterion (Browne 1982):

$$F_{GMLS} = .5Tr[(\mathbf{W}^{-1}(\mathbf{S} - \mathbf{C}))^2]$$

For unweighted least squares, the weight matrix \mathbf{W} is chosen as the identity matrix \mathbf{I} ; for generalized least squares, the default weight matrix \mathbf{W} is the sample covariance matrix \mathbf{S} ; and for normal-theory maximum likelihood, \mathbf{W} is the iteratively updated predicted moment matrix \mathbf{C} . The values of the normal-theory maximum likelihood function F_{ML} and the generally weighted least-squares criterion F_{GMLS} with $\mathbf{W} = \mathbf{C}$ are asymptotically equivalent.

The goodness-of-fit function that is minimized in weighted least-squares estimation is

$$F_{WLS} = Vec(s_{ij} - c_{ij})' \mathbf{W}^{-1} Vec(s_{ij} - c_{ij})$$

where $Vec(s_{ij} - c_{ij})$ denotes the vector of the $n(n+1)/2$ elements of the lower triangle of the symmetric matrix $\mathbf{S} - \mathbf{C}$, and $\mathbf{W} = (w_{ij,kl})$ is a positive definite symmetric matrix with $n(n+1)/2$ rows and columns.

If the moment matrix \mathbf{S} is considered as a covariance rather than a correlation matrix, the default setting of $\mathbf{W} = (w_{ij,kl})$ is the consistent but biased estimators of the asymptotic covariances $\sigma_{ij,kl}$ of the sample covariance s_{ij} with the sample covariance s_{kl}

$$w_{ij,kl} = s_{ij,kl} - s_{ij}s_{kl}$$

where

$$s_{ij,kl} = \frac{1}{N} \sum_{r=1}^N (z_{ri} - \bar{z}_i)(z_{rj} - \bar{z}_j)(z_{rk} - \bar{z}_k)(z_{rl} - \bar{z}_l)$$

The formula of the asymptotic covariances of uncorrected covariances (using the UCOV or NOINT option) is a straightforward generalization of this expression.

The resulting weight matrix \mathbf{W} is at least positive semidefinite (except for rounding errors). Using the ASYCOV option, you can use Browne's (1984, formula (3.8)) unbiased estimators

$$\begin{aligned} w_{ij,kl} = & \frac{N(N-1)}{(N-2)(N-3)}(s_{ij,kl} - s_{ij}s_{kl}) \\ & - \frac{N}{(N-2)(N-3)}(s_{ik}s_{jl} + s_{il}s_{jk} - \frac{2}{N-1}s_{ij}s_{kl}) \end{aligned}$$

There is no guarantee that this weight matrix is positive semidefinite. However, the second part is of order $O(N^{-1})$ and does not destroy the positive semidefinite first

part for sufficiently large N . For a large number of independent observations, default settings of the weight matrix \mathbf{W} result in asymptotically distribution-free parameter estimates with unbiased standard errors and a correct χ^2 test statistic (Browne 1982, 1984).

If the moment matrix \mathbf{S} is a correlation (rather than a covariance) matrix, the default setting of $\mathbf{W} = (w_{ij,kl})$ is the estimators of the asymptotic covariances $\sigma_{ij,kl}$ of the correlations $\mathbf{S} = (s_{ij})$ (Browne and Shapiro 1986; DeLeeuw 1983)

$$w_{ij,kl} = r_{ij,kl} - \frac{1}{2}r_{ij}(r_{ii,kl} + r_{jj,kl}) - \frac{1}{2}r_{kl}(r_{kk,ij} + r_{ll,ij}) \\ + \frac{1}{4}r_{ij}r_{kl}(r_{ii,kk} + r_{ii,ll} + r_{jj,kk} + r_{jj,ll})$$

where

$$r_{ij,kl} = \frac{s_{ij,kl}}{\sqrt{s_{ii}s_{jj}s_{kk}s_{ll}}}$$

The asymptotic variances of the diagonal elements of a correlation matrix are 0. Therefore, the weight matrix computed by Browne and Shapiro's formula is always singular. In this case the goodness-of-fit function for weighted least-squares estimation is modified to

$$F_{WLS} = \sum_{i=2}^n \sum_{j=1}^{i-1} \sum_{k=2}^n \sum_{l=1}^{k-1} w^{ij,kl} (s_{ij} - c_{ij})(s_{kl} - c_{kl}) + r \sum_i^n (s_{ii} - c_{ii})^2$$

where r is the penalty weight specified by the WPENALTY= r option and the $w^{ij,kl}$ are the elements of the inverse of the reduced $(n(n-1)/2) \times (n(n-1)/2)$ weight matrix that contains only the nonzero rows and columns of the full weight matrix \mathbf{W} . The second term is a penalty term to fit the diagonal elements of the moment matrix \mathbf{S} . The default value of $r = 100$ can be decreased or increased by the WPENALTY= option. The often used value of $r = 1$ seems to be too small in many cases to fit the diagonal elements of a correlation matrix properly. If your model does not fit the diagonal of the moment matrix \mathbf{S} , you can specify the NODIAG option to exclude the diagonal elements from the fit function.

Storing and inverting the huge weight matrix \mathbf{W} in WLS estimation needs considerable computer resources. A compromise is found by implementing the DWLS method that uses only the diagonal of the weight matrix \mathbf{W} from the WLS estimation in the minimization function

$$F_{DWLS} = \text{Vec}(s_{ij} - c_{ij})' \text{diag}(\mathbf{W})^{-1} \text{Vec}(s_{ij} - c_{ij})$$

The statistical properties of DWLS estimates are still not known.

In generalized, weighted, or diagonally weighted least-squares estimation, you can change from the default settings of weight matrices \mathbf{W} by using an INWGT= data set. Because the diagonal elements $w_{ii,kk}$ of the weight matrix \mathbf{W} are interpreted as asymptotic variances of the sample covariances or correlations, they cannot be negative. The CALIS procedure requires a positive definite weight matrix that has positive diagonal elements.

Relationships among Estimation Criteria

The five estimation functions, F_{ULS} , F_{GLS} , F_{ML} , F_{WLS} , and F_{DWLS} , belong to the following two groups:

- The functions F_{ULS} , F_{GLS} , and F_{ML} take into account all n^2 elements of the symmetric residual matrix $\mathbf{S} - \mathbf{C}$. This means that the off-diagonal residuals contribute twice to F , as lower and as upper triangle elements.
- The functions F_{WLS} and F_{DWLS} take into account only the $n(n+1)/2$ lower triangular elements of the symmetric residual matrix $\mathbf{S} - \mathbf{C}$. This means that the off-diagonal residuals contribute to F only once.

The F_{DWLS} function used in PROC CALIS differs from that used by the LISREL 7 program. Formula (1.25) of the LISREL 7 manual (Jöreskog and Sörbom 1988, p. 23) shows that LISREL groups the F_{DWLS} function in the first group by taking into account all n^2 elements of the symmetric residual matrix $\mathbf{S} - \mathbf{C}$.

- Relationship between DWLS and WLS:
PROC CALIS: The F_{DWLS} and F_{WLS} estimation functions deliver the same results for the special case that the weight matrix \mathbf{W} used by WLS estimation is a diagonal matrix.
LISREL 7: This is not the case.
- Relationship between DWLS and ULS:
LISREL 7: The F_{DWLS} and F_{ULS} estimation functions deliver the same results for the special case that the diagonal weight matrix \mathbf{W} used by DWLS estimation is an identity matrix (contains only 1s).
PROC CALIS: To obtain the same results with F_{DWLS} and F_{ULS} estimation, set the diagonal weight matrix \mathbf{W} used in DWLS estimation to

$$w_{ii,kk} = \begin{cases} 1. & \text{if } i = k \\ 0.5 & \text{otherwise} \end{cases}$$

Because the reciprocal elements of the weight matrix are used in the goodness-of-fit function, the off-diagonal residuals are weighted by a factor of 2.

Testing Rank Deficiency in the Approximate Covariance Matrix

The inverse of the information matrix (or approximate Hessian matrix) is used for the covariance matrix of the parameter estimates, which is needed for the computation of approximate standard errors and modification indices. The numerical condition of the information matrix (computed as the crossproduct $\mathbf{J}'\mathbf{J}$ of the Jacobian matrix \mathbf{J}) can be very poor in many practical applications, especially for the analysis of unscaled covariance data. The following four-step strategy is used for the inversion of the information matrix.

1. The inversion (usually of a normalized matrix $\mathbf{D}^{-1}\mathbf{H}\mathbf{D}^{-1}$) is tried using a modified form of the Bunch and Kaufman (1977) algorithm, which allows the

specification of a different singularity criterion for each pivot. The following three criteria for the detection of rank loss in the information matrix are used to specify thresholds:

- *ASING* specifies absolute singularity.
- *MSING* specifies relative singularity depending on the whole matrix norm.
- *VSING* specifies relative singularity depending on the column matrix norm.

If no rank loss is detected, the inverse of the information matrix is used for the covariance matrix of parameter estimates, and the next two steps are skipped.

2. The linear dependencies among the parameter subsets are displayed based on the singularity criteria.
3. If the number of parameters t is smaller than the value specified by the *G4=* option (the default value is 60), the Moore-Penrose inverse is computed based on the eigenvalue decomposition of the information matrix. If you do not specify the *NOPRINT* option, the distribution of eigenvalues is displayed, and those eigenvalues that are set to zero in the Moore-Penrose inverse are indicated. You should inspect this eigenvalue distribution carefully.
4. If PROC CALIS did not set the right subset of eigenvalues to zero, you can specify the *COVSING=* option to set a larger or smaller subset of eigenvalues to zero in a further run of PROC CALIS.

Approximate Standard Errors

Except for unweighted and diagonally weighted least-squares estimation, approximate standard errors can be computed as the diagonal elements of the matrix

$$\frac{c}{NM} \mathbf{H}^{-1}, \quad \text{where}$$

$$NM = \begin{cases} (N - 1) & \text{if the CORR or COV matrix is analyzed} \\ & \text{or the intercept variable is not used in the model} \\ N & \text{if the UCORR or UCOV matrix is analyzed} \\ & \text{and the intercept variable is not used in the model} \end{cases}$$

The matrix \mathbf{H} is the approximate Hessian matrix of F evaluated at the final estimates, $c = 1$ for the WLS estimation method, $c = 2$ for the GLS and ML method, and N is the sample size. If a given correlation or covariance matrix is singular, PROC CALIS offers two ways to compute a generalized inverse of the information matrix and, therefore, two ways to compute approximate standard errors of implicitly constrained parameter estimates, t values, and modification indices. Depending on the *G4=* specification, either a Moore-Penrose inverse or a *G2* inverse is computed. The expensive Moore-Penrose inverse computes an estimate of the null space using an eigenvalue decomposition. The cheaper *G2* inverse is produced by sweeping the linearly independent rows and columns and zeroing out the dependent ones. The

information matrix, the approximate covariance matrix of the parameter estimates, and the approximate standard errors are not computed in the cases of unweighted or diagonally weighted least-squares estimation.

Assessment of Fit

This section contains a collection of formulas used in computing indices to assess the goodness of fit by PROC CALIS. The following notation is used:

- N for the sample size
- n for the number of manifest variables
- t for the number of parameters to estimate
- $NM = \begin{cases} (N - 1) & \text{if the CORR or COV matrix is analyzed} \\ & \text{or the intercept variable is not used in the model} \\ N & \text{if the UCORR or UCOV matrix is analyzed} \\ & \text{and the intercept variable is not used in the model} \end{cases}$
- df for the degrees of freedom
- $\gamma = \mathbf{X}$ for the t vector of optimal parameter estimates
- $\mathbf{S} = (s_{ij})$ for the $n \times n$ input COV, CORR, UCOV, or UCORR matrix
- $\mathbf{C} = (c_{ij}) = \hat{\Sigma} = \Sigma(\hat{\gamma})$ for the predicted model matrix
- \mathbf{W} for the weight matrix ($\mathbf{W} = \mathbf{I}$ for ULS, $\mathbf{W} = \mathbf{S}$ for default GLS, and $\mathbf{W} = \mathbf{C}$ for ML estimates)
- \mathbf{U} for the $n^2 \times n^2$ asymptotic covariance matrix of sample covariances
- $\Phi(x|\lambda, df)$ for the cumulative distribution function of the noncentral chi-squared distribution with noncentrality parameter λ

The following notation is for indices that allow testing nested models by a χ^2 difference test:

- f_0 for the function value of the independence model
- df_0 for the degrees of freedom of the independence model
- $f_{min} = F$ for the function value of the fitted model
- $df_{min} = df$ for the degrees of freedom of the fitted model

The degrees of freedom df_{min} and the number of parameters t are adjusted automatically when there are active constraints in the analysis. The computation of many fit statistics and indices are affected. You can turn off the automatic adjustment using the NOADJDF option. See the section “Counting the Degrees of Freedom” on page 563 for more information.

Residuals

PROC CALIS computes four types of residuals and writes them to the OUTSTAT= data set.

- **Raw Residuals**

$$Res = \mathbf{S} - \mathbf{C}, Res_{ij} = s_{ij} - c_{ij}$$

The raw residuals are displayed whenever the PALL, the PRINT, or the RESIDUAL option is specified.

- **Variance Standardized Residuals**

$$VSR_{ij} = \frac{s_{ij} - c_{ij}}{\sqrt{s_{ii}s_{jj}}}$$

The variance standardized residuals are displayed when you specify

- the PALL, the PRINT, or the RESIDUAL option and METHOD=NONE, METHOD=ULS, or METHOD=DWLS
- RESIDUAL=VARSTAND

The variance standardized residuals are equal to those computed by the EQS 3 program (Bentler 1989).

- **Asymptotically Standardized Residuals**

$$ASR_{ij} = \frac{s_{ij} - c_{ij}}{\sqrt{c_{ij,ij}}}, \quad \text{where}$$

$$c_{ij,ij} = \text{diag}(\mathbf{U} - \mathbf{J} \text{Cov}(\boldsymbol{\gamma}) \mathbf{J}')_{ij}$$

The matrix \mathbf{J} is the $n^2 \times t$ Jacobian matrix $d\boldsymbol{\Sigma}/d\boldsymbol{\gamma}$, and $\text{Cov}(\boldsymbol{\gamma})$ is the $t \times t$ asymptotic covariance matrix of parameter estimates (the inverse of the information matrix). Asymptotically standardized residuals are displayed when one of the following conditions is met:

- The PALL, the PRINT, or the RESIDUAL option is specified, and METHOD=ML, METHOD=GLS, or METHOD=WLS, and the expensive information and Jacobian matrices are computed for some other reason.
- RESIDUAL=ASYSTAND is specified.

The asymptotically standardized residuals are equal to those computed by the LISREL 7 program (Jöreskog and Sörbom 1988) except for the denominator NM in the definition of matrix \mathbf{U} .

- **Normalized Residuals**

$$NR_{ij} = \frac{s_{ij} - c_{ij}}{\sqrt{u_{ij,ij}}}$$

where the diagonal elements $u_{ij,ij}$ of the $n^2 \times n^2$ asymptotic covariance matrix \mathbf{U} of sample covariances are defined for the following methods.

- **GLS** as $u_{ij,ij} = \frac{1}{NM}(s_{ii}s_{jj} + s_{ij}^2)$
- **ML** as $u_{ij,ij} = \frac{1}{NM}(c_{ii}c_{jj} + c_{ij}^2)$
- **WLS** as $u_{ij,ij} = w_{ij,ij}$

Normalized residuals are displayed when one of the following conditions is met:

- The **PALL**, the **PRINT**, or the **RESIDUAL** option is specified, and **METHOD=ML**, **METHOD=GLS**, or **METHOD=WLS**, and the expensive information and Jacobian matrices are **not** computed for some other reason.
- **RESIDUAL=NORM** is specified.

The normalized residuals are equal to those computed by the LISREL VI program (Jöreskog and Sörbom 1985) except for the definition of the denominator NM in matrix **U**.

For estimation methods that are not BGLS estimation methods (Browne 1982, 1984), such as **METHOD=NONE**, **METHOD=ULS**, or **METHOD=DWLS**, the assumption of an asymptotic covariance matrix **U** of sample covariances does not seem to be appropriate. In this case, the normalized residuals should be replaced by the more relaxed variance standardized residuals. Computation of asymptotically standardized residuals requires computing the Jacobian and information matrices. This is computationally very expensive and is done only if the Jacobian matrix has to be computed for some other reason, that is, if at least one of the following items is true:

- The default, **PRINT**, or **PALL** displayed output is requested, and neither the **NOMOD** nor **NOSTDERR** option is specified.
- Either the **MODIFICATION** (included in **PALL**), **PCOVES**, or **STDERR** (included in default, **PRINT**, and **PALL** output) option is requested or **RESIDUAL=ASYSTAND** is specified.
- The **LEVMAR** or **NEWWRAP** optimization technique is used.
- An **OUTRAM=** data set is specified without using the **NOSTDERR** option.
- An **OUTEST=** data set is specified without using the **NOSTDERR** option.

Since normalized residuals use an overestimate of the asymptotic covariance matrix of residuals (the diagonal of **U**), the normalized residuals cannot be larger than the asymptotically standardized residuals (which use the diagonal of $\mathbf{U} - \mathbf{J}Cov(\boldsymbol{\gamma})\mathbf{J}'$).

Together with the residual matrices, the values of the average residual, the average off-diagonal residual, and the rank order of the largest values are displayed. The distribution of the normalized and standardized residuals is displayed also.

Goodness-of-Fit Indices Based on Residuals

The following items are computed for all five kinds of estimation: **ULS**, **GLS**, **ML**, **WLS**, and **DWLS**. All these indices are written to the **OUTRAM=** data set. The goodness of fit (**GFI**), adjusted goodness of fit (**AGFI**), and root mean square residual (**RMR**) are computed as in the LISREL VI program of Jöreskog and Sörbom (1985).

- **Goodness-of-Fit Index**

The goodness-of-fit index for the ULS, GLS, and ML estimation methods is

$$GFI = 1 - \frac{Tr((\mathbf{W}^{-1}(\mathbf{S} - \mathbf{C}))^2)}{Tr((\mathbf{W}^{-1}\mathbf{S})^2)}$$

but for WLS and DWLS estimation, it is

$$GFI = 1 - \frac{Vec(s_{ij} - c_{ij})' \mathbf{W}^{-1} Vec(s_{ij} - c_{ij})}{Vec(s_{ij})' \mathbf{W}^{-1} Vec(s_{ij})}$$

where $\mathbf{W} = diag$ for DWLS estimation, and $Vec(s_{ij} - c_{ij})$ denotes the vector of the $n(n+1)/2$ elements of the lower triangle of the symmetric matrix $\mathbf{S} - \mathbf{C}$. For a constant weight matrix \mathbf{W} , the goodness-of-fit index is 1 minus the ratio of the minimum function value and the function value before any model has been fitted. The GFI should be between 0 and 1. The data probably do not fit the model if the GFI is negative or much larger than 1.

- **Adjusted Goodness-of-Fit Index**

The AGFI is the GFI adjusted for the degrees of freedom of the model

$$AGFI = 1 - \frac{n(n+1)}{2df}(1 - GFI)$$

The AGFI corresponds to the GFI in replacing the total sum of squares by the mean sum of squares.

Caution:

- Large n and small df can result in a negative AGFI. For example, $GFI=0.90$, $n=19$, and $df=2$ result in an AGFI of -8.5.
- AGFI is not defined for a saturated model, due to division by $df = 0$.
- AGFI is not sensitive to losses in df .

The AGFI should be between 0 and 1. The data probably do not fit the model if the AGFI is negative or much larger than 1. For more information, refer to Mulaik et al. (1989).

- **Root Mean Square Residual**

The RMR is the mean of the squared residuals:

$$RMR = \sqrt{\frac{2}{n(n+1)} \sum_i^n \sum_j^i (s_{ij} - c_{ij})^2}$$

- **Parsimonious Goodness-of-Fit Index**

The PGFI (Mulaik et al. 1989) is a modification of the GFI that takes the parsimony of the model into account:

$$PGFI = \frac{df_{min}}{df_0} GFI$$

The PGFI uses the same parsimonious factor as the parsimonious normed Bentler-Bonett index (James, Mulaik, and Brett 1982).

Goodness-of-Fit Indices Based on the χ^2

The following items are transformations of the overall χ^2 value and in general depend on the sample size N . These indices are not computed for ULS or DWLS estimates.

- **Uncorrected χ^2**

The overall χ^2 measure is the optimum function value F multiplied by $N - 1$ if a CORR or COV matrix is analyzed, or multiplied by N if a UCORR or UCOV matrix is analyzed. This gives the likelihood ratio test statistic for the null hypothesis that the predicted matrix \mathbf{C} has the specified model structure against the alternative that \mathbf{C} is unconstrained. The χ^2 test is valid only if the observations are independent and identically distributed, the analysis is based on the nonstandardized sample covariance matrix \mathbf{S} , and the sample size N is sufficiently large (Browne 1982; Bollen 1989b; Jöreskog and Sörbom 1985). For ML and GLS estimates, the variables must also have an approximately multivariate normal distribution. The notation Prob>Chi**2 means “the probability under the null hypothesis of obtaining a greater χ^2 statistic than that observed.”

$$\chi^2 = NM * F$$

where F is the function value at the minimum.

- **χ_0^2 Value of the Independence Model**

The χ_0^2 value of the independence model

$$\chi_0^2 = NM * f_0$$

and the corresponding degrees of freedom df_0 can be used (in large samples) to evaluate the gain of explanation by fitting the specific model (Bentler 1989).

- **RMSEA Index (Steiger and Lind 1980)**

The Steiger and Lind (1980) root mean squared error approximation (RMSEA) coefficient is

$$\epsilon_\alpha = \sqrt{\max\left(\frac{F}{df} - \frac{1}{NM}, 0\right)}$$

The lower and upper limits of the confidence interval are computed using the cumulative distribution function of the noncentral chi-squared distribution $\Phi(x|\lambda, df) = \alpha$, with $x = NM * F$, λ_L satisfying $\Phi(x|\lambda_L, df) = 1 - \frac{\alpha}{2}$, and λ_U satisfying $\Phi(x|\lambda_U, df) = \frac{\alpha}{2}$:

$$(\epsilon_{\alpha_L}; \epsilon_{\alpha_U}) = \left(\sqrt{\frac{\lambda_L}{NM * df}}, \sqrt{\frac{\lambda_U}{NM * df}}\right)$$

Refer to Browne and Du Toit (1992) for more details. The size of the confidence interval is defined by the option ALPHARMS= α , $0 \leq \alpha \leq 1$. The default is $\alpha = 0.1$, which corresponds to the 90% confidence interval for the RMSEA.

- **Probability for Test of Close Fit (Browne and Cudeck 1993)**

The traditional exact χ^2 test hypothesis $H_0: \epsilon_\alpha = 0$ is replaced by the null hypothesis of close fit $H_0: \epsilon_\alpha \leq 0.05$ and the exceedance probability P is computed as

$$P = 1 - \Phi(x|\lambda^*, df)$$

where $x = NM * F$ and $\lambda^* = 0.05^2 * NM * df$. The null hypothesis of close fit is rejected if P is smaller than a prespecified level (for example, $P < 0.05$).

- **Expected Cross Validation Index (Browne and Cudeck 1993)**

For GLS and WLS, the estimator c of the ECVI is linearly related to AIC:

$$c = F(\mathbf{S}, \mathbf{C}) + \frac{2t}{NM}$$

For ML estimation, c_{ML} is used.

$$c_{ML} = F_{ML}(\mathbf{S}, \mathbf{C}) + \frac{2t}{NM - n - 1}$$

The confidence interval $(c_L; c_U)$ for c is computed using the cumulative distribution function $\Phi(x|\lambda, df)$ of the noncentral chi-squared distribution,

$$(c_L; c_U) = \left(\frac{\lambda_L + nnt}{NM}; \frac{\lambda_U + nnt}{NM} \right)$$

with $nnt = n(n+1)/2 + t$, $x = NM * F$, $\Phi(x|\lambda_U, df) = 1 - \frac{\alpha}{2}$, and $\Phi(x|\lambda_L, df) = \frac{\alpha}{2}$. The confidence interval $(c_L^*; c_U^*)$ for c_{ML} is

$$(c_L^*; c_U^*) = \left(\frac{\lambda_L^* + nnt}{NM - n - 1}; \frac{\lambda_U^* + nnt}{NM - n - 1} \right)$$

where $nnt = n(n+1)/2 + t$, $x = (NM - n - 1) * F$, $\Phi(x|\lambda_U^*, df) = 1 - \frac{\alpha}{2}$ and $\Phi(x|\lambda_L^*, df) = \frac{\alpha}{2}$. Refer to Browne and Cudeck (1993). The size of the confidence interval is defined by the option ALPHAECV= α , $0 \leq \alpha \leq 1$. The default is $\alpha = 0.1$, which corresponds to the 90% confidence interval for the ECVI.

- **Comparative Fit Index (Bentler 1989)**

$$CFI = 1 - \frac{\max(NM * f_{min} - df_{min}, 0)}{\max(NM * f_0 - df_0, 0)}$$

- **Adjusted χ^2 Value (Browne 1982)**

If the variables are n -variate elliptic rather than normal and have significant amounts of multivariate kurtosis (leptokurtic or platykurtic), the χ^2 value can be adjusted to

$$\chi_{ell}^2 = \frac{\chi^2}{\eta_2}$$

where η_2 is the multivariate relative kurtosis coefficient.

- **Normal Theory Reweighted LS χ^2 Value**

This index is displayed only if METHOD=ML. Instead of the function value F_{ML} , the reweighted goodness-of-fit function F_{GMLS} is used,

$$\chi_{GMLS}^2 = NM * F_{GMLS}$$

where F_{GMLS} is the value of the function at the minimum.

- **Akaike's Information Criterion (AIC) (Akaike 1974; Akaike 1987)**

This is a criterion for selecting the best model among a number of candidate models. The model that yields the smallest value of AIC is considered the best.

$$AIC = \chi^2 - 2df$$

- **Consistent Akaike's Information Criterion (CAIC) (Bozdogan 1987)**

This is another criterion, similar to AIC, for selecting the best model among alternatives. The model that yields the smallest value of CAIC is considered the best. CAIC is preferred by some people to AIC or the χ^2 test.

$$CAIC = \chi^2 - (\ln(N) + 1)df$$

- **Schwarz's Bayesian Criterion (SBC) (Schwarz 1978; Sclove 1987)**

This is another criterion, similar to AIC, for selecting the best model. The model that yields the smallest value of SBC is considered the best. SBC is preferred by some people to AIC or the χ^2 test.

$$SBC = \chi^2 - \ln(N)df$$

- **McDonald's Measure of Centrality (McDonald and Hartmann 1992)**

$$CENT = \exp\left(-\frac{(\chi^2 - df)}{2N}\right)$$

- **Parsimonious Normed Fit Index (James, Mulaik, and Brett 1982)**

The PNFI is a modification of Bentler-Bonett's normed fit index that takes parsimony of the model into account,

$$PNFI = \frac{df_{min}}{df_0} \frac{(f_0 - f_{min})}{f_0}$$

The PNFI uses the same parsimonious factor as the parsimonious GFI of Mulaik et al. (1989).

- **Z-Test (Wilson and Hilferty 1931)**

The Z-Test of Wilson and Hilferty assumes an n -variate normal distribution:

$$Z = \frac{\sqrt[3]{\frac{\chi^2}{df}} - (1 - \frac{2}{9df})}{\sqrt{\frac{2}{9df}}}$$

Refer to McArdle (1988) and Bishop, Fienberg, and Holland (1977, p. 527) for an application of the Z-Test.

- **Nonnormed Coefficient (Bentler and Bonett 1980)**

$$\rho = \frac{f_0/df_0 - f_{min}/df_{min}}{f_0/df_0 - 1/NM}$$

Refer to Tucker and Lewis (1973).

- **Normed Coefficient (Bentler and Bonett 1980)**

$$\Delta = \frac{f_0 - f_{min}}{f_0}$$

Mulaik et al. (1989) recommend the parsimonious weighted form PNFI.

- **Normed Index ρ_1 (Bollen 1986)**

$$\rho_1 = \frac{f_0/df_0 - f_{min}/df_{min}}{f_0/df_0}$$

ρ_1 is always less than or equal to 1; $\rho_1 < 0$ is unlikely in practice. Refer to the discussion in Bollen (1989a).

- **Nonnormed Index Δ_2 (Bollen 1989a)**

$$\Delta_2 = \frac{f_0 - f_{min}}{f_0 - \frac{df}{NM}}$$

is a modification of Bentler & Bonett's Δ that uses df and “lessens the dependence” on N . Refer to the discussion in Bollen (1989b). Δ_2 is identical to Mulaik et al.'s (1989) IFI2 index.

- **Critical N Index (Hoelter 1983)**

$$CN = \frac{\chi_{crit}^2}{F} + 1$$

where χ_{crit}^2 is the critical chi-square value for the given df degrees of freedom and probability $\alpha = 0.05$, and F is the value of the estimation criterion (minimization function). Refer to Bollen (1989b, p. 277). Hoelter (1983) suggests that CN should be at least 200; however, Bollen (1989b) notes that the CN value may lead to an overly pessimistic assessment of fit for small samples.

Squared Multiple Correlation

The following are measures of the squared multiple correlation for manifest and endogenous variables and are computed for all five estimation methods: ULS, GLS, ML, WLS, and DWLS. These coefficients are computed as in the LISREL VI program of Jöreskog and Sörbom (1985). The DETAE, DETSE, and DETMV determination coefficients are intended to be global means of the squared multiple correlations for different subsets of model equations and variables. These coefficients are

displayed only when you specify the PDETERM option with a RAM or LINEQS model.

- **R^2 Values Corresponding to Endogenous Variables**

$$R_i^2 = 1 - \frac{\widehat{\text{var}}(\zeta_i)}{\widehat{\text{var}}(\eta_i)}$$

- **Total Determination of All Equations**

$$DETAE = 1 - \frac{\det(\hat{\Theta}, \hat{\Psi})}{\det(\widehat{\text{Cov}}(y, x, \eta))}$$

- **Total Determination of the Structural Equations**

$$DETSE = 1 - \frac{\det(\hat{\Psi})}{\det(\widehat{\text{Cov}}(\eta))}$$

- **Total Determination of the Manifest Variables**

$$DETMV = 1 - \frac{\det(\hat{\Theta})}{\det(\mathbf{S})}$$

Caution: In the LISREL program, the structural equations are defined by specifying the BETA matrix. In PROC CALIS, a structural equation has a dependent left-hand-side variable that appears at least once on the right-hand side of another equation, or the equation has at least one right-hand-side variable that is the left-hand-side variable of another equation. Therefore, PROC CALIS sometimes identifies more equations as structural equations than the LISREL program does.

Measures of Multivariate Kurtosis

In many applications, the manifest variables are not even approximately multivariate normal. If this happens to be the case with your data set, the default generalized least-squares and maximum likelihood estimation methods are not appropriate, and you should compute the parameter estimates and their standard errors by an asymptotically distribution-free method, such as the WLS estimation method. If your manifest variables are multivariate normal, then they have a zero relative multivariate kurtosis, and all marginal distributions have zero kurtosis (Browne 1982). If your DATA= data set contains raw data, PROC CALIS computes univariate skewness and kurtosis and a set of multivariate kurtosis values. By default, the values of univariate skewness and kurtosis are corrected for bias (as in PROC UNIVARIATE), but using the BIASKUR option enables you to compute the uncorrected values also. The values are displayed when you specify the PROC CALIS statement option KURTOSIS.

- **Corrected Variance for Variable z_j**

$$\sigma_j^2 = \frac{1}{N-1} \sum_i^N (z_{ij} - \bar{z}_j)^2$$

- **Corrected Univariate Skewness for Variable z_j**

$$\gamma_{1(j)} = \frac{N}{(N-1)(N-2)} \frac{\sum_i^N (z_{ij} - \bar{z}_j)^3}{\sigma_j^3}$$

- **Uncorrected Univariate Skewness for Variable z_j**

$$\gamma_{1(j)} = \frac{N \sum_i^N (z_{ij} - \bar{z}_j)^3}{\sqrt{N [\sum_i^N (z_{ij} - \bar{z}_j)^2]^3}}$$

- **Corrected Univariate Kurtosis for Variable z_j**

$$\gamma_{2(j)} = \frac{N(N+1)}{(N-1)(N-2)(N-3)} \frac{\sum_i^N (z_{ij} - \bar{z}_j)^4}{\sigma_j^4} - \frac{3(N-1)^2}{(N-2)(N-3)}$$

- **Uncorrected Univariate Kurtosis for Variable z_j**

$$\gamma_{2(j)} = \frac{N \sum_i^N (z_{ij} - \bar{z}_j)^4}{[\sum_i^N (z_{ij} - \bar{z}_j)^2]^2} - 3$$

- **Mardia's Multivariate Kurtosis**

$$\gamma_2 = \frac{1}{N} \sum_i^N [(z_i - \bar{z})' S^{-1} (z_i - \bar{z})]^2 - n(n+2)$$

- **Relative Multivariate Kurtosis**

$$\eta_2 = \frac{\gamma_2 + n(n+2)}{n(n+2)}$$

- **Normalized Multivariate Kurtosis**

$$\kappa_0 = \frac{\gamma_2}{\sqrt{8n(n+2)/N}}$$

- **Mardia Based Kappa**

$$\kappa_1 = \frac{\gamma_2}{n(n+2)}$$

- **Mean Scaled Univariate Kurtosis**

$$\kappa_2 = \frac{1}{3n} \sum_j^n \gamma_{2(j)}$$

- **Adjusted Mean Scaled Univariate Kurtosis**

$$\kappa_3 = \frac{1}{3n} \sum_j^n \gamma_{2(j)}^*$$

with

$$\gamma_{2(j)}^* = \begin{cases} \gamma_{2(j)} & , \quad \text{if } \gamma_{2(j)} > \frac{-6}{n+2} \\ \frac{-6}{n+2} & , \quad \text{otherwise} \end{cases}$$

If variable Z_j is normally distributed, the uncorrected univariate kurtosis $\gamma_{2(j)}$ is equal to 0. If Z has an n -variate normal distribution, Mardia's multivariate kurtosis γ_2 is equal to 0. A variable Z_j is called *leptokurtic* if it has a positive value of $\gamma_{2(j)}$ and is called *platykurtic* if it has a negative value of $\gamma_{2(j)}$. The values of κ_1 , κ_2 , and κ_3 should not be smaller than a lower bound (Bentler 1985):

$$\hat{\kappa} \geq \frac{-2}{n+2}$$

PROC CALIS displays a message if this happens.

If weighted least-squares estimates (METHOD=WLS or METHOD=ADF) are specified and the weight matrix is computed from an input raw data set, the CALIS procedure computes two further measures of multivariate kurtosis.

- **Multivariate Mean Kappa**

$$\kappa_4 = \frac{1}{m} \sum_i^n \sum_j^i \sum_k^j \sum_l^k \hat{\kappa}_{ij,kl} - 1$$

where

$$\hat{\kappa}_{ij,kl} = \frac{s_{ij,kl}}{s_{ij}s_{kl} + s_{ik}s_{jl} + s_{il}s_{jk}}$$

and $m = n(n+1)(n+2)(n+3)/24$ is the number of elements in the vector $s_{ij,kl}$ (Bentler 1985).

- **Multivariate Least-Squares Kappa**

$$\kappa_5 = \frac{s'_4 s_2}{s'_2 s_2} - 1$$

where

$$s_{ij,kl} = \frac{1}{N} \sum_{r=1}^N (z_{ri} - \bar{z}_i)(z_{rj} - \bar{z}_j)(z_{rk} - \bar{z}_k)(z_{rl} - \bar{z}_l)$$

s_4 is the vector of the $s_{ij,kl}$, and s_2 is the vector of the elements in the denominator of $\hat{\kappa}$ (Bentler 1985).

The occurrence of significant nonzero values of Mardia's multivariate kurtosis γ_2 and significant amounts of some of the univariate kurtosis values $\gamma_{2(j)}$ indicate that your variables are not multivariate normal distributed. Violating the multivariate normality assumption in (default) generalized least-squares and maximum likelihood estimation usually leads to the wrong approximate standard errors and incorrect fit statistics based on the χ^2 value. In general, the parameter estimates are more stable against violation of the normal distribution assumption. For more details, refer to Browne (1974, 1982, 1984).

Initial Estimates

Each optimization technique requires a set of initial values for the parameters. To avoid local optima, the initial values should be as close as possible to the globally optimal solution. You can check for local optima by running the analysis with several different sets of initial values; the RANDOM= option in the PROC CALIS statement is useful in this regard.

- **RAM and LINEQS:** There are several default estimation methods available in PROC CALIS for initial values of parameters in a linear structural equation model specified by a RAM or LINEQS model statement, depending on the form of the specified model.
 - two-stage least-squares estimation
 - instrumental variable method (Hägglund 1982; Jennrich 1987)
 - approximative factor analysis method
 - ordinary least-squares estimation
 - estimation method of McDonald (McDonald and Hartmann 1992)
- **FACTOR:** For default (exploratory) factor analysis, PROC CALIS computes initial estimates for factor loadings and unique variances by an algebraic method of approximate factor analysis. If you use a MATRIX statement together with a FACTOR model specification, initial values are computed by McDonald's (McDonald and Hartmann 1992) method if possible. McDonald's

method of computing initial values works better if you scale the factors by setting the factor variances to 1 rather than setting the loadings of the reference variables equal to 1. If none of the two methods seems to be appropriate, the initial values are set by the `START=` option.

- **COSAN:** For the more general COSAN model, there is no default estimation method for the initial values. In this case, the `START=` or `RANDOM=` option can be used to set otherwise unassigned initial values.

Poor initial values can cause convergence problems, especially with maximum likelihood estimation. You should not specify a constant initial value for all parameters since this would produce a singular predicted model matrix in the first iteration. Sufficiently large positive diagonal elements in the central matrices of each model matrix term provide a nonnegative definite initial predicted model matrix. If maximum likelihood estimation fails to converge, it may help to use `METHOD=LSML`, which uses the final estimates from an unweighted least-squares analysis as initial estimates for maximum likelihood. Or you can fit a slightly different but better-behaved model and produce an `OUTRAM=` data set, which can then be modified in accordance with the original model and used as an `INRAM=` data set to provide initial values for another analysis.

If you are analyzing a covariance or scalar product matrix, be sure to take into account the scales of the variables. The default initial values may be inappropriate when some variables have extremely large or small variances.

Automatic Variable Selection

You can use the `VAR` statement to reorder the variables in the model and to delete the variables not used. Using the `VAR` statement saves memory and computation time. If a linear structural equation model using the `RAM` or `LINEQS` statement (or an `INRAM=` data set specifying a `RAM` or `LINEQS` model) does not use all the manifest variables given in the input `DATA=` data set, `PROC CALIS` automatically deletes those manifest variables not used in the model.

In some special circumstances, the automatic variable selection performed for the `RAM` and `LINEQS` statements may be inappropriate, for example, if you are interested in modification indices connected to some of the variables that are not used in the model. You can include such manifest variables as exogenous variables in the analysis by specifying constant zero coefficients.

For example, the first three steps in a stepwise regression analysis of the Werner Blood Chemistry data (Jöreskog and Sörbom 1988, p. 111) can be performed as follows:

```
proc calis data=dixon method=glS nobs=180 print mod;
  lineqs y=0 x1+0 x2+0 x3+0 x4+0 x5+0 x6+0 x7+e;
  std    e=var;
run;
proc calis data=dixon method=glS nobs=180 print mod;
  lineqs y=g1 x1+0 x2+0 x3+0 x4+0 x5+0 x6+0 x7+e;
  std    e=var;
```

```

run;
proc calis data=dixon method=glis nobs=180 print mod;
  lineqs y=g1 x1+0 x2+0 x3+0 x4+0 x5+g6 x6+0 x7+e;
  std    e=var;
run;

```

Using the COSAN statement does not automatically delete those variables from the analysis that are not used in the model. You can use the output of the predetermined values in the predicted model matrix (PREDET option) to detect unused variables. Variables that are not used in the model are indicated by 0 in the rows and columns of the predetermined predicted model matrix.

Exogenous Manifest Variables

If there are exogenous manifest variables in the linear structural equation model, then there is a one-to-one relationship between the given covariances and corresponding estimates in the central model matrix (\mathbf{P} or Φ). In general, using exogenous manifest variables reduces the degrees of freedom since the corresponding sample correlations or covariances are not part of the exogenous information provided for the parameter estimation. See the section “Counting the Degrees of Freedom” on page 563 for more information.

If you specify a RAM or LINEQS model statement, or if such a model is recognized in an INRAM= data set, those elements in the central model matrices that correspond to the exogenous manifest variables are reset to the sample values after computing covariances or correlations within the current BY group.

The COSAN statement does not automatically set the covariances in the central model matrices that correspond to manifest exogenous variables.

You can use the output of the predetermined values in the predicted model matrix (PREDET option) that correspond to manifest exogenous variables to see which of the manifest variables are exogenous variables and to help you set the corresponding locations of the central model matrices with their covariances.

The following two examples show how different the results of PROC CALIS can be if manifest variables are considered either as endogenous or as exogenous variables. (See Figure 19.5.) In both examples, a correlation matrix \mathbf{S} is tested against an identity model matrix \mathbf{C} ; that is, no parameter is estimated. The three runs of the first example (specified by the COSAN, LINEQS, and RAM statements) consider the two variables y and x as endogenous variables.

```

title2 'Data: FULLER (1987, p.18)';
data corn;
  input y x;
  datalines;
86  70
115 97
90  53
86  64
110 95

```

```

91  64
99  50
96  70
99  94
104 69
96  51
;

title3 'Endogenous Y and X';
proc calis data=corn;
    cosal corr(2,ide);
run;
proc calis data=corn;
    lineqs
        y=ey,
        x=ex;
    std    ey ex=2 * 1;
run;
proc calis data=corn;
    ram
        1  1  3  1.,
        1  2  4  1.,
        2  3  3  1.,
        2  4  4  1.;
run;

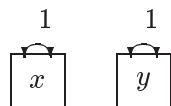
```

The two runs of the second example (specified by the LINEQS and RAM statements) consider y and x as exogenous variables.

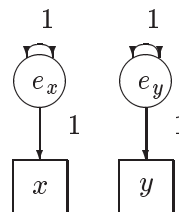
```

title3 'Exogenous Y and X';
proc calis data=corn;
    std y x=2 * 1;
run;
proc calis data=corn;
    ram
        2  1  1  1.,
        2  2  2  1.;
run;

```



Exogenous x, y



Endogenous x, y

Figure 19.5. Exogenous and Endogenous Variables

The LINEQS and the RAM model statements set the covariances (correlations) of exogenous manifest variables in the estimated model matrix and automatically reduce the degrees of freedom.

Use of Optimization Techniques

No algorithm for optimizing general nonlinear functions exists that will always find the global optimum for a general nonlinear minimization problem in a reasonable amount of time. Since no single optimization technique is invariably superior to others, PROC CALIS provides a variety of optimization techniques that work well in various circumstances. However, you can devise problems for which none of the techniques in PROC CALIS will find the correct solution. All optimization techniques in PROC CALIS use $O(n^2)$ memory except the conjugate gradient methods, which use only $O(n)$ of memory and are designed to optimize problems with many parameters.

The PROC CALIS statement NLOPTIONS can be especially helpful for tuning applications with nonlinear equality and inequality constraints on the parameter estimates. Some of the options available in NLOPTIONS may also be invoked as PROC CALIS options. The NLOPTIONS statement can specify almost the same options as the SAS/OR NLP procedure.

Nonlinear optimization requires the repeated computation of

- the function value (optimization criterion)
- the gradient vector (first-order partial derivatives)
- for some techniques, the (approximate) Hessian matrix (second-order partial derivatives)
- values of linear and nonlinear constraints
- the first-order partial derivatives (Jacobian) of nonlinear constraints

For the criteria used by PROC CALIS, computing the gradient takes more computer time than computing the function value, and computing the Hessian takes *much* more computer time and memory than computing the gradient, especially when there are many parameters to estimate. Unfortunately, optimization techniques that do not use the Hessian usually require many more iterations than techniques that do use the (approximate) Hessian, and so they are often slower. Techniques that do not use the Hessian also tend to be less reliable (for example, they may terminate at local rather than global optima).

The available optimization techniques are displayed in Table 19.13 and can be chosen by the TECH=*name* option.

Table 19.13. Optimization Techniques

TECH=	Optimization Technique
LEVMAR	Levenberg-Marquardt Method
TRUREG	Trust-Region Method
NEWRAP	Newton-Raphson Method with Line Search
NRRIDG	Newton-Raphson Method with Ridging
QUANEW	Quasi-Newton Methods (DBFGS, DDFP, BFGS, DFP)
DBLDOG	Double-Dogleg Method (DBFGS, DDFP)
CONGRA	Conjugate Gradient Methods (PB, FR, PR, CD)

Table 19.14 shows, for each optimization technique, which derivatives are needed (first-order or second-order) and what kind of constraints (boundary, linear, or non-linear) can be imposed on the parameters.

Table 19.14. Derivatives Needed and Constraints Allowed

TECH=	Derivatives		Constraints		
	First Order	Second Order	Boundary	Linear	Nonlinear
LEVMAR	x	x	x	x	-
TRUREG	x	x	x	x	-
NEWRAP	x	x	x	x	-
NRRIDG	x	x	x	x	-
QUANEW	x	-	x	x	x
DBLDOG	x	-	x	x	-
CONGRA	x	-	x	x	-

The Levenberg-Marquardt, trust-region, and Newton-Raphson techniques are usually the most reliable, work well with boundary and general linear constraints, and generally converge after a few iterations to a precise solution. However, these techniques need to compute a Hessian matrix in each iteration. For HESSALG=1, this means that you need about $4(n(n+1)/2)t$ bytes of work memory (n = the number of manifest variables, t = the number of parameters to estimate) to store the Jacobian and its cross product. With HESSALG=2 or HESSALG=3, you do not need this work memory, but the use of a utility file increases execution time. Computing the approximate Hessian in each iteration can be very time- and memory-consuming, especially for large problems (more than 60 or 100 parameters, depending on the computer used). For large problems, a quasi-Newton technique, especially with the BFGS update, can be far more efficient.

For a poor choice of initial values, the Levenberg-Marquardt method seems to be more reliable.

If memory problems occur, you can use one of the conjugate gradient techniques, but they are generally slower and less reliable than the methods that use second-order information.

There are several options to control the optimization process. First of all, you can specify various termination criteria. You can specify the GCONV= option to specify a relative gradient termination criterion. If there are active boundary constraints, only those gradient components that correspond to inactive constraints contribute to the criterion. When you want very precise parameter estimates, the GCONV= option is

useful. Other criteria that use relative changes in function values or parameter estimates in consecutive iterations can lead to early termination when active constraints cause small steps to occur. The small default value for the FCONV= option helps prevent early termination. Using the MAXITER= and MAXFUNC= options enables you to specify the maximum number of iterations and function calls in the optimization process. These limits are especially useful in combination with the INRAM= and OUTRAM= options; you can run a few iterations at a time, inspect the results, and decide whether to continue iterating.

Nonlinearly Constrained QN Optimization

The algorithm used for nonlinearly constrained quasi-Newton optimization is an efficient modification of Powell's (1978a, 1978b, 1982a, 1982b) *Variable Metric Constrained WatchDog* (VMCWD) algorithm. A similar but older algorithm (VF02AD) is part of the Harwell library. Both VMCWD and VF02AD use Fletcher's VE02AD algorithm (also part of the Harwell library) for positive definite quadratic programming. The PROC CALIS QUANEW implementation uses a quadratic programming subroutine that updates and downdates the approximation of the Cholesky factor when the active set changes. The nonlinear QUANEW algorithm is not a feasible point algorithm, and the value of the objective function need not decrease (minimization) or increase (maximization) monotonically. Instead, the algorithm tries to reduce a linear combination of the objective function and constraint violations, called the *merit function*.

The following are similarities and differences between this algorithm and VMCWD:

- A modification of this algorithm can be performed by specifying VERSION=1, which replaces the update of the Lagrange vector μ with the original update of Powell (1978a, 1978b), which is used in VF02AD. This can be helpful for some applications with linearly dependent active constraints.
- If the VERSION= option is not specified or VERSION=2 is specified, the evaluation of the Lagrange vector μ is performed in the same way as Powell (1982a, 1982b) describes.
- Instead of updating an approximate Hessian matrix, this algorithm uses the dual BFGS (or DFP) update that updates the Cholesky factor of an approximate Hessian. If the condition of the updated matrix gets too bad, a restart is done with a positive diagonal matrix. At the end of the first iteration after each restart, the Cholesky factor is scaled.
- The Cholesky factor is loaded into the quadratic programming subroutine, automatically ensuring positive definiteness of the problem. During the quadratic programming step, the Cholesky factor of the projected Hessian matrix $\mathbf{Z}_k' \mathbf{G} \mathbf{Z}_k$ and the $\mathbf{Q}\mathbf{T}$ decomposition are updated simultaneously when the active set changes. Refer to Gill et al. (1984) for more information.
- The line-search strategy is very similar to that of Powell (1982a, 1982b). However, this algorithm does not call for derivatives during the line search; hence, it generally needs fewer derivative calls than function calls. The VMCWD algorithm always requires the same number of derivative and function calls. It was also found in several applications of VMCWD that Powell's line-search method sometimes uses steps that are too long during the first iterations. In

those cases, you can use the `INSTEP=` option specification to restrict the step length α of the first iterations.

- Also the watchdog strategy is similar to that of Powell (1982a, 1982b). However, this algorithm doesn't return automatically after a fixed number of iterations to a former better point. A return here is further delayed if the observed function reduction is close to the expected function reduction of the quadratic model.
- Although Powell's termination criterion still is used (as `FCONV2`), the `QUANEW` implementation uses two additional termination criteria (`GCONV` and `ABSGCONV`).

This algorithm is automatically invoked when you specify the `NLINCON` statement. The nonlinear `QUANEW` algorithm needs the Jacobian matrix of the first-order derivatives (constraints normals) of the constraints

$$(\nabla c_i) = \left(\frac{\partial c_i}{\partial x_j} \right), \quad i = 1, \dots, nc, j = 1, \dots, n$$

where nc is the number of nonlinear constraints for a given point x .

You can specify two update formulas with the `UPDATE=` option:

- `UPDATE=DBFGS` performs the dual BFGS update of the Cholesky factor of the Hessian matrix. This is the default.
- `UPDATE=DDFP` performs the dual DFP update of the Cholesky factor of the Hessian matrix.

This algorithm uses its own line-search technique. All options and parameters (except the `INSTEP=` option) controlling the line search in the other algorithms do not apply here. In several applications, large steps in the first iterations are troublesome. You can specify the `INSTEP=` option to impose an upper bound for the step size α during the first five iterations. The values of the `LCSINGULAR=`, `LCEPSILON=`, and `LCDEACT=` options, which control the processing of linear and boundary constraints, are valid only for the quadratic programming subroutine used in each iteration of the nonlinear constraints `QUANEW` algorithm.

Optimization and Iteration History

The optimization and iteration histories are displayed by default because it is important to check for possible convergence problems.

The optimization history includes the following summary of information about the initial state of the optimization.

- the number of constraints that are active at the starting point, or more precisely, the number of constraints that are currently members of the working set. If this number is followed by a plus sign, there are more active constraints, of which at least one is temporarily released from the working set due to negative Lagrange multipliers.
- the value of the objective function at the starting point

- if the (projected) gradient is available, the value of the largest absolute (projected) gradient element
- for the TRUREG and LEVMAR subroutines, the initial radius of the trust region around the starting point

The optimization history ends with some information concerning the optimization result:

- the number of constraints that are active at the final point, or more precisely, the number of constraints that are currently members of the working set. If this number is followed by a plus sign, there are more active constraints, of which at least one is temporarily released from the working set due to negative Lagrange multipliers.
- the value of the objective function at the final point
- if the (projected) gradient is available, the value of the largest absolute (projected) gradient element
- other information specific to the optimization technique

The iteration history generally consists of one line of displayed output containing the most important information for each iteration. The `_LIST_` variable (see the “SAS Program Statements” section on page 514) also enables you to display the parameter estimates and the gradient in some or all iterations.

The iteration history always includes the following (the words in parentheses are the column header output):

- the iteration number (Iter)
- the number of iteration restarts (rest)
- the number of function calls (nfun)
- the number of active constraints (act)
- the value of the optimization criterion (optcrit)
- the difference between adjacent function values (difcrit)
- the maximum of the absolute gradient components corresponding to inactive boundary constraints (maxgrad)

An apostrophe trailing the number of active constraints indicates that at least one of the active constraints is released from the active set due to a significant Lagrange multiplier.

For the Levenberg-Marquardt technique (LEVMAR), the iteration history also includes the following information:

- An asterisk trailing the iteration number means that the computed Hessian approximation is singular and consequently ridged with a positive lambda value. If all or the last several iterations show a singular Hessian approximation, the

problem is not sufficiently identified. Thus, there are other locally optimal solutions that lead to the same optimum function value for different parameter values. This implies that standard errors for the parameter estimates are not computable without the addition of further constraints.

- the value of the Lagrange multiplier (λ); this is 0 if the optimum of the quadratic function approximation is inside the trust region (a trust-region-scaled Newton step can be performed) and is greater than 0 when the optimum of the quadratic function approximation is located at the boundary of the trust region (the scaled Newton step is too long to fit in the trust region and a quadratic constraint optimization is performed). Large values indicate optimization difficulties. For a nonsingular Hessian matrix, the value of λ should go to 0 during the last iterations, indicating that the objective function can be well approximated by a quadratic function in a small neighborhood of the optimum point. An increasing λ value often indicates problems in the optimization process.
- the value of the ratio ρ (rho) between the actually achieved difference in function values and the predicted difference in the function values on the basis of the quadratic function approximation. Values much less than 1 indicate optimization difficulties. The value of the ratio ρ indicates the goodness of the quadratic function approximation; in other words, $\rho \ll 1$ means that the radius of the trust region has to be reduced. A fairly large value of ρ means that the radius of the trust region need not be changed. And a value close to or larger than 1 means that the radius can be increased, indicating a good quadratic function approximation.

For the Newton-Raphson technique (NRRIDG), the iteration history also includes the following information:

- the value of the ridge parameter. This is 0 when a Newton step can be performed, and it is greater than 0 when either the Hessian approximation is singular or a Newton step fails to reduce the optimization criterion. Large values indicate optimization difficulties.
- the value of the ratio ρ (rho) between the actually achieved difference in function values and the predicted difference in the function values on the basis of the quadratic function approximation. Values much less than 1.0 indicate optimization difficulties.

For the Newton-Raphson with line-search technique (NEWRAP), the iteration history also includes

- the step size α (alpha) computed with one of the line-search algorithms
- the slope of the search direction at the current parameter iterate. For minimization, this value should be significantly negative. Otherwise, the line-search algorithm has difficulty reducing the function value sufficiently.

For the Trust-Region technique (TRUREG), the iteration history also includes the following information.

- An asterisk after the iteration number means that the computed Hessian approximation is singular and consequently ridged with a positive lambda value.
- the value of the Lagrange multiplier (lambda). This value is zero when the optimum of the quadratic function approximation is inside the trust region (a trust-region-scaled Newton step can be performed) and is greater than zero when the optimum of the quadratic function approximation is located at the boundary of the trust region (the scaled Newton step is too long to fit in the trust region and a quadratically constrained optimization is performed). Large values indicate optimization difficulties. As in Gay (1983), a negative lambda value indicates the special case of an indefinite Hessian matrix (the smallest eigenvalue is negative in minimization).
- the value of the radius Δ of the trust region. Small trust region radius values combined with large lambda values in subsequent iterations indicate optimization problems.

For the quasi-Newton (QUANEW) and conjugate gradient (CONGRA) techniques, the iteration history also includes the following information:

- the step size (alpha) computed with one of the line-search algorithms
- the descent of the search direction at the current parameter iterate. This value should be significantly smaller than 0. Otherwise, the line-search algorithm has difficulty reducing the function value sufficiently.

Frequent update restarts (rest) of a quasi-Newton algorithm often indicate numerical problems related to required properties of the approximate Hessian update, and they decrease the speed of convergence. This can happen particularly if the ABSGCONV= termination criterion is too small, that is, when the requested precision cannot be obtained by quasi-Newton optimization. Generally, the number of automatic restarts used by conjugate gradient methods are much higher.

For the nonlinearly constrained quasi-Newton technique, the iteration history also includes the following information:

- the maximum value of all constraint violations,

$$\text{conmax} = \max(|c_i(x)| : c_i(x) < 0)$$

- the value of the predicted function reduction used with the GCONV and FCONV2 termination criteria,

$$\text{pred} = |g(x^{(k)})s(x^{(k)})| + \sum_{i=1}^m |\lambda_i c_i(x^{(k)})|$$

- the step size α of the quasi-Newton step. Note that this algorithm works with a special line-search algorithm.

- the maximum element of the gradient of the Lagrange function,

$$\begin{aligned}\text{Ifgmax} &= \nabla_x L(x^{(k)}, \lambda^{(k)}) \\ &= \nabla_x f(x^{(k)}) - \sum_{i=1}^m \lambda_i^{(k)} \nabla_x c_i(x^{(k)})\end{aligned}$$

For the double dogleg technique, the iteration history also includes the following information:

- the parameter λ of the double-dogleg step. A value $\lambda = 0$ corresponds to the full (quasi) Newton step.
- the slope of the search direction at the current parameter iterate. For minimization, this value should be significantly negative.

Line-Search Methods

In each iteration k , the (dual) quasi-Newton, hybrid quasi-Newton, conjugate gradient, and Newton-Raphson minimization techniques use iterative line-search algorithms that try to optimize a linear, quadratic, or cubic approximation of the nonlinear objective function f of n parameters x along a feasible descent search direction $s^{(k)}$

$$f(x^{(k+1)}) = f(x^{(k)} + \alpha^{(k)} s^{(k)})$$

by computing an approximately optimal scalar $\alpha^{(k)} > 0$. Since the outside iteration process is based only on the approximation of the objective function, the inside iteration of the line-search algorithm does not have to be perfect. Usually, it is satisfactory that the choice of α significantly reduces (in a minimization) the objective function. Criteria often used for termination of line-search algorithms are the Goldstein conditions (Fletcher 1987).

Various line-search algorithms can be selected by using the LIS= option (page 468). The line-search methods LIS=1, LIS=2, and LIS=3 satisfy the left-hand-side and right-hand-side Goldstein conditions (refer to Fletcher 1987). When derivatives are available, the line-search methods LIS=6, LIS=7, and LIS=8 try to satisfy the right-hand-side Goldstein condition; if derivatives are not available, these line-search algorithms use only function calls.

The line-search method LIS=2 seems to be superior when function evaluation consumes significantly less computation time than gradient evaluation. Therefore, LIS=2 is the default value for Newton-Raphson, (dual) quasi-Newton, and conjugate gradient optimizations.

Restricting the Step Length

Almost all line-search algorithms use iterative extrapolation techniques that can easily lead to feasible points where the objective function f is no longer defined (resulting in indefinite matrices for ML estimation) or is difficult to compute (resulting in floating point overflows). Therefore, PROC CALIS provides options that restrict the step length or trust region radius, especially during the first main iterations.

The inner product $g's$ of the gradient g and the search direction s is the slope of $f(\alpha) = f(x + \alpha s)$ along the search direction s with step length α . The default

starting value $\alpha^{(0)} = \alpha^{(k,0)}$ in each line-search algorithm ($\min_{\alpha>0} f(x + \alpha s)$) during the main iteration k is computed in three steps.

1. Use either the difference $df = |f^{(k)} - f^{(k-1)}|$ of the function values during the last two consecutive iterations or the final stepsize value α^- of the previous iteration $k - 1$ to compute a first value $\alpha_1^{(0)}$.

- Using the DAMPSTEP=<=r> option:

$$\alpha_1^{(0)} = \min(1, r\alpha^-)$$

The initial value for the new step length can be no larger than r times the final step length α^- of the previous iteration. The default is $r = 2$.

- Not using the DAMPSTEP option:

$$\alpha_1^{(0)} = \begin{cases} step & \text{if } 0.1 \leq step \leq 10 \\ 10 & \text{if } step > 10 \\ 0.1 & \text{if } step < 0.1 \end{cases}$$

with

$$step = \begin{cases} df/|g's| & \text{if } |g's| \geq \epsilon \max(100df, 1) \\ 1 & \text{otherwise} \end{cases}$$

This value of $\alpha_1^{(0)}$ can be too large and can lead to a difficult or impossible function evaluation, especially for highly nonlinear functions such as the EXP function.

2. During the first five iterations, the second step enables you to reduce $\alpha_1^{(0)}$ to a smaller starting value $\alpha_2^{(0)}$ using the INSTEP= r option:

$$\alpha_2^{(0)} = \min(\alpha_1^{(0)}, r)$$

After more than five iterations, $\alpha_2^{(0)}$ is set to $\alpha_1^{(0)}$.

3. The third step can further reduce the step length by

$$\alpha_3^{(0)} = \min(\alpha_2^{(0)}, \min(10, u))$$

where u is the maximum length of a step inside the feasible region.

The INSTEP= r option lets you specify a smaller or larger radius of the trust region used in the first iteration by the trust-region, double-dogleg, and Levenberg-Marquardt algorithm. The default initial trust region radius is the length of the scaled gradient (Moré 1978). This step corresponds to the default radius factor of $r = 1$. This choice is successful in most practical applications of the TRUREG, DBLDOG, and LEVMAR algorithms. However, for bad initial values used in the analysis of a covariance matrix with high variances, or for highly nonlinear constraints (such as using the EXP function) in your programming code, the default start radius can result in arithmetic overflows. If this happens, you can try decreasing values of INSTEP= r , $0 < r < 1$, until the iteration starts successfully. A small factor r also affects the trust region radius of the next steps because the radius is changed in each iteration by a factor $0 < c \leq 4$ depending on the ρ ratio. Reducing the radius corresponds to increasing the ridge parameter λ that produces smaller steps directed closer toward the gradient direction.

Modification Indices

While fitting structural models, you may want to modify the specified model in order to

- reduce the χ^2 value significantly
- reduce the number of parameters to estimate without increasing the χ^2 value too much

If you specify the MODIFICATION or MOD option, PROC CALIS computes and displays a default set of modification indices:

- **Univariate Lagrange multiplier test indices** for most elements in the model matrices that are constrained to *equal constants*. These are second-order approximations of the decrease in the χ^2 value that would result from allowing the constant matrix element to vary. Besides the value of the Lagrange multiplier, the corresponding probability ($df = 1$) and the approximate change of the parameter value (should the constant be changed to a parameter) are displayed. If allowing the constant to be a free estimated parameter would result in a singular information matrix, the string 'sing' is displayed instead of the Lagrange multiplier index. Not all elements in the model matrices should be allowed to vary; the diagonal elements of the inverse matrices in the RAM or LINEQS model must be constant ones. The univariate Lagrange multipliers are displayed at the constant locations of the model matrices.
- **Univariate Wald test indices** for those matrix elements that correspond to *parameter estimates* in the model. These are second-order approximations of the increase in the χ^2 value that would result from constraining the parameter to a 0 constant. The univariate Wald test indices are the same as the t values that are displayed together with the parameter estimates and standard errors. The univariate Wald test indices are displayed at the parameter locations of the model matrices.
- **Univariate Lagrange multiplier test indices** that are second-order approximations of the decrease in the χ^2 value that would result from the release of *equality constraints*. Multiple equality constraints containing $n > 2$ parameters are tested successively in n steps, each assuming the release of one of the equality-constrained parameters. The expected change of the parameter values of the separated parameter and the remaining parameter cluster are displayed, too.
- **Univariate Lagrange multiplier test indices** for releasing *active boundary constraints* specified by the BOUNDS statement
- **Stepwise multivariate Wald test indices** for constraining estimated parameters to 0 are computed and displayed. In each step, the parameter that would lead to the smallest increase in the multivariate χ^2 value is set to 0. Besides the multivariate χ^2 value and its probability, the univariate increments are also displayed. The process stops when the univariate probability is smaller than the specified value in the SLMW= option.

All of the preceding tests are approximations. You can often get more accurate tests by actually fitting different models and computing likelihood ratio tests. For more details about the Wald and the Lagrange multiplier test, refer to MacCallum (1986), Buse (1982), Bentler (1986), or Lee (1985).

Note that, for large model matrices, the computation time for the default modification indices can considerably exceed the time needed for the minimization process.

The modification indices are not computed for unweighted least-squares or diagonally weighted least-squares estimation.

Caution: Modification indices are not computed if the model matrix is an identity matrix (IDE or ZID), a selection matrix (PER), or the first matrix **J** in the LINEQS model. If you want to display the modification indices for such a matrix, you should specify the matrix as another type; for example, specify an identity matrix used in the COSAN statement as a diagonal matrix with constant diagonal elements of 1.

Constrained Estimation Using Program Code

The CALIS procedure offers a very flexible way to constrain parameter estimates. You can use your own programming statements to express special properties of the parameter estimates. This tool is also present in McDonald's COSAN implementation but is considerably easier to use in the CALIS procedure. PROC CALIS is able to compute analytic first- and second-order derivatives that you would have to specify using the COSAN program. There are also three PROC CALIS statements you can use:

- the BOUNDS statement, to specify simple bounds on the parameters used in the optimization process
- the LINCON statement, to specify general linear equality and inequality constraints on the parameters used in the optimization process
- the NLINCON statement, to specify general nonlinear equality and inequality constraints on the parameters used in the optimization process. The variables listed in the NLINCON statement must be specified in the program code.

There are some traditional ways to enforce parameter constraints by using parameter transformations (McDonald 1980).

- **One-sided boundary constraints:** For example, the parameter q_k should be at least as large (or at most as small) as a given constant value a (or b),

$$q_k \geq a \quad \text{or} \quad q_k \leq b$$

This inequality constraint can be expressed as an equality constraint

$$q_k = a + x_j^2 \quad \text{or} \quad q_k = b - x_j^2$$

in which the fundamental parameter x_j is unconstrained.

- **Two-sided boundary constraints:** For example, the parameter q_k should be located between two given constant values a and b , $a < b$,

$$a \leq q_k \leq b$$

This inequality constraint can be expressed as an equality constraint

$$q_k = a + b \frac{\exp(x_j)}{1 + \exp(x_j)}$$

in which the fundamental parameter x_j is unconstrained.

- **One-sided order constraints:** For example, the parameters q_1, \dots, q_k should be ordered in the form

$$q_1 \leq q_2, \quad q_1 \leq q_3, \quad \dots, \quad q_1 \leq q_k$$

These inequality constraints can be expressed as a set of equality constraints

$$q_1 = x_1, \quad q_2 = x_1 + x_2^2, \quad \dots, \quad q_k = x_1 + x_k^2$$

in which the fundamental parameters x_1, \dots, x_k are unconstrained.

- **Two-sided order constraints:** For example, the parameters q_1, \dots, q_k should be ordered in the form

$$q_1 \leq q_2 \leq q_3 \leq \dots \leq q_k$$

These inequality constraints can be expressed as a set of equality constraints

$$q_1 = x_1, \quad q_2 = q_1 + x_2^2, \quad \dots, \quad q_k = q_{k-1} + x_k^2$$

in which the fundamental parameters x_1, \dots, x_k are unconstrained.

- **Linear equation constraints:** For example, the parameters q_1, q_2, q_3 should be linearly constrained in the form

$$q_1 + q_2 + q_3 = a$$

which can be expressed in the form of three explicit equations in which the fundamental parameters x_1 and x_2 are unconstrained:

$$q_1 = x_1, \quad q_2 = x_2, \quad q_3 = a - x_1 - x_2$$

Refer to McDonald (1980) and Browne (1982) for further notes on reparameterizing techniques. If the optimization problem is not too large to apply the Levenberg-Marquardt or Newton-Raphson algorithm, boundary constraints should be requested by the BOUNDS statement rather than by reparameterizing code. If the problem is so large that you must use a quasi-Newton or conjugate gradient algorithm, reparameterizing techniques may be more efficient than the BOUNDS statement.

Counting the Degrees of Freedom

In a regression problem, the number of degrees of freedom for the error estimate is the number of observations in the data set minus the number of parameters. The NOBS=, DFR= (RDF=), and DFE= (EDF=) options refer to degrees of freedom in this sense. However, these values are not related to the degrees of freedom of a test statistic used in a covariance or correlation structure analysis. The NOBS=, DFR=, and DFE= options should be used in PROC CALIS to specify only the effective number of observations in the input DATA= data set.

In general, the number of degrees of freedom in a covariance or correlation structure analysis is defined as the difference between the number of nonredundant values q in the observed $n \times n$ correlation or covariance matrix \mathbf{S} and the number t of free parameters \mathbf{X} used in the fit of the specified model, $df = q - t$. Both values, q and t , are counted differently in different situations by PROC CALIS.

The number of nonredundant values q is generally equal to the number of lower triangular elements in the $n \times n$ moment matrix \mathbf{S} including all diagonal elements, minus a constant c dependent upon special circumstances,

$$q = n(n + 1)/2 - c$$

The number c is evaluated by adding the following quantities:

- If you specify a linear structural equation model containing exogenous manifest variables by using the RAM or LINEQS statement, PROC CALIS adds to c the number of variances and covariances among these manifest exogenous variables, which are automatically set in the corresponding locations of the central model matrices (see the section “Exogenous Manifest Variables” on page 549).
- If you specify the DFREDUCE= i option, PROC CALIS adds the specified number i to c . The number i can be a negative integer.
- If you specify the NODIAG option to exclude the fit of the diagonal elements of the data matrix \mathbf{S} , PROC CALIS adds the number n of diagonal elements to c .
- If all the following conditions hold, then PROC CALIS adds to c the number of the diagonal locations:
 - NODIAG and DFREDUC= options are not specified.
 - A correlation structure is being fitted.
 - The predicted correlation matrix contains constants on the diagonal.

In some complicated models, especially those using programming statements, PROC CALIS may not be able to detect all the constant predicted values. In such cases, you must specify the `DFREDUCE=` option to get the correct degrees of freedom.

The number t is the number of different parameter names used in constructing the model if you do not use programming statements to impose constraints on the parameters. Using programming statements in general introduces two kinds of parameters:

- independent parameters, which are used only at the right-hand side of the expressions
- dependent parameters, which are used at least once at the left-hand side of the expressions

The independent parameters belong to the parameters involved in the estimation process, whereas the dependent parameters are fully defined by the programming statements and can be computed from the independent parameters. In this case, the number t is the number of different parameter names used in the model specification, but not used in the programming statements, plus the number of independent parameters. The independent parameters and their initial values can be defined in a model specification statement or in a `PARMS` statement.

The degrees of freedom are automatically increased by the number of active constraints in the solution. Similarly, the number of parameters are decreased by the number of active constraints. This affects the computation of many fit statistics and indices. Refer to Dijkstra (1992) for a discussion of the validity of statistical inferences with active boundary constraints. If the researcher believes that the active constraints will have a small chance of occurrence in repeated sampling, it may be more suitable to turn off the automatic adjustment using the `NOADJDF` option.

Computational Problems

First Iteration Overflows

Analyzing a covariance matrix including high variances in the diagonal and using bad initial estimates for the parameters can easily lead to arithmetic overflows in the first iterations of the minimization algorithm. The line-search algorithms that work with cubic extrapolation are especially sensitive to arithmetic overflows. If this occurs with quasi-Newton or conjugate gradient minimization, you can specify the `INSTEP=` option to reduce the length of the first step. If an arithmetic overflow occurs in the first iteration of the Levenberg-Marquardt algorithm, you can specify the `INSTEP=` option to reduce the trust region radius of the first iteration. You also can change the minimization technique or the line-search method. If none of these help, you should consider

- scaling the covariance matrix
- providing better initial values
- changing the model

No Convergence of Minimization Process

If convergence does not occur during the minimization process, perform the following tasks:

- If there are *negative variance estimates* in the diagonal locations of the central model matrices, you can
 - specify the BOUNDS statement to obtain nonnegative variance estimates
 - specify the HEYWOOD option, if the FACTOR model statement is specified
- Change the estimation method to obtain a better set of initial estimates. For example, if you use METHOD=ML, you can
 - change to METHOD=LSML
 - run some iterations with METHOD=DWLS or METHOD=GLS, write the results in an OUTRAM= data set, and use the results as initial values specified by an INRAM= data set in a second run with METHOD=ML
- Change the optimization technique. For example, if you use the default TECH=LEVMAR, you can
 - change to TECH=QUANEW or to TECH=NEWRAP
 - run some iterations with TECH=CONGRA, write the results in an OUTRAM= data set, and use the results as initial values specified by an INRAM= data set in a second run with a different TECH= technique
- Change or modify the update technique or the line-search algorithm, or both, when using TECH=QUANEW or TECH=CONGRA. For example, if you use the default update formula and the default line-search algorithm, you can
 - change the update formula with the UPDATE= option
 - change the line-search algorithm with the LIS= option
 - specify a more precise line search with the LSPRECISION= option, if you use LIS=2 or LIS=3
- You can allow more iterations and function calls by using the MAXIT= and MAXFU= options.
- Change the initial values. For many categories of model specifications done by the LINEQS, RAM, or FACTOR model, PROC CALIS computes an appropriate set of initial values automatically. However, for some of the model specifications (for example, structural equations with latent variables on the left-hand side and manifest variables on the right-hand side), PROC CALIS can generate very obscure initial values. In these cases, you have to set the initial values yourself.
 - Increase the initial values of the parameters located at the diagonal of central matrices
 - * manually, by setting the values in the model specification
 - * automatically, by using the DEMPHAS= option

- Use a slightly different, but more stable, model to obtain preliminary estimates.
- Use additional information to specify initial values, for example, by using other SAS software like the FACTOR, REG, SYSLIN, and MODEL (SYSNLIN) procedures for the modified, unrestricted model case.
- Change the optimization technique. For example, if you use the default TECH=LEVMAR, you can
 - change to TECH=QUANEW or to TECH=NEWRAP
 - run some iterations with TECH=CONGRA, write the results in an OUTRAM= data set, and use the results as initial values specified by an INRAM= data set in a second run with a different TECH= technique
- Change or modify the update technique or the line-search algorithm, or both, when using TECH=QUANEW or TECH=CONGRA. For example, if you use the default update formula and the default line-search algorithm, you can
 - change the update formula with the UPDATE= option
 - change the line-search algorithm with the LIS= option
 - specify a more precise line search with the LSPRECISION= option, if you use LIS=2 or LIS=3
- Temporarily change the estimation method to obtain a better set of initial estimates. For example, if you use METHOD=ML, you can
 - change to METHOD=LSML
 - run some iterations with METHOD=DWLS or GLS, write the results in an OUTRAM= data set, and use the results as initial values specified by an INRAM= data set in a second run with METHOD=ML
- You can allow more iterations and function calls by using the MAXIT= and MAXFU= options.

Unidentified Model

The parameter vector \mathbf{x} in the covariance structure model

$$\mathbf{C} = \mathbf{C}(\mathbf{x})$$

is said to be identified in a parameter space G , if

$$\mathbf{C}(\mathbf{x}) = \mathbf{C}(\tilde{\mathbf{x}}), \quad \tilde{\mathbf{x}} \in G$$

implies $\mathbf{x} = \tilde{\mathbf{x}}$. The parameter estimates that result from an unidentified model can be very far from the parameter estimates of a very similar but identified model. They are usually machine dependent. Don't use parameter estimates of an unidentified model as initial values for another run of PROC CALIS.

Singular Predicted Model Matrix

You can easily specify models with singular predicted model matrices, for example, by fixing diagonal elements of central matrices to 0. In such cases, you cannot compute maximum likelihood estimates (the ML function value F is not defined). Since

singular predicted model matrices can also occur temporarily in the minimization process, PROC CALIS tries in such cases to change the parameter estimates so that the predicted model matrix becomes positive definite. In such cases, the following message is displayed:

NOTE: Parameter set changed.

This process does not always work well, especially if there are fixed instead of variable diagonal elements in the central model matrices. A famous example where you cannot compute ML estimates is a component analysis with fewer components than given manifest variables. See the section “FACTOR Model Statement” on page 493 for more details. If you continue to get a singular predicted model matrix after changing initial values and optimization techniques, then your model is perhaps specified so that ML estimates cannot be computed.

Saving Computing Time

For large models, the most computing time is needed to compute the modification indices. If you don’t really need the Lagrange multipliers or multiple Wald test indices (the univariate Wald test indices are the same as the t values), using the NOMOD option can save a considerable amount of computing time.

Central Matrices with Negative Eigenvalues

A covariance matrix cannot have negative eigenvalues, since a negative eigenvalue means that some linear combination of the variables has negative variance. PROC CALIS displays a warning if a central model matrix has negative eigenvalues but does not actually compute the eigenvalues. Sometimes this warning can be triggered by 0 or very small positive eigenvalues that appear negative because of numerical error. If you want to be sure that the central model matrix you are fitting can be considered to be a variance-covariance matrix, you can use the SAS/IML command $VAL=EIGVAL(U)$ to compute the vector VAL of eigenvalues of matrix U .

Negative R^2 Values

The estimated squared multiple correlations R^2 of the endogenous variables are computed using the estimated error variances

$$R_i^2 = 1 - \frac{\widehat{var}(\zeta_i)}{\widehat{var}(\eta_i)}$$

If the model is a poor fit, it is possible that $\widehat{var}(\zeta_i) > \widehat{var}(\eta_i)$, which results in $R_i^2 < 0$.

Displayed Output

The output displayed by PROC CALIS depends on the statement used to specify the model. Since an analysis requested by the LINEQS or RAM statement implies the analysis of a structural equation model, more statistics can be computed and displayed than for a covariance structure analysis following the generalized COSAN model requested by the COSAN statement. The displayed output resulting from use of the FACTOR statement includes all the COSAN displayed output as well as more statistics displayed only when you specify the FACTOR statement. Since the displayed output using the RAM statement differs only in its form from that generated by the LINEQS statement, in this section distinctions are made between COSAN and LINEQS output only.

The unweighted least-squares and diagonally weighted least-squares estimation methods do not provide a sufficient statistical basis to provide the following output (neither displayed nor written to an OUTEST= data set):

- most of the fit indices
- approximate standard errors
- normalized or asymptotically standardized residuals
- modification indices
- information matrix
- covariance matrix of parameter estimates

The notation $\mathbf{S} = (s_{ij})$ is used for the analyzed covariance or correlation matrix, $\mathbf{C} = (c_{ij})$ for the predicted model matrix, \mathbf{W} for the weight matrix (for example, $\mathbf{W} = \mathbf{I}$ for ULS, $\mathbf{W} = \mathbf{S}$ for GLS, $\mathbf{W} = \mathbf{C}$ for ML estimates), \mathbf{X} for the vector of optimal parameter estimates, n for the number of manifest variables, t for the number of parameter estimates, and N for the sample size.

The output of PROC CALIS includes the following:

- COSAN and LINEQS: List of the matrices and their properties specified by the generalized COSAN model if you specify at least the PSHORT option.
- LINEQS: List of manifest variables that are not used in the specified model and that are automatically omitted from the analysis. Note that there is no automatic variable reduction with the COSAN or FACTOR statement. If necessary, you should use the VAR statement in these cases.
- LINEQS: List of the endogenous and exogenous variables specified by the LINEQS, STD, and COV statements if you specify at least the PSHORT option.
- COSAN: Initial values of the parameter matrices indicating positions of constants and parameters. The output, or at least the default output, is displayed if you specify the PINITIAL option.

- LINEQS: The set of structural equations containing the initial values and indicating constants and parameters, and output of the initial error variances and covariances. The output, or at least the default output, is displayed if you specify the PINITIAL option.
- COSAN and LINEQS: The weight matrix **W** is displayed if GLS, WLS, or DWLS estimation is used and you specify the PWEIGHT or PALL option.
- COSAN and LINEQS: General information about the estimation problem: number of observations (N), number of manifest variables (n), amount of independent information in the data matrix (information, $n(n + 1)/2$), number of terms and matrices in the specified generalized COSAN model, and number of parameters to be estimated (parameters, t). If there are no exogenous manifest variables, the difference between the amount of independent information ($n(n + 1)/2$) and the number of requested estimates (t) is equal to the degrees of freedom (df). A necessary condition for a model to be identified is that the degrees of freedom are nonnegative. The output, or at least the default output, is displayed if you specify the SIMPLE option.
- COSAN and LINEQS: Mean and Std Dev (standard deviation) of each variable if you specify the SIMPLE option, as well as skewness and kurtosis if the DATA= data set is a raw data set and you specify the KURTOSIS option.
- COSAN and LINEQS: Various coefficients of multivariate kurtosis and the numbers of observations that contribute most to the normalized multivariate kurtosis if the DATA= data set is a raw data set and the KURTOSIS option, or you specify at least the PRINT option. See the section “Measures of Multivariate Kurtosis” on page 544 for more information.
- COSAN and LINEQS: Covariance or correlation matrix to be analyzed and the value of its determinant if you specify the output option PCORR or PALL. A 0 determinant indicates a singular data matrix. In this case, the generalized least-squares estimates with default weight matrix **S** and maximum likelihood estimates cannot be computed.
- LINEQS: If exogenous manifest variables in the linear structural equation model are specified, then there is a one-to-one relationship between the given covariances and corresponding estimates in the central model matrix Φ or P . The output indicates which manifest variables are recognized as exogenous, that is, for which variables the entries in the central model matrix are set to fixed parameters. The output, or at least the default output, is displayed if you specify the PINITIAL option.
- COSAN and LINEQS: Vector of parameter names, initial values, and corresponding matrix locations, also indicating dependent parameter names used in your program statements that are not allocated to matrix locations and have no influence on the fit function. The output, or at least the default output, is displayed if you specify the PINITIAL option.
- COSAN and LINEQS: The pattern of variable and constant elements of the predicted moment matrix that is predetermined by the analysis model is displayed if there are significant differences between constant elements in the predicted model matrix and the data matrix and you specify at least the PSHORT option.

It is also displayed if you specify the PREDET option. The output indicates the differences between constant values in the predicted model matrix and the data matrix that is analyzed.

- COSAN and LINEQS: Special features of the optimization technique chosen if you specify at least the PSHORT option.
- COSAN and LINEQS: Optimization history if at least the PSHORT option is specified. For more details, see the section “Use of Optimization Techniques” on page 551.
- COSAN and LINEQS: Specific output requested by options in the NLOPTIONS statement; for example, parameter estimates, gradient, gradient of Lagrange function, constraints, Lagrange multipliers, projected gradient, Hessian, projected Hessian, Hessian of Lagrange function, Jacobian of nonlinear constraints.
- COSAN and LINEQS: The predicted model matrix and its determinant, if you specify the output option PCORR or PALL.
- COSAN and LINEQS: Residual and normalized residual matrix if you specify the RESIDUAL, or at least the PRINT option. The variance standardized or asymptotically standardized residual matrix can be displayed also. The average residual and the average off-diagonal residual are also displayed. See the section “Assessment of Fit” on page 536 for more details.
- COSAN and LINEQS: Rank order of the largest normalized residuals if you specify the RESIDUAL, or at least the PRINT option.
- COSAN and LINEQS: Bar chart of the normalized residuals if you specify the RESIDUAL, or at least the PRINT option.
- COSAN and LINEQS: Value of the fit function F . See the section “Estimation Criteria” on page 531 for more details. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: Goodness-of-fit index (GFI), adjusted goodness-of-fit index (AGFI), and root mean square residual (RMR) (Jöreskog and Sörbom 1985). See the section “Assessment of Fit” on page 536 for more details. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: Parsimonious goodness-of-fit index (PGFI) of Mulaik et al. (1989). See the section “Assessment of Fit” on page 536 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: Overall χ^2 , df , and Prob>Chi**2 if the METHOD= option is not ULS or DWLS. The χ^2 measure is the optimum function value F multiplied by $(N - 1)$ if a CORR or COV matrix is analyzed or multiplied by N if a UCORR or UCOV matrix is analyzed; χ^2 measures the likelihood ratio test statistic for the null hypothesis that the predicted matrix \mathbf{C} has the specified model structure against the alternative that \mathbf{C} is unconstrained. The notation Prob>Chi**2 means “the probability under the null hypothesis of obtaining a greater χ^2 statistic than that observed.” This output can be suppressed only by the NOPRINT option.

- COSAN and LINEQS: If METHOD= is not ULS or DWLS, the χ^2_0 value of the independence model and the corresponding degrees of freedom can be used (in large samples) to evaluate the gain of explanation by fitting the specific model (Bentler 1989). See the section “Assessment of Fit” on page 536 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: If METHOD= is not ULS or DWLS, the value of the Steiger & Lind (1980) root mean squared error of approximation (RMSEA) coefficient and the lower and upper limits of the confidence interval. The size of the confidence interval is defined by the option ALPHARMS= α , $0 \leq \alpha \leq 1$. The default is $\alpha = 0.1$, which corresponds to a 90% confidence interval. See the section “Assessment of Fit” on page 536 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: If the value of the METHOD= option is not ULS or DWLS, the value of the *probability of close fit* (Browne and Cudeck 1993). See the section “Assessment of Fit” on page 536 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: If the value of the METHOD= option is not ULS or DWLS, the value of the Browne & Cudeck (1993) expected cross validation (ECVI) index and the lower and upper limits of the confidence interval. The size of the confidence interval is defined by the option ALPHAECV= α , $0 \leq \alpha \leq 1$. The default is $\alpha = 0.1$, which corresponds to a 90% confidence interval. See the section “Assessment of Fit” on page 536 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: If the value of the METHOD= option is not ULS or DWLS, Bentler’s (1989) Comparative Fit Index. See the section “Assessment of Fit” on page 536 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: If you specify METHOD=ML or METHOD=GLS, the χ^2 value and corresponding probability adjusted by the relative kurtosis coefficient η_2 , which should be a close approximation of the χ^2 value for elliptically distributed data (Browne 1982). See the section “Assessment of Fit” on page 536 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: The Normal Theory Reweighted LS χ^2 Value is displayed if METHOD= ML. Instead of the function value F_{ML} , the reweighted goodness-of-fit function F_{GWLs} is used. See the section “Assessment of Fit” on page 536 for more detail.
- COSAN and LINEQS: Akaike’s Information Criterion if the value of the METHOD= option is not ULS or DWLS. See the section “Assessment of Fit” on page 536. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: Bozdogan’s (1987) Consistent Information Criterion, CAIC. See the section “Assessment of Fit” on page 536. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: Schwarz’s Bayesian Criterion (SBC) if the value of the METHOD= option is not ULS or DWLS (Schwarz 1978). See the section

“Assessment of Fit” on page 536. This output can be suppressed only by the NOPRINT option.

- COSAN and LINEQS: If the value of the METHOD= option is not ULS or DWLS, the following fit indices based on the overall χ^2 value are displayed:
 - McDonald’s (McDonald and Hartmann 1992) measure of centrality
 - Parsimonious index of James, Mulaik, and Brett (1982)
 - Z-Test of Wilson and Hilferty (1931)
 - Bentler and Bonett’s (1980) nonnormed coefficient
 - Bentler and Bonett’s (1980) normed coefficient
 - Bollen’s (1986) normed index ρ_1
 - Bollen’s (1989a) nonnormed index Δ_2

See the section “Assessment of Fit” on page 536 for more detail. This output can be suppressed only by the NOPRINT option.

- COSAN and LINEQS: Hoelter’s (1983) Critical N Index is displayed (Bollen 1989b, p. 277). See the section “Assessment of Fit” on page 536 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: Equations of linear dependencies among the parameters used in the model specification if the information matrix is recognized as singular at the final solution.
- COSAN: Model matrices containing the parameter estimates. Except for ULS or DWLS estimates, the approximate standard errors and t values are also displayed. This output is displayed if you specify the PESTIM option or at least the PSHORT option.
- LINEQS: Linear equations containing the parameter estimates. Except for ULS and DWLS estimates, the approximate standard errors and t values are also displayed. This output is displayed if you specify the PESTIM option, or at least the PSHORT option.
- LINEQS: Variances and covariances of the exogenous variables. This output is displayed if you specify the PESTIM option, or at least the PSHORT option.
- LINEQS: Linear equations containing the standardized parameter estimates. This output is displayed if you specify the PESTIM option, or at least the PSHORT option.
- LINEQS: Table of correlations among the exogenous variables. This output is displayed if you specify the PESTIM option, or at least the PSHORT option.
- LINEQS: Correlations among the exogenous variables. This output is displayed if you specify the PESTIM option, or at least the PSHORT option.
- LINEQS: Squared Multiple Correlations table, which displays the error variances of the endogenous variables. These are the diagonal elements of the predicted model matrix. Also displayed is the Total Variance and the R^2 values corresponding to all endogenous variables. See the section “Assessment of Fit” on page 536 for more detail. This output is displayed if you specify the PESTIM option, or at least the PSHORT option.

- **LINEQS**: If you specify the **PDETERM** or the **PALL** option, the total determination of all equations (**DETAE**), the total determination of the structural equations (**DETSE**), and the total determination of the manifest variables (**DETMV**) are displayed. See the section “Assessment of Fit” on page 536 for more details. If one of the determinants in the formulas is 0, the corresponding coefficient is displayed as a missing value. If there are structural equations, **PROC CALIS** also displays the Stability Coefficient of Reciprocal Causation, that is, the largest eigenvalue of the **BB'** matrix, where **B** is the causal coefficient matrix of the structural equations.
- **LINEQS**: The matrix of estimated covariances among the latent variables if you specify the **PLATCOV** option, or at least the **PRINT** option.
- **LINEQS**: The matrix of estimated covariances between latent and manifest variables used in the model if you specify the **PLATCOV** option, or at least the **PRINT** option.
- **LINEQS** and **FACTOR**: The matrix **FSR** of latent variable scores regression coefficients if you specify the **PLATCOV** option, or at least the **PRINT** option. The **FSR** matrix is a generalization of Lawley and Maxwell's (1971, p.109) factor scores regression matrix,

$$\mathbf{FSR} = \mathbf{C}_{yx} \mathbf{C}_{xx}^{-1}$$

where \mathbf{C}_{xx} is the $n \times n$ predicted model matrix (predicted covariances among manifest variables) and \mathbf{C}_{yx} is the $n_{lat} \times n$ matrix of the predicted covariances between latent and manifest variables. You can multiply the manifest observations by this matrix to estimate the scores of the latent variables used in your model.

- **LINEQS**: The matrix **TEF** of total effects if you specify the **TOTEFF** option, or at least the **PRINT** option. For the **LINEQS** model, the matrix of total effects is

$$\mathbf{TEF} = (\mathbf{I} - \boldsymbol{\beta})^{-1} \boldsymbol{\gamma} - (\mathbf{O} : \mathbf{I})$$

(For the **LISREL** model, refer to Jöreskog and Sörbom 1985) The matrix of indirect effects is displayed also.

- **FACTOR**: The matrix of rotated factor loadings and the orthogonal transformation matrix if you specify the **ROTATE=** and **PESTIM** options, or at least the **PSHORT** options.
- **FACTOR**: Standardized (rotated) factor loadings, variance estimates of endogenous variables, R^2 values, correlations among factors, and factor scores regression matrix, if you specify the **PESTIM** option, or at least the **PSHORT** option. The determination of manifest variables is displayed only if you specify the **PDETERM** option.
- **COSAN** and **LINEQS**: Univariate Lagrange multiplier and Wald test indices are displayed in matrix form if you specify the **MODIFICATION** (or **MOD**) or the **PALL** option. Those matrix locations that correspond to constants in the model in general contain three values: the value of the Lagrange multiplier, the

corresponding probability ($df = 1$), and the estimated change of the parameter value should the constant be changed to a parameter. If allowing the constant to be an estimated parameter would result in a singular information matrix, the string 'sing' is displayed instead of the Lagrange multiplier index. Those matrix locations that correspond to parameter estimates in the model contain the Wald test index and the name of the parameter in the model. See the section “Modification Indices” on page 560 for more detail.

- COSAN and LINEQS: Univariate Lagrange multiplier test indices for releasing equality constraints if you specify the MODIFICATION (or MOD) or the PALL option. See the section “Modification Indices” on page 560 for more detail.
- COSAN and LINEQS: Univariate Lagrange multiplier test indices for releasing active boundary constraints specified by the BOUNDS statement if you specify the MODIFICATION (or MOD) or the PALL option. See the section “Modification Indices” on page 560 for more detail.
- COSAN and LINEQS: If the MODIFICATION (or MOD) or the PALL option is specified, the stepwise multivariate Wald test for constraining estimated parameters to zero constants is performed as long as the univariate probability is larger than the value specified in the PMW= option (default PMW=0.05). See the section “Modification Indices” on page 560 for more details.

ODS Table Names

PROC CALIS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 15, “Using the Output Delivery System.”

Table 19.15. ODS Tables Created in PROC CALIS

ODS Table Name	Model ¹	Description	Option ²
AddParms	C, F, L, R	Additional parameters in the PARAMETERS statement	PINITIAL, or default
AsymStdRes	C, F, L, R	Asymptotically standardized residual matrix	RESIDUAL=, or PRINT
AveAsymStdRes	C, F, L, R	Average absolute asymptotically standardized residuals	RESIDUAL=, or PRINT
AveNormRes	C, F, L, R	Average absolute normalized residuals	RESIDUAL=, or PRINT
AveRawRes	C, F, L, R	Average absolute raw residuals	RESIDUAL=, or PRINT
AveVarStdRes	C, F, L, R	Average absolute variance standardized residuals	RESIDUAL=, or PRINT
CholeskyWeights	C, F, L, R	Cholesky factor of weights	PWEIGHT, or PALL
ContKurtosis	C, F, L, R	Contributions to kurtosis	KURTOSIS, or PRINT
ConvergenceStatus	C, F, L, R	Convergence status	PSHORT
CorrExog	L	Correlations among exogenous variables	PESTIM, or PSHORT
CorrParm	C, F, L, R	Correlations among parameter estimates	PCOVES, and default
CovMat	C, F, L, R	Assorted cov matrices	PCOVES, and default

Table 19.15. (continued)

ODS Table Name	Model ¹	Description	Option ²
DependParms	C, F, L, R	Dependent parameters (if specified by program statements)	PRIVEC, and default
Determination	L, F, R	Coefficients of determination	PDETERM, and default
DistAsymStdRes	C, F, L, R	Distribution of asymptotically standardized residuals	RESIDUAL=, or PRINT
DistNormRes	C, F, L, R	Distribution of normalized residuals	RESIDUAL=, or PRINT
DistRawRes	C, F, L, R	Distribution of residuals	RESIDUAL=, or PRINT
DistVarStdRes	C, F, L, R	Distribution of variance standardized residuals	RESIDUAL=, or PRINT
EndogenousVar	L	Endogenous variables	PESTIM, or PSHORT
EstCovExog	L	Estimated covariances among exogenous variables	PESTIM, or PSHORT
Estimates	C, F, L, R	Vector of estimates	PRIVEC
EstLatentEq	L	Estimated latent variable equations	PESTIM, or PSHORT
EstManifestEq	L	Estimated manifest variable equations	PESTIM, or PSHORT
EstParms	C, F	Estimated parameter matrix	PESTIM, or PSHORT
EstVarExog	L	Estimated variances of exogenous variables	PESTIM, or PSHORT
ExogenousVar	L	List of exogenous variables	PESTIM, or PSHORT
FACTCorrExog	F	Correlations among factors	PESTIM, or PSHORT
FactScoreCoef	F	Factor score regression coefficients	PESTIM, or PSHORT
FirstDer	C, F, L, R	Vector of the first partial derivatives	PRIVEC, and default
Fit	C, F, L, R	Fit statistics	PSUMMARY
GenModInfo	C, F, L, R	General modeling information	PSIMPLE, or default
Gradient	C, F, L, R	First partial derivatives (Gradient)	PRIVEC, and default
InCorr	C, F, L, R	Input correlation matrix	PCORR, or PALL
InCorrDet	C, F, L, R	Determinant of the input correlation matrix	PCORR, or PALL
InCov	C, F, L, R	Input covariance matrix	PCORR, or PALL
InCovDet	C, F, L, R	Determinant of the input covariance matrix	PCORR, or PALL
InCovExog	L	Input covariances among exogenous variables	PESTIM, or PSHORT
Indirect Effects	L, R	Indirect effects	TOTEFF, or PRINT
Information	C, F, L, R	Information matrix	PCOVES, and default
InitEstimates	C, F, L, R	Initial vector of parameter estimates	PINITIAL, or default
InitParms	C, F	Initial matrix of parameter estimates	PINITIAL, or default
InitParms	L, R	Initial matrix of parameter estimates	PRIMAT, and default
InitRAMEstimates	R	Initial RAM estimates	PESTIM, or PSHORT
InLatentEq	L	Input latent variable equations	PESTIM, or PSHORT
InManifestEq	L	Input manifest variable equations	PESTIM, or PSHORT
InSymmetric	C, F, L, R	Input symmetric matrix	PCORR, or PALL
InVarExog	L	Input variances of exogenous variables	PESTIM, or PSHORT
InvDiagWeights	C, F, L, R	Inverse of diagonal of weights	PWEIGHT, or PALL
IterHist	C, F, L, R	Iteration history	PSHORT

Table 19.15. (continued)

ODS Table Name	Model ¹	Description	Option ²
IterStart	C, F, L, R	Iteration start	PSHORT
IterStop	C, F, L, R	Iteration stop	PSHORT
Jacobian	C, F, L, R	Jacobi column pattern	PJACPAT
Kurtosis	C, F, L, R	Kurtosis, with raw data input	KURTOSIS, or PRINT
LagrangeBoundary	C, F, L, R	Lagrange, releasing active boundary constraints	MODIFICATION, or PALL
LagrangeEquality	C, F, L, R	Lagrange, releasing equality constraints	MODIFICATION, or PALL
LatentScoreCoef	L, R	Latent variable regression score coefficients	PLATCOV, or PRINT
ModelStatement	C, F, L, R	Model summary	PSHORT
ModIndices	C, F, L, R	Lagrange multiplier and Wald test statistics	MODIFICATION, or PALL
NormRes	C, F, L, R	Normalized residual matrix	RESIDUAL=, or PRINT
PredetElements	C, F, L, R	Predetermined elements	PREDET, or PALL
PredModel	C, F, L, R	Predicted model matrix	PCORR, or PALL
PredModelDet	C, F, L, R	Predicted model determinant	PCORR, or PALL
PredMomentLatent	L, R	Predicted latent variable moments	PLATCOV, or PRINT
PredMomentManLat	L, R	Predicted manifest and latent variable moments	PLATCOV, or PRINT
ProblemDescription	C, F, L, R	Problem description	PSHORT
RAMCorrExog	R	Correlations among exogenous variables	PESTIM, or PSHORT
RAMEstimates	R	RAM Final Estimates	PESTIM, or PSHORT
RAMStdEstimates	R	Standardized estimates	PESTIM, or PSHORT
RankAsymStdRes	C, F, L, R	Ranking of the largest asymptotically standardized residuals	RESIDUAL=, or PRINT
RankLagrange	C, F, L, R	Ranking of the largest Lagrange indices	RESIDUAL=, or PRINT
RankNormRes	C, F, L, R	Ranking of the largest normalized residuals	RESIDUAL=, or PRINT
RankRawRes	C, F, L, R	Ranking of the largest raw residuals	RESIDUAL=, or PRINT
RankVarStdRes	C, F, L, R	Ranking of the largest variance standardized residuals	RESIDUAL=, or PRINT
RawRes	C, F, L, R	Raw residual matrix	RESIDUAL=, or PRINT
RotatedLoadings	F	Rotated loadings, with ROTATE= option in FACTOR statement	PESTIM, or PSHORT
Rotation	F	Rotation Matrix, with ROTATE= option in FACTOR statement	PESTIM, or PSHORT
SetCovExog	L, R	Set covariance parameters for manifest exogenous variables	PINITIAL, or default
SimpleStatistics	C, F, L, R	Simple statistics, with raw data input	SIMPLE, or default
SqMultCorr	F, L, R	Squared multiple correlations	PESTIM, or PSHORT
Stability	L, R	Stability of reciprocal causation	PDETERM, and default
StdErrs	C, F, L, R	Vector of standard errors	PRIVEC, and default
StdLatentEq	L	Standardized latent variable equations	PESTIM, or PSHORT
StdLoadings	F	Standardized factor loadings	PESTIM, or PSHORT
StdManifestEq	L	Standardized manifest variable equations	PESTIM, or PSHORT

Table 19.15. (continued)

ODS Table Name	Model ¹	Description	Option ²
StructEq	L, R	Variables in the structural equations	PDETERM, and default
SumSqDif	C, F, L, R	Sum of squared differences of pre-determined elements	PREDET, or PALL
TotalEffects	L, R	Total effects	TOTEFF, or PRINT
tValues	C, F, L, R	Vector of t values	PRIVEC, and default
VarSelection	L, R	Manifest variables, if not all are used, selected for Modeling	default
VarStdRes	C, F, L, R	Variance standardized residual matrix	RESIDUAL=, or PRINT
WaldTest	C, F, L, R	Wald test	MODIFICATION, or PALL
Weights	C, F, L, R	Weight matrix	PWEIGHT, or PALL
WeightsDet	C, F, L, R	Determinant of the weight matrix	PWEIGHT, or PALL

1. Most CALIS output tables are specific to the model statement used. Keys: C: COSAN model, F: FACTOR model, L: LINEQS model, R: RAM model.

2. The printing options PALL, PRINT, “default”, PSHORT, and PSUMM form hierarchical levels of output control, with PALL including all the output enabled by the options at the lower levels, and so on. The “default” option means that NOPRINT is not specified. Therefore, in the table, for example, if PSHORT is the printing option for an output, PALL, PRINT, or “default” will also enable the same output printing.

Examples

Example 19.1. Path Analysis: Stability of Alienation

The following covariance matrix from Wheaton, Muthen, Alwin, and Summers (1977) has served to illustrate the performance of several implementations for the analysis of structural equation models. Two different models have been analyzed by an early implementation of LISREL and are mentioned in Jöreskog (1978). You also can find a more detailed discussion of these models in the LISREL VI manual (Jöreskog and Sörbom 1985). A slightly modified model for this covariance matrix is included in the EQS 2.0 manual (Bentler 1985, p. 28). The path diagram of this model is displayed in Figure 19.1. The same model is reanalyzed here by PROC CALIS. However, for the analysis with the EQS implementation, the last variable (V6) is rescaled by a factor of 0.1 to make the matrix less ill-conditioned. Since the Levenberg-Marquardt or Newton-Raphson optimization techniques are used with PROC CALIS, rescaling the data matrix is not necessary and, therefore, is not done here. The results reported here reflect the estimates based on the original covariance matrix.

```
data Wheaton(TYPE=COV);
title "Stability of Alienation";
title2 "Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)";
_type_ = 'cov'; input _name_ $ v1-v6;
label v1='Anomia (1967)' v2='Anomia (1971)' v3='Education'
      v4='Powerlessness (1967)' v5='Powerlessness (1971)'
```

```

        v6='Occupational Status Index';
    datalines;
v1    11.834      .      .      .      .      .
v2     6.947     9.364      .      .      .      .
v3     6.819     5.091    12.532      .      .      .
v4     4.783     5.028     7.495     9.986      .      .
v5    -3.839    -3.889    -3.841    -3.625     9.610      .
v6   -21.899   -18.831   -21.748   -18.775    35.522   450.288
;

proc calis cov data=Wheaton tech=nr edf=931 pall;
  Lineqs
    V1 =          F1                      + E1,
    V2 =      .833 F1                      + E2,
    V3 =          F2                      + E3,
    V4 =      .833 F2                      + E4,
    V5 =          F3                      + E5,
    V6 = Lamb (.5) F3                      + E6,
    F1 = Gam1(-.5) F3                      + D1,
    F2 = Beta (.5) F1 + Gam2(-.5) F3 + D2;
  Std
    E1-E6 = The1-The2 The1-The4 (6 * 3.),
    D1-D2 = Psi1-Psi2 (2 * 4.),
    F3     = Phi (6.) ;
  Cov
    E1 E3 = The5 (.2),
    E4 E2 = The5 (.2);
run;

```

The COV option in the PROC CALIS statement requests the analysis of the covariance matrix. Without the COV option, the correlation matrix would be computed and analyzed. Since no METHOD= option has been used, maximum likelihood estimates are computed by default. The TECH=NR option requests the Newton-Raphson optimization method. The PALL option produces the almost complete set of displayed output, as displayed in Output 19.1.1 through Output 19.1.11. Note that, when you specify the PALL option, you can produce large amounts of output. The PALL option is used in this example to show how you can get a wide spectrum of useful information from PROC CALIS.

Output 19.1.1 displays the model specification in matrix terms, followed by the lists of endogenous and exogenous variables. Equations and initial parameter estimates are also displayed. You can use this information to ensure that the desired model is the model being analyzed.

Output 19.1.1. Model Specification

```

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Pattern and Initial Values

LINEQS Model Statement

Matrix      Rows      Columns      -----Matrix Type-----
Term 1      1  _SEL_      6          17      SELECTION
            2  _BETA_     17         17      EQSBETA      IMINUSINV
            3  _GAMMA_    17          9      EQSGAMMA
            4  _PHI_      9          9      SYMMETRIC

The 8 Endogenous Variables

Manifest      v1 v2 v3 v4 v5 v6
Latent        F1 F2

The 9 Exogenous Variables

Manifest
Latent        F3
Error         E1 E2 E3 E4 E5 E6 D1 D2

```

```

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Pattern and Initial Values

v1      =  1.0000 F1  +  1.0000 E1
v2      =  0.8330 F1  +  1.0000 E2
v3      =  1.0000 F2  +  1.0000 E3
v4      =  0.8330 F2  +  1.0000 E4
v5      =  1.0000 F3  +  1.0000 E5
v6      =  0.5000 F3  +  1.0000 E6
Lamb

```

```

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Pattern and Initial Values

F1      =  -0.5000 F3  +  1.0000 D1
Gam1
F2      =  0.5000 F1  + -0.5000 F3  +  1.0000 D2
Beta      Gam2

```

```

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Pattern and Initial Values

Variances of Exogenous Variables

Variable Parameter      Estimate
-----
F3      Phi             6.00000
E1      The1            3.00000
E2      The2            3.00000
E3      The1            3.00000
E4      The2            3.00000
E5      The3            3.00000
E6      The4            3.00000
D1      Psi1            4.00000
D2      Psi2            4.00000

Covariances Among Exogenous Variables

Var1 Var2 Parameter      Estimate
-----
E1   E3   The5            0.20000
E2   E4   The5            0.20000

```

General modeling information and simple descriptive statistics are displayed in Output 19.1.2. Because the input data set contains only the covariance matrix, the means of the manifest variables are assumed to be zero. Note that this has no impact on the estimation, unless a mean structure model is being analyzed. The twelve parameter estimates in the model and their respective locations in the parameter matrices are also displayed. Each of the parameters, The1, The2, and The5, is specified for two elements in the parameter matrix _PHI_.

Output 19.1.2. Modeling Information, Simple Statistics and Parameter Vector

```

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Observations      932   Model Terms           1
Variables          6   Model Matrices       4
Informations       21   Parameters           12

Variable              Mean      Std Dev
-----
v1  Anomia (1967)      0         3.44006
v2  Anomia (1971)      0         3.06007
v3  Education          0         3.54006
v4  Powerlessness (1967) 0         3.16006
v5  Powerlessness (1971) 0         3.10000
v6  Occupational Status Index 0        21.21999

```

Stability of Alienation							
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)							
The CALIS Procedure							
Covariance Structure Analysis: Maximum Likelihood Estimation							
Covariances							
		v1	v2	v3	v4	v5	v6
v1	Anomia (1967)	11.83400000	6.94700000	6.81900000	4.78300000	-3.83900000	-21.89900000
v2	Anomia (1971)	6.94700000	9.36400000	5.09100000	5.02800000	-3.88900000	-18.83100000
v3	Education	6.81900000	5.09100000	12.53200000	7.49500000	-3.84100000	-21.74800000
v4	Powerlessness (1967)	4.78300000	5.02800000	7.49500000	9.98600000	-3.62500000	-18.77500000
v5	Powerlessness (1971)	-3.83900000	-3.88900000	-3.84100000	-3.62500000	9.61000000	35.52200000
v6	Occupational Status Index	-21.89900000	-18.83100000	-21.74800000	-18.77500000	35.52200000	450.28800000
Determinant			6080570	Ln	15.620609		

Stability of Alienation			
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)			
The CALIS Procedure			
Covariance Structure Analysis: Maximum Likelihood Estimation			
Vector of Initial Estimates			
	Parameter	Estimate	Type
1	Beta	0.50000	Matrix Entry: _BETA_[8:7]
2	Lamb	0.50000	Matrix Entry: _GAMMA_[6:1]
3	Gam1	-0.50000	Matrix Entry: _GAMMA_[7:1]
4	Gam2	-0.50000	Matrix Entry: _GAMMA_[8:1]
5	Phi	6.00000	Matrix Entry: _PHI_[1:1]
6	The1	3.00000	Matrix Entry: _PHI_[2:2] _PHI_[4:4]
7	The2	3.00000	Matrix Entry: _PHI_[3:3] _PHI_[5:5]
8	The5	0.20000	Matrix Entry: _PHI_[4:2] _PHI_[5:3]
9	The3	3.00000	Matrix Entry: _PHI_[6:6]
10	The4	3.00000	Matrix Entry: _PHI_[7:7]
11	Psi1	4.00000	Matrix Entry: _PHI_[8:8]
12	Psi2	4.00000	Matrix Entry: _PHI_[9:9]

PROC CALIS examines whether each element in the moment matrix is modeled by the parameters defined in the model. If an element is not structured by the model parameters, it is predetermined by its observed value. This occurs, for example, when there are exogenous manifest variables in the model. If present, the predetermined values of the elements will be displayed. In the current example, the '.' displayed for all elements in the predicted moment matrix (Output 19.1.3) indicates that there are no predetermined elements in the model.

Output 19.1.3. Predetermined Elements

Stability of Alienation							
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)							
The CALIS Procedure							
Covariance Structure Analysis: Maximum Likelihood Estimation							
Predetermined Elements of the Predicted Moment Matrix							
		v1	v2	v3	v4	v5	v6
v1	Anomia (1967)
v2	Anomia (1971)
v3	Education
v4	Powerlessness (1967)
v5	Powerlessness (1971)
v6	Occupational Status Index
Sum of Squared Differences				0			

Output 19.1.4 displays the optimization information. You can check this table to determine whether the convergence criterion is satisfied. PROC CALIS displays an error message when problematic solutions are encountered.

Output 19.1.4. Optimization

Stability of Alienation									
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)									
The CALIS Procedure									
Covariance Structure Analysis: Maximum Likelihood Estimation									
Parameter Estimates					12				
Functions (Observations)					21				
Optimization Start									
Active Constraints			0		Objective Function			119.33282242	
Max Abs Gradient Element			74.016932345						

The predicted model matrix is displayed next, followed by a list of model test statistics or fit indices (Output 19.1.5). Depending on your modeling philosophy, some indices may be preferred to others. In this example, all indices and test statistics point to a good fit of the model.

Output 19.1.5. Predicted Model Matrix and Fit Statistics

Stability of Alienation							
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)							
The CALIS Procedure							
Covariance Structure Analysis: Maximum Likelihood Estimation							
Predicted Model Matrix							
		v1	v2	v3	v4	v5	v6
v1	Anomia (1967)	11.90390632	6.91059048	6.83016211	4.93499582	-4.16791157	-22.3768816
v2	Anomia (1971)	6.91059048	9.35145064	4.93499582	5.01664889	-3.47187034	-18.6399424
v3	Education	6.83016211	4.93499582	12.61574998	7.50355625	-4.06565606	-21.8278873
v4	Powerlessness (1967)	4.93499582	5.01664889	7.50355625	9.84539112	-3.38669150	-18.1826302
v5	Powerlessness (1971)	-4.16791157	-3.47187034	-4.06565606	-3.38669150	9.61000000	35.5219999
v6	Occupational Status Index	-22.37688158	-18.63994236	-21.82788734	-18.18263015	35.52199986	450.2879993
Determinant		6169285		Ln	15.635094		

Stability of Alienation	
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)	
The CALIS Procedure	
Covariance Structure Analysis: Maximum Likelihood Estimation	
Fit Function	0.0145
Goodness of Fit Index (GFI)	0.9953
GFI Adjusted for Degrees of Freedom (AGFI)	0.9890
Root Mean Square Residual (RMR)	0.2281
Parsimonious GFI (Mulaik, 1989)	0.5972
Chi-Square	13.4851
Chi-Square DF	9
Pr > Chi-Square	0.1419
Independence Model Chi-Square	2131.4
Independence Model Chi-Square DF	15
RMSEA Estimate	0.0231
RMSEA 90% Lower Confidence Limit	.
RMSEA 90% Upper Confidence Limit	0.0470
ECVI Estimate	0.0405
ECVI 90% Lower Confidence Limit	.
ECVI 90% Upper Confidence Limit	0.0556
Probability of Close Fit	0.9705
Bentler's Comparative Fit Index	0.9979
Normal Theory Reweighted LS Chi-Square	13.2804
Akaike's Information Criterion	-4.5149
Bozdogan's (1987) CAIC	-57.0509
Schwarz's Bayesian Criterion	-48.0509
McDonald's (1989) Centrality	0.9976
Bentler & Bonett's (1980) Non-normed Index	0.9965
Bentler & Bonett's (1980) NFI	0.9937
James, Mulaik, & Brett (1982) Parsimonious NFI	0.5962
Z-Test of Wilson & Hilferty (1931)	1.0754
Bollen (1986) Normed Index Rho1	0.9895
Bollen (1988) Non-normed Index Delta2	0.9979
Hoelter's (1983) Critical N	1170

PROC CALIS can perform a detailed residual analysis. Large residuals may indicate misspecification of the model. In Output 19.1.6 for example, note the table for the 10 largest asymptotically standardized residuals. As the table shows, the specified model performs the poorest concerning the variable V5 and its covariance with V2, V1, and V3. This may be the result of a misspecification of the model equation for

V5. However, because the model fit is quite good, such a possible misspecification may have no practical significance and is not a serious concern in the analysis.

Output 19.1.6. Residual Analysis

```

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Raw Residual Matrix

      v1      v2      v3      v4      v5      v6
v1 Anomia (1967)      -.0699063150  0.0364095216  -.0111621061  -.1519958205  0.3289115712  0.4778815840
v2 Anomia (1971)      0.0364095216  0.0125493646  0.1560041795  0.0113511059  -.4171296612  -.1910576405
v3 Education          -.0111621061  0.1560041795  -.0837499788  -.0085562504  0.2246560598  0.0798873380
v4 Powerlessness (1967) -.1519958205  0.0113511059  -.0085562504  0.1406088766  -.2383085022  -.5923698474
v5 Powerlessness (1971) 0.3289115712  -.4171296612  0.2246560598  -.2383085022  0.0000000000  0.0000000000
v6 Occupational Status Index 0.4778815840  -.1910576405  0.0798873380  -.5923698474  0.0000000000  0.0000000000

Average Absolute Residual      0.153928
Average Off-diagonal Absolute Residual 0.195045

```

```

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Rank Order of the 10 Largest Raw Residuals

      Row      Column      Residual
v6      v4      -0.59237
v6      v1      0.47788
v5      v2      -0.41713
v5      v1      0.32891
v5      v4      -0.23831
v5      v3      0.22466
v6      v2      -0.19106
v3      v2      0.15600
v4      v1      -0.15200
v4      v4      0.14061

```

```

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Asymptotically Standardized Residual Matrix

      v1      v2      v3      v4      v5      v6
v1 Anomia (1967)      -0.308548787  0.526654452  -0.056188826  -0.865070455  2.553366366  0.464866661
v2 Anomia (1971)      0.526654452  0.054363484  0.876120855  0.057354415  -2.763708659  -0.170127806
v3 Education          -0.056188826  0.876120855  -0.354347092  -0.121874301  1.697931678  0.070202664
v4 Powerlessness (1967) -0.865070455  0.057354415  -0.121874301  0.584930625  -1.557412695  -0.495982427
v5 Powerlessness (1971) 2.553366366  -2.763708659  1.697931678  -1.557412695  0.000000000  0.000000000
v6 Occupational Status Index 0.464866661  -0.170127806  0.070202664  -0.495982427  0.000000000  0.000000000

Average Standardized Residual      0.646622
Average Off-diagonal Standardized Residual 0.818457

```

```

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Rank Order of the 10 Largest Asymptotically Standardized Residuals

```

Row	Column	Residual
v5	v2	-2.76371
v5	v1	2.55337
v5	v3	1.69793
v5	v4	-1.55741
v3	v2	0.87612
v4	v1	-0.86507
v4	v4	0.58493
v2	v1	0.52665
v6	v4	-0.49598
v6	v1	0.46487

```

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Distribution of Asymptotically Standardized Residuals

Each * Represents 1 Residuals

```

-----Range-----		Freq	Percent	
-3.00000	-2.75000	1	4.76	*
-2.75000	-2.50000	0	0.00	
-2.50000	-2.25000	0	0.00	
-2.25000	-2.00000	0	0.00	
-2.00000	-1.75000	0	0.00	
-1.75000	-1.50000	1	4.76	*
-1.50000	-1.25000	0	0.00	
-1.25000	-1.00000	0	0.00	
-1.00000	-0.75000	1	4.76	*
-0.75000	-0.50000	0	0.00	
-0.50000	-0.25000	3	14.29	***
-0.25000	0	3	14.29	***
0	0.25000	6	28.57	*****
0.25000	0.50000	1	4.76	*
0.50000	0.75000	2	9.52	**
0.75000	1.00000	1	4.76	*
1.00000	1.25000	0	0.00	
1.25000	1.50000	0	0.00	
1.50000	1.75000	1	4.76	*
1.75000	2.00000	0	0.00	
2.00000	2.25000	0	0.00	
2.25000	2.50000	0	0.00	
2.50000	2.75000	1	4.76	*

Output 19.1.7 displays the equations and parameter estimates. Each parameter estimate is displayed with its standard error and the corresponding *t* ratio. As a general rule, a *t* ratio larger than 2 represents a statistically significant departure from 0. From these results, it is observed that both F1 (Alienation 1967) and F2 (Alienation 1971)

are regressed negatively on F3 (Socioeconomic Status), and F1 has a positive effect on F2. The estimates and significance tests for the variance and covariance of the exogenous variables are also displayed.

Output 19.1.7. Equations and Parameter Estimates

```

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

v1      = 1.0000 F1      + 1.0000 E1
v2      = 0.8330 F1      + 1.0000 E2
v3      = 1.0000 F2      + 1.0000 E3
v4      = 0.8330 F2      + 1.0000 E4
v5      = 1.0000 F3      + 1.0000 E5
v6      = 5.3688*F3      + 1.0000 E6
Std Err 0.4337 Lamb
t Value 12.3788

```

```

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

F1      = -0.6299*F3      + 1.0000 D1
Std Err 0.0563 Gam1
t Value -11.1809
F2      = 0.5931*F1      + -0.2409*F3      + 1.0000 D2
Std Err 0.0468 Beta      0.0549 Gam2
t Value 12.6788      -4.3885

```


Stability of Alienation				
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)				
The CALIS Procedure				
Covariance Structure Analysis: Maximum Likelihood Estimation				
Variances of Exogenous Variables				
Variable	Parameter	Estimate	Standard Error	t Value
F3	Phi	6.61632	0.63914	10.35
E1	The1	3.60788	0.20092	17.96
E2	The2	3.59493	0.16448	21.86
E3	The1	3.60788	0.20092	17.96
E4	The2	3.59493	0.16448	21.86
E5	The3	2.99368	0.49861	6.00
E6	The4	259.57580	18.31150	14.18
D1	Psi1	5.67047	0.42301	13.41
D2	Psi2	4.51480	0.33532	13.46
Covariances Among Exogenous Variables				
Var1	Var2	Parameter	Standard Error	t Value
E1	E3	The5	0.90580	7.44
E2	E4	The5	0.90580	7.44

The measurement scale of variables is often arbitrary. Therefore, it can be useful to look at the standardized equations produced by PROC CALIS. Output 19.1.8 displays the standardized equations and predicted moments. From the standardized structural equations for F1 and F2, you can conclude that SES (F3) has a larger impact on earlier Alienation (F1) than on later Alienation (F3).

The squared multiple correlation for each equation, the correlation among the exogenous variables, and the covariance matrices among the latent variables and between the observed and the latent variables help to describe the relationships among all variables.

Output 19.1.8. Standardized Equations and Predicted Moments

```

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

v1      =  0.8348 F1  +  0.5505 E1
v2      =  0.7846 F1  +  0.6200 E2
v3      =  0.8450 F2  +  0.5348 E3
v4      =  0.7968 F2  +  0.6043 E4
v5      =  0.8297 F3  +  0.5581 E5
v6      =  0.6508 F3  +  0.7593 E6
Lamb

```

```

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

F1      =  -0.5626 F3  +  0.8268 D1
          Gam1
F2      =  0.5692 F1  + -0.2064 F3  +  0.7080 D2
          Beta          Gam2

```

```

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Squared Multiple Correlations

Variable      Error      Total
              Variance    Variance    R-Square
1  v1          3.60788     11.90391     0.6969
2  v2          3.59493     9.35145     0.6156
3  v3          3.60788     12.61575     0.7140
4  v4          3.59493     9.84539     0.6349
5  v5          2.99368     9.61000     0.6885
6  v6        259.57580    450.28800     0.4235
7  F1          5.67047     8.29603     0.3165
8  F2          4.51480     9.00787     0.4988

Correlations Among Exogenous Variables

Var1 Var2 Parameter      Estimate
E1   E3   The5           0.25106
E2   E4   The5           0.25197

```

```

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Predicted Moments of Latent Variables

      F1              F2              F3
F1      8.296026985      5.924364730      -4.167911571
F2      5.924364730      9.007870649      -4.065656060
F3     -4.167911571     -4.065656060      6.616317547

Predicted Moments between Manifest and Latent Variables

      F1              F2              F3
v1      8.29602698      5.92436473      -4.16791157
v2      6.91059048      4.93499582      -3.47187034
v3      5.92436473      9.00787065      -4.06565606
v4      4.93499582      7.50355625      -3.38669150
v5     -4.16791157     -4.06565606      6.61631755
v6     -22.37688158     -21.82788734     35.52199986

```

Output 19.1.9 displays the latent variable score regression coefficients that produce the latent variable scores. Each latent variable is expressed as a linear combination of the observed variables. See Chapter 57, “The SCORE Procedure,” for more information on the creation of latent variable scores. Note that the total effects and indirect effects of the exogenous variables are also displayed.

Output 19.1.9. Latent Variable Score Regression, Direct and Indirect Effects

Stability of Alienation				
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)				
The CALIS Procedure				
Covariance Structure Analysis: Maximum Likelihood Estimation				
Latent Variable Score Regression Coefficients				
		F1	F2	F3
v1	Anomia (1967)	0.4131113567	0.0482681051	-.0521264408
v2	Anomia (1971)	0.3454029627	0.0400143300	-.0435560637
v3	Education	0.0526632293	0.4306175653	-.0399927539
v4	Powerlessness (1967)	0.0437036855	0.3600452776	-.0334000265
v5	Powerlessness (1971)	-.0749215200	-.0639697183	0.5057060770
v6	Occupational Status Index	-.0046390513	-.0039609288	0.0313127184
Total Effects				
	F3	F1	F2	
v1	-0.629944307	1.000000000	0.000000000	
v2	-0.524743608	0.833000000	0.000000000	
v3	-0.614489258	0.593112208	1.000000000	
v4	-0.511869552	0.494062469	0.833000000	
v5	1.000000000	0.000000000	0.000000000	
v6	5.368847492	0.000000000	0.000000000	
F1	-0.629944307	0.000000000	0.000000000	
F2	-0.614489258	0.593112208	0.000000000	
Indirect Effects				
	F3	F1	F2	
v1	-.6299443069	0.0000000000	0	
v2	-.5247436076	0.0000000000	0	
v3	-.6144892580	0.5931122083	0	
v4	-.5118695519	0.4940624695	0	
v5	0.0000000000	0.0000000000	0	
v6	0.0000000000	0.0000000000	0	
F1	0.0000000000	0.0000000000	0	
F2	-.3736276589	0.0000000000	0	

PROC CALIS can display Lagrange multiplier and Wald statistics for model modifications. Modification indices are displayed for each parameter matrix. Only the Lagrange multiplier statistics have significance levels and approximate changes of values displayed. The significance level of the Wald statistic for a given parameter is the same as that shown in the equation output. An insignificant p -value for a Wald statistic means that the corresponding parameter can be dropped from the model without significantly worsening the fit of the model.

A significant p -value for a Lagrange multiplier test indicates that the model would achieve a better fit if the corresponding parameter is free. To aid in determining significant results, PROC CALIS displays the rank order of the ten largest Lagrange multiplier statistics. For example, [E5:E2] in the `_PHI_` matrix is associated with the largest Lagrange multiplier statistic; the associated p -value is 0.0067. This means that adding a parameter for the covariance between E5 and E2 will lead to a significantly better fit of the model. However, adding parameters indiscriminately can result in

specification errors. An over-fitted model may not perform well with future samples. As always, the decision to add parameters should be accompanied with consideration and knowledge of the application area.

Output 19.1.10. Lagrange Multiplier and Wald Tests

Stability of Alienation									
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)									
The CALIS Procedure									
Covariance Structure Analysis: Maximum Likelihood Estimation									
Lagrange Multiplier and Wald Test Indices _PHI_[9:9]									
Symmetric Matrix									
Univariate Tests for Constant Constraints									
Lagrange Multiplier or Wald Index / Probability / Approx Change of Value									
	F3	E1	E2	E3	E4	E5	E6	D1	D2
F3	107.1619	3.3903	3.3901	0.5752	0.5753
	.	0.0656	0.0656	0.4482	0.4482
	.	0.5079	-0.4231	0.2090	-0.1741
	[Phi]					Sing	Sing	Sing	Sing
E1	3.3903	322.4501	0.1529	55.4237	1.2037	5.8025	0.7398	0.4840	0.0000
	0.0656	.	0.6958	.	0.2726	0.0160	0.3897	0.4866	0.9961
	0.5079	.	0.0900	.	-0.3262	0.5193	-1.2587	0.2276	0.0014
		[The1]		[The5]					
E2	3.3901	0.1529	477.6768	0.5946	55.4237	7.3649	1.4168	0.4840	0.0000
	0.0656	0.6958	.	0.4406	.	0.0067	0.2339	0.4866	0.9961
	-0.4231	0.0900	.	0.2328	.	-0.5060	1.5431	-0.1896	-0.0011
			[The2]		[The5]				
E3	0.5752	55.4237	0.5946	322.4501	0.1528	1.5982	0.0991	1.1825	0.5942
	0.4482	.	0.4406	.	0.6958	0.2062	0.7529	0.2768	0.4408
	0.2090	.	0.2328	.	-0.0900	0.2709	-0.4579	0.2984	-0.2806
		[The5]		[The1]					
E4	0.5753	1.2037	55.4237	0.1528	477.6768	1.2044	0.0029	1.1825	0.5942
	0.4482	0.2726	.	0.6958	.	0.2724	0.9568	0.2768	0.4408
	-0.1741	-0.3262	.	-0.0900	.	-0.2037	0.0700	-0.2486	0.2338
			[The5]		[The2]				
E5	.	5.8025	7.3649	1.5982	1.2044	36.0486	.	0.1033	0.1035
	.	0.0160	0.0067	0.2062	0.2724	.	.	0.7479	0.7477
	.	0.5193	-0.5060	0.2709	-0.2037	.	.	-0.2776	0.1062
	Sing					[The3]	Sing		
E6	.	0.7398	1.4168	0.0991	0.0029	.	200.9466	0.1034	0.1035
	.	0.3897	0.2339	0.7529	0.9568	.	.	0.7478	0.7477
	.	-1.2587	1.5431	-0.4579	0.0700	.	.	1.4906	-0.5700
	Sing					Sing	[The4]		
D1	.	0.4840	0.4840	1.1825	1.1825	0.1033	0.1034	179.6950	.
	.	0.4866	0.4866	0.2768	0.2768	0.7479	0.7478	.	.
	.	0.2276	-0.1896	0.2984	-0.2486	-0.2776	1.4906	.	.
	Sing							[Psi1]	Sing
D2	.	0.0000	0.0000	0.5942	0.5942	0.1035	0.1035	.	181.2787
	.	0.9961	0.9961	0.4408	0.4408	0.7477	0.7477	.	.
	.	0.0014	-0.0011	-0.2806	0.2338	0.1062	-0.5700	.	.
	Sing							Sing	[Psi2]

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Rank Order of the 10 Largest Lagrange Multipliers in _PHI_

Row	Column	Chi-Square	Pr > ChiSq
E5	E2	7.36486	0.0067
E5	E1	5.80246	0.0160
E1	F3	3.39030	0.0656
E2	F3	3.39013	0.0656
E5	E3	1.59820	0.2062
E6	E2	1.41677	0.2339
E5	E4	1.20437	0.2724
E4	E1	1.20367	0.2726
D1	E3	1.18251	0.2768
D1	E4	1.18249	0.2768

```

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Lagrange Multiplier and Wald Test Indices _GAMMA_[8:1]
General Matrix
Univariate Tests for Constant Constraints
Lagrange Multiplier or Wald Index / Probability / Approx Change of Value

F3
v1      3.3903
        0.0656
        0.0768

v2      3.3901
        0.0656
       -0.0639

v3      0.5752
        0.4482
        0.0316

v4      0.5753
        0.4482
       -0.0263

v5      .
        .
        .
        Sing

v6      153.2354
        .
        .
        [Lamb]

F1      125.0132
        .
        .
        [Gam1]

F2      19.2585
        .
        .
        [Gam2]

```

```

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Rank Order of the 4 Largest Lagrange Multipliers in _GAMMA_

```

Row	Column	Chi-Square	Pr > ChiSq
v1	F3	3.39030	0.0656
v2	F3	3.39013	0.0656
v4	F3	0.57526	0.4482
v3	F3	0.57523	0.4482

```

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Lagrange Multiplier and Wald Test Indices _BETA_[8:8]
General Matrix
Identity-Minus-Inverse Model Matrix
Univariate Tests for Constant Constraints
Lagrange Multiplier or Wald Index / Probability / Approx Change of Value

```

	v1	v2	v3	v4	v5	v6	F1	F2
v1	.	0.1647	0.0511	0.8029	5.4083	0.1233	0.4047	0.4750
	.	0.6849	0.8212	0.3702	0.0200	0.7255	0.5247	0.4907
	.	-0.0159	-0.0063	-0.0284	0.0697	0.0015	-0.0257	-0.0239
	Sing							
v2	0.5957	.	0.6406	0.0135	5.8858	0.0274	0.4047	0.4750
	0.4402	.	0.4235	0.9076	0.0153	0.8686	0.5247	0.4907
	0.0218	.	0.0185	0.0032	-0.0609	-0.0006	0.0214	0.0199
		Sing						
v3	0.3839	0.3027	.	0.1446	1.1537	0.0296	0.1588	0.0817
	0.5355	0.5822	.	0.7038	0.2828	0.8634	0.6902	0.7750
	0.0178	0.0180	.	-0.0145	0.0322	0.0007	0.0144	-0.0110
			Sing					
v4	0.4487	0.2519	0.0002	.	0.9867	0.1442	0.1588	0.0817
	0.5030	0.6157	0.9877	.	0.3206	0.7041	0.6903	0.7750
	-0.0160	-0.0144	-0.0004	.	-0.0249	-0.0014	-0.0120	0.0092
				Sing				
v5	5.4085	8.6455	2.7123	2.1457	.	.	0.1033	0.1035
	0.0200	0.0033	0.0996	0.1430	.	.	0.7479	0.7476
	0.1242	-0.1454	0.0785	-0.0674	.	.	-0.0490	0.0329
					Sing	Sing		
v6	0.4209	1.4387	0.3044	0.0213	.	.	0.1034	0.1035
	0.5165	0.2304	0.5811	0.8841	.	.	0.7478	0.7477
	-0.2189	0.3924	-0.1602	0.0431	.	.	0.2629	-0.1765
					Sing	Sing		
F1	1.0998	1.1021	1.6114	1.6128	0.1032	0.1035	.	.
	0.2943	0.2938	0.2043	0.2041	0.7480	0.7477	.	.
	0.0977	-0.0817	0.0993	-0.0831	-0.0927	0.0057	.	.
							Sing	Sing
F2	0.0193	0.0194	0.4765	0.4760	0.1034	0.1035	160.7520	.
	0.8896	0.8892	0.4900	0.4902	0.7477	0.7477	.	.
	-0.0104	0.0087	-0.0625	0.0522	0.0355	-0.0022	.	.
							[Beta]	Sing


```

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Rank Order of the 10 Largest Lagrange Multipliers in _BETA_

```

Row	Column	Chi-Square	Pr > ChiSq
v5	v2	8.64546	0.0033
v2	v5	5.88576	0.0153
v5	v1	5.40848	0.0200
v1	v5	5.40832	0.0200
v5	v3	2.71233	0.0996
v5	v4	2.14572	0.1430
F1	v4	1.61279	0.2041
F1	v3	1.61137	0.2043
v6	v2	1.43867	0.2304
v3	v5	1.15372	0.2828

When you specify equality constraints, PROC CALIS displays Lagrange multiplier tests for releasing the constraints. In the current example, none of the three constraints achieve a p -value smaller than 0.05. This means that releasing the constraints may not lead to a significantly better fit of the model. Therefore, all constraints are retained in the model.

Output 19.1.11. Tests for Equality Constraints

```

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Univariate Lagrange Multiplier Test for Releasing Equality Constraints

```

Equality Constraint	-----Changes-----	Chi-Square	Pr > ChiSq
[E1:E1] = [E3:E3]	0.0293 -0.0308	0.02106	0.8846
[E2:E2] = [E4:E4]	-0.1342 0.1388	0.69488	0.4045
[E3:E1] = [E4:E2]	0.2468 -0.1710	1.29124	0.2558

The model is specified using the LINEQS, STD, and COV statements. The section “Getting Started” on page 448 also contains the COSAN and RAM specifications of this model. These model specifications would give essentially the same results.

```

proc calis cov data=Wheaton tech=nr edf=931;
  Cosan J(9, Ide) * A(9, Gen, Imi) * P(9, Sym);
  Matrix A
    [ ,7] = 1. .833 5 * 0. Beta (.5) ,
    [ ,8] = 2 * 0. 1. .833 ,
    [ ,9] = 4 * 0. 1. Lamb Gam1-Gam2 (.5 2 * -.5);
  Matrix P
    [1,1] = The1-The2 The1-The4 (6 * 3.) ,
    [7,7] = Psi1-Psi2 Phi (2 * 4. 6.) ,
    [3,1] = The5 (.2) ,

```

```

          [4,2] = The5 (.2) ;
Vnames J V1-V6 F1-F3 ,
        A = J ,
        P E1-E6 D1-D3 ;
run;

proc calis cov data=Wheaton tech=nr edf=931;
  Ram
    1 1 7 1. ,
    1 2 7 .833 ,
    1 3 8 1. ,
    1 4 8 .833 ,
    1 5 9 1. ,
    1 6 9 .5 Lamb ,
    1 7 9 -.5 Gam1 ,
    1 8 7 .5 Beta ,
    1 8 9 -.5 Gam2 ,
    2 1 1 3. The1 ,
    2 2 2 3. The2 ,
    2 3 3 3. The1 ,
    2 4 4 3. The2 ,
    2 5 5 3. The3 ,
    2 6 6 3. The4 ,
    2 1 3 .2 The5 ,
    2 2 4 .2 The5 ,
    2 7 7 4. Psi1 ,
    2 8 8 4. Psi2 ,
    2 9 9 6. Phi ;
  Vnames 1 F1-F3,
          2 E1-E6 D1-D3;
run;

```

Example 19.2. Simultaneous Equations with Intercept

The demand-and-supply food example of Kmenta (1971, pp. 565, 582) is used to illustrate the use of PROC CALIS for the estimation of intercepts and coefficients of simultaneous equations. The model is specified by two simultaneous equations containing two endogenous variables Q and P and three exogenous variables D , F , and Y ,

$$Q_t(\text{demand}) = \alpha_1 + \beta_1 P_t + \gamma_1 D_t$$

$$P_t(\text{supply}) = \alpha_2 + \beta_2 P_t + \gamma_2 F_t + \gamma_3 Y_t$$

for $t = 1, \dots, 20$.

The LINEQS statement requires that each endogenous variable appear on the left-hand side of exactly one equation. Instead of analyzing the system

$$\mathbf{B}^* \boldsymbol{\eta} = \boldsymbol{\Gamma} \boldsymbol{\xi} + \boldsymbol{\zeta}$$

PROC CALIS analyzes the equivalent system

$$\eta = \mathbf{B}\eta + \mathbf{\Gamma}\xi + \zeta$$

with $\mathbf{B}^* = \mathbf{I} - \mathbf{B}$. This requires that one of the preceding equations be solved for P_t . Solving the second equation for P_t yields

$$P_t = \frac{1}{\beta_2} Q_t - \frac{\alpha_2}{\beta_2} - \frac{\gamma_2}{\beta_2} F_t - \frac{\gamma_3}{\beta_2} Y_t$$

You can estimate the intercepts of a system of simultaneous equations by applying PROC CALIS on the uncorrected covariance (UCOV) matrix of the data set that is augmented by an additional constant variable with the value 1. In the following example, the uncorrected covariance matrix is augmented by an additional variable INTERCEPT by using the AUGMENT option. The PROC CALIS statement contains the options UCOV and AUG to compute and analyze an augmented UCOV matrix from the input data set FOOD.

```
data food;
  title 'Food example of KMENTA(1971, p.565 & 582)';
  input Q P D F Y;
  label Q='Food Consumption per Head'
        P='Ratio of Food Prices to General Price'
        D='Disposable Income in Constant Prices'
        F='Ratio of Preceding Years Prices'
        Y='Time in Years 1922-1941';
  datalines;
    98.485 100.323 87.4 98.0 1
    99.187 104.264 97.6 99.1 2
    102.163 103.435 96.7 99.1 3
    101.504 104.506 98.2 98.1 4
    104.240 98.001 99.8 110.8 5
    103.243 99.456 100.5 108.2 6
    103.993 101.066 103.2 105.6 7
    99.900 104.763 107.8 109.8 8
    100.350 96.446 96.6 108.7 9
    102.820 91.228 88.9 100.6 10
    95.435 93.085 75.1 81.0 11
    92.424 98.801 76.9 68.6 12
    94.535 102.908 84.6 70.9 13
    98.757 98.756 90.6 81.4 14
    105.797 95.119 103.1 102.3 15
    100.225 98.451 105.1 105.0 16
    103.522 86.498 96.4 110.5 17
    99.929 104.016 104.4 92.5 18
    105.223 105.769 110.7 89.3 19
    106.232 113.490 127.1 93.0 20
  ;
```

```

proc calis ucov aug data=food pshort;
  Title2 'Compute ML Estimates With Intercept';
  Lineqs
    Q = alf1 Intercept + alf2 P + alf3 D + E1,
    P = gam1 Intercept + gam2 Q + gam3 F + gam4 Y + E2;
  Std
    E1-E2 = eps1-eps2;
  Cov
    E1-E2 = eps3;
  Bounds
    eps1-eps2 >= 0. ;
run;

```

The following, essentially equivalent model definition uses program code to reparameterize the model in terms of the original equations; the output is displayed in Output 19.2.1.

```

proc calis data=food ucov aug pshort;
  Lineqs
    Q = alpha1 Intercept + beta1 P + gamma1 D + E1,
    P = alpha2_b Intercept + gamma2_b F + gamma3_b Y + _b Q + E2;
  Std
    E1-E2 = eps1-eps2;
  Cov
    E1-E2 = eps3;

  Parameters alpha2 (50.) beta2 gamma2 gamma3 (3*.25);
    alpha2_b = -alpha2 / beta2;
    gamma2_b = -gamma2 / beta2;
    gamma3_b = -gamma3 / beta2;
    _b       = 1 / beta2;

  Bounds
    eps1-eps2 >= 0. ;
run;

```

Output 19.2.1. Food Example of Kmenta

Food example of KMENTA(1971, p.565 & 582)					
The CALIS Procedure					
Covariance Structure Analysis: Pattern and Initial Values					
LINEQS Model Statement					
	Matrix	Rows	Columns	-----Matrix Type-----	
Term 1	1 _SEL_	6	8	SELECTION	
	2 _BETA_	8	8	EQSBETA	IMINUSINV
	3 _GAMMA_	8	6	EQSGAMMA	
	4 _PHI_	6	6	SYMMETRIC	
The 2 Endogenous Variables					
Manifest	Q	P			
Latent					
The 6 Exogenous Variables					
Manifest	D	F	Y	Intercept	
Latent					
Error	E1	E2			

Food example of KMENTA(1971, p.565 & 582)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Parameter Estimates	10
Functions (Observations)	21
Lower Bounds	2
Upper Bounds	0

Optimization Start

Active Constraints	0	Objective Function	2.3500065042
Max Abs Gradient Element	203.9741437	Radius	62167.829174

Iter	Restarts	Function Calls	Active Constraints	Objective Function	Objective Function Change	Max Abs Gradient Element	Lambda	Ratio Between Actual and Predicted Change
1	0	2	0	1.19094	1.1591	3.9410	0	0.688
2	0	5	0	0.32678	0.8642	9.9864	0.00127	2.356
3	0	7	0	0.19108	0.1357	5.5100	0.00006	0.685
4	0	10	0	0.16682	0.0243	2.0513	0.00005	0.867
5	0	12	0	0.16288	0.00393	1.0570	0.00014	0.828
6	0	13	0	0.16132	0.00156	0.3643	0.00004	0.864
7	0	15	0	0.16077	0.000557	0.2176	0.00006	0.984
8	0	16	0	0.16052	0.000250	0.1819	0.00001	0.618
9	0	17	0	0.16032	0.000201	0.0662	0	0.971
10	0	18	0	0.16030	0.000011	0.0195	0	1.108
11	0	19	0	0.16030	6.116E-7	0.00763	0	1.389
12	0	20	0	0.16030	9.454E-8	0.00301	0	1.389
13	0	21	0	0.16030	1.461E-8	0.00118	0	1.388
14	0	22	0	0.16030	2.269E-9	0.000465	0	1.395
15	0	23	0	0.16030	3.59E-10	0.000182	0	1.427

Optimization Results

Iterations	15	Function Calls	24
Jacobian Calls	16	Active Constraints	0
Objective Function	0.1603035477	Max Abs Gradient Element	0.0001820805
Lambda	0	Actual Over Pred Change	1.4266532872
Radius	0.0010322573		

GCONV convergence criterion satisfied.

Food example of KMENTA(1971, p.565 & 582)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Fit Function	0.1603
Goodness of Fit Index (GFI)	0.9530
GFI Adjusted for Degrees of Freedom (AGFI)	0.0120
Root Mean Square Residual (RMR)	2.0653
Parsimonious GFI (Mulaik, 1989)	0.0635
Chi-Square	3.0458
Chi-Square DF	1
Pr > Chi-Square	0.0809
Independence Model Chi-Square	534.27
Independence Model Chi-Square DF	15
RMSEA Estimate	0.3281
RMSEA 90% Lower Confidence Limit	.
RMSEA 90% Upper Confidence Limit	0.7777
ECVI Estimate	1.8270
ECVI 90% Lower Confidence Limit	.
ECVI 90% Upper Confidence Limit	3.3493
Probability of Close Fit	0.0882
Bentler's Comparative Fit Index	0.9961
Normal Theory Reweighted LS Chi-Square	2.8142
Akaike's Information Criterion	1.0458
Bozdogan's (1987) CAIC	-0.9500
Schwarz's Bayesian Criterion	0.0500
McDonald's (1989) Centrality	0.9501
Bentler & Bonett's (1980) Non-normed Index	0.9409
Bentler & Bonett's (1980) NFI	0.9943
James, Mulaik, & Brett (1982) Parsimonious NFI	0.0663
Z-Test of Wilson & Hilferty (1931)	1.4250
Bollen (1986) Normed Index Rho1	0.9145
Bollen (1988) Non-normed Index Delta2	0.9962
Hoelter's (1983) Critical N	25

Food example of KMENTA(1971, p.565 & 582)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Q	=	-0.2295*P	+	0.3100*D	+	93.6193*Intercept	+	1.0000 E1
		beta1		gamma1		alpha1		
P	=	4.2140*Q	+	-0.9305*F	+	-1.5579*Y	+	-218.9*Intercept
		_b		gamma2_b		gamma3_b		alpha2_b
								1.0000 E2

Food example of KMENTA(1971, p.565 & 582)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Variances of Exogenous Variables

Variable	Parameter	Estimate
D		10154
F		9989
Y		151.05263
Intercept		1.05263
E1	eps1	3.51274
E2	eps2	105.06746

Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate
D	F		9994
D	Y		1101
F	Y		1046
D	Intercept		102.66842
F	Intercept		101.71053
Y	Intercept		11.05263
E1	E2	eps3	-18.87270


```

Food example of KMENTA(1971, p.565 & 582)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Q      = -0.2278*P      + 0.3016*D      + 0.9272*Intercept + 0.0181 E1
        beta1          gamma1          alphas1
P      = 4.2467*Q      + -0.9048*F      + -0.1863*Y      + -2.1849*Intercept + 0.0997 E2
        _b            gamma2_b        gamma3_b        alpha2_b

Squared Multiple Correlations

Variable      Error      Total
              Variance   Variance   R-Square
1   Q          3.51274    10730     0.9997
2   P          105.06746   10565     0.9901

Correlations Among Exogenous Variables

Var1      Var2      Parameter      Estimate
D          F          eps3          0.99237
D          Y          eps3          0.88903
F          Y          eps3          0.85184
D          Intercept  eps3          0.99308
F          Intercept  eps3          0.99188
Y          Intercept  eps3          0.87652
E1         E2         eps3          -0.98237

Additional PARMS and Dependent Parameters

The Number of Dependent Parameters is 4

Parameter      Estimate      Standard      t Value
                Error
alpha2          51.94453      .
beta2           0.23731      .
gamma2          0.22082      .
gamma3          0.36971      .
_b             4.21397      .
gamma2_b        -0.93053      .
gamma3_b        -1.55794      .
alpha2_b        -218.89288     .

```

You can obtain almost equivalent results by applying the SAS/ETS procedure SYSLIN on this problem.

Example 19.3. Second-Order Confirmatory Factor Analysis

A second-order confirmatory factor analysis model is applied to a correlation matrix of Thurstone reported by McDonald (1985). Using the LINEQS statement, the three-term second-order factor analysis model is specified in equations notation. The first-order loadings for the three factors, F1, F2, and F3, each refer to three variables, X1-X3, X4-X6, and X7-X9. One second-order factor, F4, reflects the correlations among the three first-order factors. The second-order factor correlation matrix **P** is defined as a 1×1 identity matrix. Choosing the second-order uniqueness matrix **U2** as a diagonal matrix with parameters U21-U23 gives an unidentified model. To compute identified maximum likelihood estimates, the matrix **U2** is defined as a 3×3 identity matrix. The following code generates results that are partially displayed in Output 19.3.1.

```

data Thurst(TYPE=CORR);
  Title "Example of THURSTONE resp. McDONALD (1985, p.57, p.105)";
  _TYPE_ = 'CORR'; Input _NAME_ $ Obs1-Obs9;
  Label Obs1='Sentences' Obs2='Vocabulary' Obs3='Sentence Completion'
        Obs4='First Letters' Obs5='Four-letter Words' Obs6='Suffices'
        Obs7='Letter series' Obs8='Pedigrees' Obs9='Letter Grouping';
  datalines;
Obs1 1. . . . . . . . .
Obs2 .828 1. . . . . . . .
Obs3 .776 .779 1. . . . . . .
Obs4 .439 .493 .460 1. . . . .
Obs5 .432 .464 .425 .674 1. . . .
Obs6 .447 .489 .443 .590 .541 1. . .
Obs7 .447 .432 .401 .381 .402 .288 1. .
Obs8 .541 .537 .534 .350 .367 .320 .555 1. .
Obs9 .380 .358 .359 .424 .446 .325 .598 .452 1.
;

proc calis data=Thurst method=max edf=212 pestim se;
  Title2 "Identified Second Order Confirmatory Factor Analysis";
  Title3 "C = F1 * F2 * P * F2' * F1' + F1 * U2 * F1' + U1, With P=U2=Ide";
  Lineqs
    Obs1 = X1 F1 + E1,
    Obs2 = X2 F1 + E2,
    Obs3 = X3 F1 + E3,
    Obs4 = X4 F2 + E4,
    Obs5 = X5 F2 + E5,
    Obs6 = X6 F2 + E6,
    Obs7 = X7 F3 + E7,
    Obs8 = X8 F3 + E8,
    Obs9 = X9 F3 + E9,
    F1 = X10 F4 + E10,
    F2 = X11 F4 + E11,
    F3 = X12 F4 + E12;
  Std
    F4 = 1. ,
    E1-E9 = U11-U19 ,
    E10-E12 = 3 * 1.;
  Bounds
    0. <= U11-U19;
run;

```

Output 19.3.1. Second-Order Confirmatory Factor Analysis

Example of THURSTONE resp. McDONALD (1985, p.57, p.105)
 Identified Second Order Confirmatory Factor Analysis
 $C = F1 * F2 * P * F2' * F1' + F1 * U2 * F1' + U1$, With $P=U2=Id$

The CALIS Procedure
 Covariance Structure Analysis: Maximum Likelihood Estimation

Parameter Estimates	21
Functions (Observations)	45
Lower Bounds	9
Upper Bounds	0

Optimization Start

Active Constraints	0	Objective Function	0.7151823452
Max Abs Gradient Element	0.4067179803	Radius	2.2578762496

Iter	Restarts	Function Calls	Active Constraints	Objective Function	Objective Function Change	Max Abs Gradient Element	Lambda	Ratio Between Actual and Predicted Change
1	0	2	0	0.23113	0.4840	0.1299	0	1.363
2	0	3	0	0.18322	0.0479	0.0721	0	1.078
3	0	4	0	0.18051	0.00271	0.0200	0	1.006
4	0	5	0	0.18022	0.000289	0.00834	0	1.093
5	0	6	0	0.18018	0.000041	0.00251	0	1.201
6	0	7	0	0.18017	6.523E-6	0.00114	0	1.289
7	0	8	0	0.18017	1.085E-6	0.000388	0	1.347
8	0	9	0	0.18017	1.853E-7	0.000173	0	1.380
9	0	10	0	0.18017	3.208E-8	0.000063	0	1.399
10	0	11	0	0.18017	5.593E-9	0.000028	0	1.408
11	0	12	0	0.18017	9.79E-10	0.000011	0	1.414

Optimization Results

Iterations	11	Function Calls	13
Jacobian Calls	12	Active Constraints	0
Objective Function	0.1801712147	Max Abs Gradient Element	0.0000105805
Lambda	0	Actual Over Pred Change	1.4135857595
Radius	0.0002026368		

GCONV convergence criterion satisfied.

Example of THURSTONE resp. McDONALD (1985, p.57, p.105)
 Identified Second Order Confirmatory Factor Analysis
 $C = F1 * F2 * P * F2' * F1' + F1 * U2 * F1' + U1$, With $P=U2=I_{de}$

The CALIS Procedure
 Covariance Structure Analysis: Maximum Likelihood Estimation

Fit Function	0.1802
Goodness of Fit Index (GFI)	0.9596
GFI Adjusted for Degrees of Freedom (AGFI)	0.9242
Root Mean Square Residual (RMR)	0.0436
Parsimonious GFI (Mulaik, 1989)	0.6397
Chi-Square	38.1963
Chi-Square DF	24
Pr > Chi-Square	0.0331
Independence Model Chi-Square	1101.9
Independence Model Chi-Square DF	36
RMSEA Estimate	0.0528
RMSEA 90% Lower Confidence Limit	0.0153
RMSEA 90% Upper Confidence Limit	0.0831
ECVI Estimate	0.3881
ECVI 90% Lower Confidence Limit	.
ECVI 90% Upper Confidence Limit	0.4888
Probability of Close Fit	0.4088
Bentler's Comparative Fit Index	0.9867
Normal Theory Reweighted LS Chi-Square	40.1947
Akaike's Information Criterion	-9.8037
Bozdogan's (1987) CAIC	-114.4747
Schwarz's Bayesian Criterion	-90.4747
McDonald's (1989) Centrality	0.9672
Bentler & Bonett's (1980) Non-normed Index	0.9800
Bentler & Bonett's (1980) NFI	0.9653
James, Mulaik, & Brett (1982) Parsimonious NFI	0.6436
Z-Test of Wilson & Hilferty (1931)	1.8373
Bollen (1986) Normed Index Rho1	0.9480
Bollen (1988) Non-normed Index Delta2	0.9868
Hoelter's (1983) Critical N	204

Example of THURSTONE resp. McDONALD (1985, p.57, p.105)
 Identified Second Order Confirmatory Factor Analysis
 $C = F1 * F2 * P * F2' * F1' + F1 * U2 * F1' + U1$, With $P=U2=I_{de}$

The CALIS Procedure
 Covariance Structure Analysis: Maximum Likelihood Estimation

Obs1	=	0.5151*F1	+	1.0000 E1
Std Err		0.0629 X1		
t Value		8.1868		
Obs2	=	0.5203*F1	+	1.0000 E2
Std Err		0.0634 X2		
t Value		8.2090		
Obs3	=	0.4874*F1	+	1.0000 E3
Std Err		0.0608 X3		
t Value		8.0151		
Obs4	=	0.5211*F2	+	1.0000 E4
Std Err		0.0611 X4		
t Value		8.5342		
Obs5	=	0.4971*F2	+	1.0000 E5
Std Err		0.0590 X5		
t Value		8.4213		
Obs6	=	0.4381*F2	+	1.0000 E6
Std Err		0.0560 X6		
t Value		7.8283		
Obs7	=	0.4524*F3	+	1.0000 E7
Std Err		0.0660 X7		
t Value		6.8584		
Obs8	=	0.4173*F3	+	1.0000 E8
Std Err		0.0622 X8		
t Value		6.7135		
Obs9	=	0.4076*F3	+	1.0000 E9
Std Err		0.0613 X9		
t Value		6.6484		

Example of THURSTONE resp. McDONALD (1985, p.57, p.105)
 Identified Second Order Confirmatory Factor Analysis
 $C = F1 * F2 * P * F2' * F1' + F1 * U2 * F1' + U1$, With $P=U2=I_{de}$

The CALIS Procedure
 Covariance Structure Analysis: Maximum Likelihood Estimation

F1	=	1.4438*F4	+	1.0000 E10
Std Err		0.2565 X10		
t Value		5.6282		
F2	=	1.2538*F4	+	1.0000 E11
Std Err		0.2114 X11		
t Value		5.9320		
F3	=	1.4065*F4	+	1.0000 E12
Std Err		0.2689 X12		
t Value		5.2307		

Example of THURSTONE resp. McDONALD (1985, p.57, p.105)
 Identified Second Order Confirmatory Factor Analysis
 $C = F1 * F2 * P * F2' * F1' + F1 * U2 * F1' + U1$, With $P=U2=I_{de}$

The CALIS Procedure
 Covariance Structure Analysis: Maximum Likelihood Estimation

Variances of Exogenous Variables

Variable	Parameter	Estimate	Standard Error	t Value
F4		1.00000		
E1	U11	0.18150	0.02848	6.37
E2	U12	0.16493	0.02777	5.94
E3	U13	0.26713	0.03336	8.01
E4	U14	0.30150	0.05102	5.91
E5	U15	0.36450	0.05264	6.93
E6	U16	0.50642	0.05963	8.49
E7	U17	0.39032	0.05934	6.58
E8	U18	0.48138	0.06225	7.73
E9	U19	0.50509	0.06333	7.98
E10		1.00000		
E11		1.00000		
E12		1.00000		

Example of THURSTONE resp. McDONALD (1985, p.57, p.105)
 Identified Second Order Confirmatory Factor Analysis
 $C = F1 * F2 * P * F2' * F1' + F1 * U2 * F1' + U1$, With $P=U2=I_{de}$

The CALIS Procedure
 Covariance Structure Analysis: Maximum Likelihood Estimation

Obs1	=	0.9047*	F1	+	0.4260	E1
			X1			
Obs2	=	0.9138*	F1	+	0.4061	E2
			X2			
Obs3	=	0.8561*	F1	+	0.5168	E3
			X3			
Obs4	=	0.8358*	F2	+	0.5491	E4
			X4			
Obs5	=	0.7972*	F2	+	0.6037	E5
			X5			
Obs6	=	0.7026*	F2	+	0.7116	E6
			X6			
Obs7	=	0.7808*	F3	+	0.6248	E7
			X7			
Obs8	=	0.7202*	F3	+	0.6938	E8
			X8			
Obs9	=	0.7035*	F3	+	0.7107	E9
			X9			

Example of THURSTONE resp. McDONALD (1985, p.57, p.105)
 Identified Second Order Confirmatory Factor Analysis
 $C = F1 * F2 * P * F2' * F1' + F1 * U2 * F1' + U1$, With $P=U2=I_{de}$

The CALIS Procedure
 Covariance Structure Analysis: Maximum Likelihood Estimation

F1	=	0.8221*	F4	+	0.5694	E10
			X10			
F2	=	0.7818*	F4	+	0.6235	E11
			X11			
F3	=	0.8150*	F4	+	0.5794	E12
			X12			

Example of THURSTONE resp. McDONALD (1985, p.57, p.105)
 Identified Second Order Confirmatory Factor Analysis
 $C = F1 * F2 * P * F2' * F1' + F1 * U2 * F1' + U1$, With $P=U2=Ide$

The CALIS Procedure
 Covariance Structure Analysis: Maximum Likelihood Estimation

Squared Multiple Correlations

	Variable	Error Variance	Total Variance	R-Square
1	Obs1	0.18150	1.00000	0.8185
2	Obs2	0.16493	1.00000	0.8351
3	Obs3	0.26713	1.00000	0.7329
4	Obs4	0.30150	1.00000	0.6985
5	Obs5	0.36450	1.00000	0.6355
6	Obs6	0.50642	1.00000	0.4936
7	Obs7	0.39032	1.00000	0.6097
8	Obs8	0.48138	1.00000	0.5186
9	Obs9	0.50509	1.00000	0.4949
10	F1	1.00000	3.08452	0.6758
11	F2	1.00000	2.57213	0.6112
12	F3	1.00000	2.97832	0.6642

To compute McDonald's unidentified model, you would have to change the STD and BOUNDS statements to include three more parameters:

```
Std
  F4      = 1. ,
  E1-E9   = U11-U19 ,
  E10-E12 = U21-U23 ;
Bounds
  0. <= U11-U19,
  0. <= U21-U23;
```

The unidentified model is indicated in the output by an analysis of the linear dependencies in the approximate Hessian matrix (not shown). Because the information matrix is singular, standard errors are computed based on a Moore-Penrose inverse. The results computed by PROC CALIS differ from those reported by McDonald (1985). In the case of an unidentified model, the parameter estimates are not unique.

To specify the identified model using the COSAN model statement, you can use the following statements:

```
Title2 "Identified Second Order Confirmatory Factor Analysis Using COSAN";
Title3 "C = F1*F2*P*F2'*F1' + F1*U2*F1' + U1, With P=U2=Ide";
proc calis data=Thurst method=max edf=212 pestim se;
  Cosan F1(3) * F2(1) * P(1,Ide) + F1(3) * U2(3,Ide) + U1(9,Dia);
  Matrix F1
    [ ,1] = X1-X3,
    [ ,2] = 3 * 0. X4-X6,
    [ ,3] = 6 * 0. X7-X9;
  Matrix F2
    [ ,1] = X10-X12;
```



```

Matrix U1
      [1,1] = U11-U19;
Bounds
      0. <= U11-U19;
run;

```

Because PROC CALIS cannot compute initial estimates for a model specified by the general COSAN statement, this analysis may require more iterations than one using the LINEQS statement, depending on the precision of the processor.

Example 19.4. Linear Relations Among Factor Loadings

The correlation matrix from Kinzer and Kinzer (N=326) is used by Guttman (1957) as an example that yields an approximate simplex. McDonald (1980) uses this data set as an example of factor analysis where he supposes that the loadings of the second factor are a linear function of the loadings on the first factor, for example

$$b_{j2} = \alpha + \beta b_{j1}, \quad j = 1, \dots, n$$

This example is also discussed in Browne (1982). The matrix specification of the model is

$$\mathbf{C} = \mathbf{F}_1 \mathbf{F}_1'$$

with

$$F_1 = \begin{pmatrix} b_{11} & \alpha + \beta b_{11} & & & & & \\ b_{21} & \alpha + \beta b_{21} & & & & & \\ b_{31} & \alpha + \beta b_{31} & & & & & \\ b_{41} & \alpha + \beta b_{41} & & & & & \\ b_{51} & \alpha + \beta b_{51} & & & & & \\ b_{61} & \alpha + \beta b_{61} & & & & & \end{pmatrix} \begin{matrix} u_{11} \\ u_{22} \\ u_{33} \\ u_{44} \\ u_{55} \\ u_{66} \end{matrix}$$

This example is recomputed by PROC CALIS to illustrate a simple application of the COSAN model statement combined with program statements. This example also serves to illustrate the identification problem.

```

data Kinzer(TYPE=CORR);
  Title "Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)";
  _TYPE_ = 'CORR'; INPUT _NAME_ $ Obs1-Obs6;
  datalines;
Obs1  1.00  .    .    .    .    .
Obs2  .51  1.00  .    .    .    .
Obs3  .46  .51  1.00  .    .    .
Obs4  .46  .47  .54  1.00  .    .
Obs5  .40  .39  .49  .57  1.00  .
Obs6  .33  .39  .47  .45  .56  1.00
;

```

In a first test run of PROC CALIS, the same model is used as reported in McDonald (1980). Using the Levenberg-Marquardt optimization algorithm, this example specifies maximum likelihood estimation in the following code:

```
proc calis data=Kinzer method=max outram=ram nobs=326;
  Title2 "Linearly Related Factor Analysis, (McDonald,1980)";
  Title3 "Identification Problem";
  Cosan F(8,Gen) * I(8,Ide);
  Matrix F
    [ ,1]= X1-X6,
    [ ,2]= X7-X12,
    [1,3]= X13-X18;
  Parm Alfa = .5 Beta = -.5;
  X7 = Alfa + Beta * X1;
  X8 = Alfa + Beta * X2;
  X9 = Alfa + Beta * X3;
  X10 = Alfa + Beta * X4;
  X11 = Alfa + Beta * X5;
  X12 = Alfa + Beta * X6;
  Bounds X13-X18 >= 0.;
  Vnames F Fact1 Fact2 Uvar1-Uvar6;
run;
```

The pattern of the initial values is displayed in vector and in matrix form. You should always read this output very carefully, particularly when you use your own programming statements to constrain the matrix elements. The vector form shows the mapping of the model parameters to indices of the vector X that is optimized. The matrix form indicates parameter elements that are constrained by program statements by indices of X in angle brackets ($< >$). An asterisk trailing the iteration number in the displayed optimization history of the Levenberg-Marquardt algorithm indicates that the optimization process encountered a singular Hessian matrix. When this happens, especially in the last iterations, the model may not be properly identified. The computed χ^2 value of 10.337 for 7 degrees of freedom and the computed unique loadings agree with those reported by McDonald (1980), but the maximum likelihood estimates for the common factor loadings differ to some degree. The common factor loadings can be subjected to transformations that do not increase the value of the optimization criterion because the problem is not identified. An estimation problem that is not fully identified can lead to different solutions caused only by different initial values, different optimization techniques, or computers with different machine precision or floating-point arithmetic.

To overcome the identification problem in the first model, restart PROC CALIS with a simple modification to the model in which the former parameter $X1$ is fixed to 0. This leads to 8 instead of 7 degrees of freedom. The following code produces results that are partially displayed in Output 19.4.1.

```

data ram2(TYPE=RAM); set ram;
  if _type_ = 'ESTIM' then
    if _name_ = 'X1' then do;
      _name_ = ' '; _estim_ = 0.;
    end;
run;

proc calis data=Kinzer method=max inram=ram2 nobs=326;
  Title2 "Linearly Related Factor Analysis, (Mcdonald,1980)";
  Title3 "Identified Model";
  Parms Alfa = .5 Beta = -.5;
  X7 = Alfa;
  X8 = Alfa + Beta * X2;
  X9 = Alfa + Beta * X3;
  X10 = Alfa + Beta * X4;
  X11 = Alfa + Beta * X5;
  X12 = Alfa + Beta * X6;
  Bounds X13-X18 >= 0.;
run;

```

Output 19.4.1. Linearly Related Factor Analysis: Identification Problem

```

Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)
Linearly Related Factor Analysis, (Mcdonald,1980)
Identified Model

The CALIS Procedure
Covariance Structure Analysis: Pattern and Initial Values

COSAN Model Statement

Matrix      Rows      Columns      -----Matrix Type-----
Term 1      1      F      6      8      GENERAL
            2      I      8      8      IDENTITY

Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)
Linearly Related Factor Analysis, (Mcdonald,1980)
Identified Model

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Parameter Estimates      13
Functions (Observations) 21
Lower Bounds             6
Upper Bounds             0

Optimization Start

Active Constraints      0      Objective Function      0.3234289189
Max Abs Gradient Element      2.2633860283      Radius      5.8468569273

```

Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)
 Linearly Related Factor Analysis, (Mcdonald,1980)
 Identified Model

The CALIS Procedure
 Covariance Structure Analysis: Maximum Likelihood Estimation

Iter	Restarts	Function Calls	Active Constraints	Objective Function	Objective Function Change	Max Abs Gradient Element	Lambda	Ratio Between Actual and Predicted Change
1	0	2	0	0.07994	0.2435	0.3984	0	0.557
2	0	3	0	0.03334	0.0466	0.0672	0	1.202
3	0	4	0	0.03185	0.00150	0.00439	0	1.058
4	0	5	0	0.03181	0.000034	0.00236	0	0.811
5	0	6	0	0.03181	3.982E-6	0.000775	0	0.591
6	0	7	0	0.03181	9.275E-7	0.000490	0	0.543
7	0	8	0	0.03181	2.402E-7	0.000206	0	0.526
8	0	9	0	0.03181	6.336E-8	0.000129	0	0.514
9	0	10	0	0.03181	1.687E-8	0.000054	0	0.505
10	0	11	0	0.03181	4.521E-9	0.000034	0	0.498
11	0	12	0	0.03181	1.217E-9	0.000014	0	0.493
12	0	13	0	0.03181	3.29E-10	8.971E-6	0	0.489

Optimization Results

Iterations	12	Function Calls	14
Jacobian Calls	13	Active Constraints	0
Objective Function	0.0318073951	Max Abs Gradient Element	8.9711916E-6
Lambda	0	Actual Over Pred Change	0.4888109559
Radius	0.0002016088		

ABSGCONV convergence criterion satisfied.

Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)
 Linearly Related Factor Analysis, (McDonald, 1980)
 Identified Model

The CALIS Procedure
 Covariance Structure Analysis: Maximum Likelihood Estimation

Fit Function	0.0318
Goodness of Fit Index (GFI)	0.9897
GFI Adjusted for Degrees of Freedom (AGFI)	0.9730
Root Mean Square Residual (RMR)	0.0409
Parsimonious GFI (Mulaik, 1989)	0.5278
Chi-Square	10.3374
Chi-Square DF	8
Pr > Chi-Square	0.2421
Independence Model Chi-Square	682.87
Independence Model Chi-Square DF	15
RMSEA Estimate	0.0300
RMSEA 90% Lower Confidence Limit	.
RMSEA 90% Upper Confidence Limit	0.0756
ECVI Estimate	0.1136
ECVI 90% Lower Confidence Limit	.
ECVI 90% Upper Confidence Limit	0.1525
Probability of Close Fit	0.7137
Bentler's Comparative Fit Index	0.9965
Normal Theory Reweighted LS Chi-Square	10.1441
Akaike's Information Criterion	-5.6626
Bozdogan's (1987) CAIC	-43.9578
Schwarz's Bayesian Criterion	-35.9578
McDonald's (1989) Centrality	0.9964
Bentler & Bonett's (1980) Non-normed Index	0.9934
Bentler & Bonett's (1980) NFI	0.9849
James, Mulaik, & Brett (1982) Parsimonious NFI	0.5253
Z-Test of Wilson & Hilferty (1931)	0.7019
Bollen (1986) Normed Index Rho1	0.9716
Bollen (1988) Non-normed Index Delta2	0.9965
Hoelter's (1983) Critical N	489

Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)
 Linearly Related Factor Analysis, (Mcdonald,1980)
 Identified Model

The CALIS Procedure
 Covariance Structure Analysis: Maximum Likelihood Estimation

Estimated Parameter Matrix F[6:8]
 Standard Errors and t Values
 General Matrix

	Fact1	Fact2	Uvar1	Uvar2	Uvar3	Uvar4	Uvar5	Uvar6
Obs1	0	0.7151	0.7283	0	0	0	0	0
	0	0.0405	0.0408	0	0	0	0	0
	0	17.6382	17.8276	0	0	0	0	0
		<X7>	[X13]					
Obs2	-0.0543	0.7294	0	0.6707	0	0	0	0
	0.1042	0.0438	0	0.0472	0	0	0	0
	-0.5215	16.6655	0	14.2059	0	0	0	0
	[X2]	<X8>		[X14]				
Obs3	0.1710	0.6703	0	0	0.6983	0	0	0
	0.0845	0.0396	0	0	0.0324	0	0	0
	2.0249	16.9077	0	0	21.5473	0	0	0
	[X3]	<X9>			[X15]			
Obs4	0.2922	0.6385	0	0	0	0.6876	0	0
	0.0829	0.0462	0	0	0	0.0319	0	0
	3.5224	13.8352	0	0	0	21.5791	0	0
	[X4]	<X10>				[X16]		
Obs5	0.5987	0.5582	0	0	0	0	0.5579	0
	0.1003	0.0730	0	0	0	0	0.0798	0
	5.9665	7.6504	0	0	0	0	6.9938	0
	[X5]	<X11>					[X17]	
Obs6	0.4278	0.6029	0	0	0	0	0	0.7336
	0.0913	0.0586	0	0	0	0	0	0.0400
	4.6844	10.2928	0	0	0	0	0	18.3580
	[X6]	<X12>						[X18]

Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)
 Linearly Related Factor Analysis, (Mcdonald,1980)
 Identified Model

The CALIS Procedure
 Covariance Structure Analysis: Maximum Likelihood Estimation

Additional PARMS and Dependent Parameters

The Number of Dependent Parameters is 6

Parameter	Estimate	Standard Error	t Value
Alfa	0.71511	0.04054	17.64
Beta	-0.26217	0.12966	-2.02
X7	0.71511	0.04054	17.64
X8	0.72936	0.04376	16.67
X9	0.67027	0.03964	16.91
X10	0.63851	0.04615	13.84
X11	0.55815	0.07296	7.65
X12	0.60295	0.05858	10.29

The lambda value of the iteration history indicates that Newton steps can always be performed. Because no singular Hessian matrices (which can slow down the convergence rate considerably) are computed, this example needs just 12 iterations com-

pared to the 17 needed in the previous example. Note that the number of iterations may be machine-dependent. The value of the fit function, the residuals, and the χ^2 value agree with the values obtained in fitting the first model. This indicates that this second model is better identified than the first one. It is fully identified, as indicated by the fact that the Hessian matrix is nonsingular.

Example 19.5. Ordinal Relations Among Factor Loadings

McDonald (1980) uses the same data set to compute a factor analysis with ordinally constrained factor loadings. The results of the linearly constrained factor analysis show that the loadings of the two factors are ordered as 2, 1, 3, 4, 6, 5. McDonald (1980) then tests the hypothesis that the factor loadings are all nonnegative and can be ordered in the following manner:

$$b_{11} \leq b_{21} \leq b_{31} \leq b_{41} \leq b_{51} \leq b_{61}$$

$$b_{12} \geq b_{22} \geq b_{32} \geq b_{42} \geq b_{52} \geq b_{62}$$

This example is recomputed by PROC CALIS to illustrate a further application of the COSAN model statement combined with program statements. The same identification problem as in Example 19.4 on page 611 occurs here. The following model specification describes an unidentified model:

```
proc calis data=Kinzer method=max outram=ram tech=nr nobs=326;
Title2 "Ordinally Related Factor Analysis, (Mcdonald,1980)";
Title3 "Identification Problem";
Cosan F(8,Gen) * I(8,Ide);
  MATRIX F
    [,1] = x1-x6,
    [,2] = x7-x12,
    [1,3] = x13-x18;
  PARAMETERS t1-t10=1.;
    x2 = x1 + t1 * t1;
    x3 = x2 + t2 * t2;
    x4 = x3 + t3 * t3;
    x5 = x4 + t4 * t4;
    x6 = x5 + t5 * t5;
    x11 = x12 + t6 * t6;
    x10 = x11 + t7 * t7;
    x9 = x10 + t8 * t8;
    x8 = x9 + t9 * t9;
    x7 = x8 + t10 * t10;
  Bounds x13-x18 >= 0.;
  Vnames F Fact1 Fact2 Uvar1-Uvar6;
run;
```

You can specify the same model with the LINCON statement:

```
proc calis data=Kinzer method=max tech=lm edf=325;
  Title3 "Identified Problem 2";
  cosan f(8,gen)*I(8,ide);
  matrix F
    [,1] = x1-x6,
    [,2] = x7-x12,
    [1,3] = x13-x18;
  lincon x1 <= x2,
        x2 <= x3,
        x3 <= x4,
        x4 <= x5,
        x5 <= x6,
        x7 >= x8,
        x8 >= x9,
        x9 >= x10,
        x10 >= x11,
        x11 >= x12;
  Bounds x13-x18 >= 0.;
  Vnames F Fact1 Fact2 Uvar1-Uvar6;
run;
```

To have an identified model, the loading, b_{11} (x_1), is fixed at 0. The information in the OUTRAM= data set (the data set *ram*), produced by the unidentified model, can be used to specify the identified model. However, because x_1 is now a fixed constant in the identified model, it should not have a parameter name in the new analysis. Thus, the data set *ram* is modified as follows:

```
data ram2(type=ram); set ram;
  if _name_ = 'x1' then do;
    _name_ = ' '; _estim_ = 0.;
  end;
run;
```

The data set *ram2* is now an OUTRAM= data set in which x_1 is no longer a parameter. PROC CALIS reads the information (that is, the set of parameters and the model specification) in the data set *ram2* for the identified model. As displayed in the following code, you can use the PARMS statement to specify the desired ordinal relationships between the parameters.

```
proc calis data=Kinzer method=max inram=ram2 tech=nr nobs=326;
  title2 "Ordinally Related Factor Analysis, (Mcdonald,1980)";
  title3 "Identified Model with X1=0";
  parms t1-t10= 10 * 1.;
  x2 = + t1 * t1;
  x3 = x2 + t2 * t2;
  x4 = x3 + t3 * t3;
  x5 = x4 + t4 * t4;
  x6 = x5 + t5 * t5;
```



```

x11 = x12 + t6 * t6;
x10 = x11 + t7 * t7;
x9  = x10 + t8 * t8;
x8  = x9  + t9 * t9;
x7  = x8  + t10 * t10;
bounds x13-x18 >= 0.;
run;

```

Selected output for the identified model is displayed in Output 19.5.1.

Output 19.5.1. Factor Analysis with Ordinal Constraints

Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)									
Ordinally Related Factor Analysis, (Mcdonald,1980)									
Identified Model with X1=0									
The CALIS Procedure									
Covariance Structure Analysis: Maximum Likelihood Estimation									
Parameter Estimates					17				
Functions (Observations)					21				
Lower Bounds					6				
Upper Bounds					0				
Optimization Start									
Active Constraints					0		Objective Function		5.2552270182
Max Abs Gradient Element					0.8821788922				

Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)
 Ordinarily Related Factor Analysis, (McDonald, 1980)
 Identified Model with X1=0

The CALIS Procedure
 Covariance Structure Analysis: Maximum Likelihood Estimation

Fit Function	0.0256
Goodness of Fit Index (GFI)	0.9916
GFI Adjusted for Degrees of Freedom (AGFI)	0.9557
Root Mean Square Residual (RMR)	0.0180
Parsimonious GFI (Mulaik, 1989)	0.2644
Chi-Square	8.3158
Chi-Square DF	4
Pr > Chi-Square	0.0807
Independence Model Chi-Square	682.87
Independence Model Chi-Square DF	15
RMSEA Estimate	0.0576
RMSEA 90% Lower Confidence Limit	.
RMSEA 90% Upper Confidence Limit	0.1133
ECVI Estimate	0.1325
ECVI 90% Lower Confidence Limit	.
ECVI 90% Upper Confidence Limit	0.1711
Probability of Close Fit	0.3399
Bentler's Comparative Fit Index	0.9935
Normal Theory Reweighted LS Chi-Square	8.2901
Akaike's Information Criterion	0.3158
Bozdogan's (1987) CAIC	-18.8318
Schwarz's Bayesian Criterion	-14.8318
McDonald's (1989) Centrality	0.9934
Bentler & Bonett's (1980) Non-normed Index	0.9758
Bentler & Bonett's (1980) NFI	0.9878
James, Mulaik, & Brett (1982) Parsimonious NFI	0.2634
Z-Test of Wilson & Hilferty (1931)	1.4079
Bollen (1986) Normed Index Rho1	0.9543
Bollen (1988) Non-normed Index Delta2	0.9936
Hoelter's (1983) Critical N	372

Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)
 Ordinarily Related Factor Analysis, (McDonald,1980)
 Identified Model with X1=0

The CALIS Procedure
 Covariance Structure Analysis: Maximum Likelihood Estimation

Estimated Parameter Matrix F[6:8]
 Standard Errors and t Values
 General Matrix

	Fact1	Fact2	Uvar1	Uvar2	Uvar3	Uvar4	Uvar5	Uvar6
Obs1	0	0.7101	0.7131	0	0	0	0	0
	0	0.0435	0.0404	0	0	0	0	0
	0	16.3317	17.6427	0	0	0	0	0
		<x7>	[x13]					
Obs2	0.0261	0.7101	0	0.6950	0	0	0	0
	0.0875	0.0435	0	0.0391	0	0	0	0
	0.2977	16.3317	0	17.7571	0	0	0	0
	<x2>	<x8>		[x14]				
Obs3	0.2382	0.6827	0	0	0.6907	0	0	0
	0.0851	0.0604	0	0	0.0338	0	0	0
	2.7998	11.3110	0	0	20.4239	0	0	0
	<x3>	<x9>			[x15]			
Obs4	0.3252	0.6580	0	0	0	0.6790	0	0
	0.0823	0.0621	0	0	0	0.0331	0	0
	3.9504	10.5950	0	0	0	20.5361	0	0
	<x4>	<x10>				[x16]		
Obs5	0.5395	0.5528	0	0	0	0	0.6249	0
	0.0901	0.0705	0	0	0	0	0.0534	0
	5.9887	7.8359	0	0	0	0	11.7052	0
	<x5>	<x11>					[x17]	
Obs6	0.5395	0.4834	0	0	0	0	0	0.7005
	0.0918	0.0726	0	0	0	0	0	0.0524
	5.8776	6.6560	0	0	0	0	0	13.3749
	<x6>	[x12]						[x18]

Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)
 Ordinarily Related Factor Analysis, (McDonald,1980)
 Identified Model with X1=0

The CALIS Procedure
 Covariance Structure Analysis: Maximum Likelihood Estimation

Additional PARMS and Dependent Parameters

The Number of Dependent Parameters is 10

Parameter	Estimate	Standard Error	t Value
t1	0.16143	0.27111	0.60
t2	0.46060	0.09289	4.96
t3	0.29496	0.13702	2.15
t4	0.46297	0.10756	4.30
t5	0.0000522	1311	0.00
t6	0.26347	0.12203	2.16
t7	0.32430	0.09965	3.25
t8	0.15721	0.21134	0.74
t9	0.16543	0.20537	0.81
t10	-4.2528E-7	0.47736	-0.00
x7	0.71007	0.04348	16.33
x2	0.02606	0.08753	0.30
x8	0.71007	0.04348	16.33
x3	0.23821	0.08508	2.80
x9	0.68270	0.06036	11.31
x4	0.32521	0.08232	3.95
x10	0.65799	0.06210	10.60
x5	0.53955	0.09009	5.99
x11	0.55282	0.07055	7.84
x6	0.53955	0.09180	5.88

By fixing the loading b_{11} (x1) to constant 0, you obtain $\chi^2 = 8.316$ on $df = 4$ ($p < .09$). McDonald reports the same χ^2 value, but on $df = 3$, and thus, he obtains a smaller p -value. An analysis without the fixed loading shows typical signs of an unidentified problem: after more iterations it leads to a parameter set with a χ^2 value of 8.174 on $df = 3$. A singular Hessian matrix occurs.

The singular Hessian matrix of the unidentified problem slows down the convergence rate of the Levenberg-Marquardt algorithm considerably. Compared to the unidentified problem with 30 iterations, the identified problem needs only 20 iterations. Note that the number of iterations may depend on the precision of the processor.

The same model can also be specified using the LINCON statement for linear constraints:

```
proc calis data=Kinzer method=max tech=lm edf=325;
  Title3 "Identified Model 2";
  cosan f(8,gen)*I(8,ide);
  matrix f
    [,1] = 0. x2-x6,
    [,2] = x7-x12,
    [1,3] = x13-x18;
```

```
lincon  x2  <= x3,
        x3  <= x4,
        x4  <= x5,
        x5  <= x6,
        x7  >= x8,
        x8  >= x9,
        x9  >= x10,
        x10 >= x11,
        x11 >= x12;
bounds x2 x13-x18 >= 0.;
run;
```

Example 19.6. Longitudinal Factor Analysis

The following example (McDonald 1980) illustrates both the ability of PROC CALIS to formulate complex covariance structure analysis problems by the generalized COSAN matrix model and the use of program statements to impose nonlinear constraints on the parameters. The example is a longitudinal factor analysis using the Swaminathan (1974) model. For $m = 3$ tests, $k = 3$ occasions, and $r = 2$ factors the matrix model is formulated in the section “First-Order Autoregressive Longitudinal Factor Model” on page 443 as follows:

$$\mathbf{C} = \mathbf{F}_1 \mathbf{F}_2 \mathbf{F}_3 \mathbf{L} \mathbf{F}_3^{-1} \mathbf{F}_2^{-1} \mathbf{P} (\mathbf{F}_2^{-1})' (\mathbf{F}_3^{-1})' \mathbf{L}' \mathbf{F}_3' \mathbf{F}_2' \mathbf{F}_1' + \mathbf{U}^2$$

$$F_1 = \begin{pmatrix} B_1 & & \\ & B_2 & \\ & & B_3 \end{pmatrix}, \quad F_2 = \begin{pmatrix} I_2 & & \\ & D_2 & \\ & & D_2 \end{pmatrix}, \quad F_3 = \begin{pmatrix} I_2 & & \\ & I_2 & \\ & & D_3 \end{pmatrix}$$

$$L = \begin{pmatrix} I_2 & o & o \\ I_2 & I_2 & o \\ I_2 & I_2 & I_2 \end{pmatrix}, \quad P = \begin{pmatrix} I_2 & & \\ & S_2 & \\ & & S_3 \end{pmatrix}, \quad U = \begin{pmatrix} U_{11} & U_{12} & U_{13} \\ U_{21} & U_{22} & U_{23} \\ U_{31} & U_{32} & U_{33} \end{pmatrix}$$

$$S_2 = I_2 - D_2^2, \quad S_3 = I_2 - D_3^2$$

The Swaminathan longitudinal factor model assumes that the factor scores for each (m) common factor change from occasion to occasion (k) according to a first-order autoregressive scheme. The matrix \mathbf{F}_1 contains the k factor loading matrices $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3$ (each is $n \times m$). The matrices $\mathbf{D}_2, \mathbf{D}_3, \mathbf{S}_2, \mathbf{S}_3$ and $\mathbf{U}_{ij}, i, j = 1, \dots, k$, are diagonal, and the matrices \mathbf{D}_i and $\mathbf{S}_i, i = 2, \dots, k$, are subjected to the constraint

$$\mathbf{S}_i + \mathbf{D}_i^2 = \mathbf{I}$$

Since the constructed correlation matrix given in McDonald's (1980) paper is singular, only unweighted least-squares estimates can be computed.

```

data Mcdon(TYPE=CORR);
Title "Swaminathan's Longitudinal Factor Model, Data: McDONALD(1980)";
Title2 "Constructed Singular Correlation Matrix, GLS & ML not possible";
  _TYPE_ = 'CORR'; INPUT _NAME_ $ Obs1-Obs9;
  datalines;
Obs1  1.000  .      .      .      .      .      .      .      .
Obs2  .100  1.000  .      .      .      .      .      .      .
Obs3  .250  .400  1.000  .      .      .      .      .      .
Obs4  .720  .108  .270  1.000  .      .      .      .      .
Obs5  .135  .740  .380  .180  1.000  .      .      .      .
Obs6  .270  .318  .800  .360  .530  1.000  .      .      .
Obs7  .650  .054  .135  .730  .090  .180  1.000  .      .
Obs8  .108  .690  .196  .144  .700  .269  .200  1.000  .
Obs9  .189  .202  .710  .252  .336  .760  .350  .580  1.000
;

proc calis data=Mcdon method=ls tech=nr nobs=100;
cosan B(6,Gen) * D1(6,Dia) * D2(6,Dia) * T(6,Low) * D3(6,Dia,Inv) *
      D4(6,Dia,Inv) * P(6,Dia) + U(9,Sym);

  Matrix B
    [ ,1]= X1-X3,
    [ ,2]= 0. X4-X5,
    [ ,3]= 3 * 0. X6-X8,
    [ ,4]= 4 * 0. X9-X10,
    [ ,5]= 6 * 0. X11-X13,
    [ ,6]= 7 * 0. X14-X15;

  Matrix D1
    [1,1]= 2 * 1. X16 X17 X16 X17;

  Matrix D2
    [1,1]= 4 * 1. X18 X19;

  Matrix T
    [1,1]= 6 * 1.,
    [3,1]= 4 * 1.,
    [5,1]= 2 * 1.;

  Matrix D3
    [1,1]= 4 * 1. X18 X19;

  Matrix D4
    [1,1]= 2 * 1. X16 X17 X16 X17;

  Matrix P
    [1,1]= 2 * 1. X20-X23;

  Matrix U
    [1,1]= X24-X32,
    [4,1]= X33-X38,
    [7,1]= X39-X41;

  Bounds 0. <= X24-X32,
    -1. <= X16-X19 <= 1.;
  X20 = 1. - X16 * X16;
  X21 = 1. - X17 * X17;
  X22 = 1. - X18 * X18;
  X23 = 1. - X19 * X19;
run;

```

Because this formulation of Swaminathan's model in general leads to an unidentified problem, the results given here are different from those reported by McDonald (1980). The displayed output of PROC CALIS also indicates that the fitted central model matrices \mathbf{P} and \mathbf{U} are not positive definite. The BOUNDS statement constrains the diagonals of the matrices \mathbf{P} and \mathbf{U} to be nonnegative, but this cannot prevent \mathbf{U} from having three negative eigenvalues. The fact that many of the published results for more complex models in covariance structure analysis are connected to unidentified problems implies that more theoretical work should be done to study the general features of such models.

References

- Akaike, H. (1974), "A New Look at the Statistical Identification Model," *IEEE Transactions on Automatic Control*, 19, 716–723.
- Akaike, H. (1987), "Factor Analysis and AIC," *Psychometrika*, 52, 317–332.
- Al-Baali, M. and Fletcher, R. (1985), "Variational Methods for Nonlinear Least Squares," *J. Oper. Res. Soc.*, 36, 405–421.
- Al-Baali, M. and Fletcher, R. (1986), "An Efficient Line Search for Nonlinear Least Squares," *J. Optimiz. Theory Appl.*, 48, 359–377.
- Anderson, T.W. (1960), "Some Stochastic Process Models for Intelligence Test Scores," in *Mathematical Methods in the Social Sciences*, eds. K.J. Arrow, S. Karlin, and P. Suppes Stanford: Stanford University Press.
- Beale, E.M.L. (1972), "A Derivation of Conjugate Gradients," in *Numerical Methods for Nonlinear Optimization*, ed. F.A. Lootsma, London: Academic Press.
- Bentler, P.M. (1983), "Some Contributions to Efficient Statistics in Structural Models: Specification and Estimation of Moment Structures," *Psychometrika*, 48, 493–517.
- Bentler, P.M. (1985), *Theory and Implementation of EQS: A Structural Equations Program*, Manual for Program Version 2.0, Los Angeles: BMDP Statistical Software, Inc.
- Bentler, P.M. (1986), *Lagrange Multiplier and Wald Tests for EQS and EQS/PC*, Los Angeles: BMDP Statistical Software, Inc.
- Bentler, P.M. (1989), *EQS, Structural Equations, Program Manual*, Program Version 3.0, Los Angeles: BMDP Statistical Software, Inc.
- Bentler, P.M. and Bonett, D.G. (1980), "Significance Tests and Goodness of Fit in the Analysis of Covariance Structures," *Psychological Bulletin*, 88, 588–606.
- Bentler, P.M. and Weeks, D.G. (1980), "Linear Structural Equations with Latent Variables," *Psychometrika*, 45, 289–308.
- Bentler, P.M. and Weeks, D.G. (1982), "Multivariate Analysis with Latent Variables," in *Handbook of Statistics, Vol. 2*, eds. P.R. Krishnaiah and L.N. Kanal, Amsterdam: North Holland Publishing Company.

- Bishop, Y.M.M., Fienberg, S.E., and Holland, P.W. (1977), *Discrete Multivariate Analysis: Theory and Practice*, Cambridge and London: MIT Press.
- Bollen, K.A. (1986), "Sample Size and Bentler and Bonett's Nonnormed Fit Index," *Psychometrika*, 51, 375–377.
- Bollen, K.A. (1989a), "A New Incremental Fit Index for General Structural Equation Models," *Sociological Methods and Research*, 17, 303–316.
- Bollen, K.A. (1989b), *Structural Equations with Latent Variables*, New York: John Wiley & Sons, Inc.
- Bozdogan, H. (1987), "Model Selection and Akaike's Information Criterion (AIC): The General Theory and its Analytical Extensions," *Psychometrika*, 52, 345–370.
- Browne, M.W. (1974), "Generalized Least Squares Estimators in the Analysis of Covariance Structures," *South African Statistical Journal*, 8, 1–24.
- Browne, M.W. (1982), "Covariance Structures," in *Topics in Multivariate Analyses*, ed. D.M. Hawkins, New York: Cambridge University Press.
- Browne, M. W. (1984), "Asymptotically Distribution-Free Methods for the Analysis of Covariance Structures," *British Journal of Mathematical and Statistical Psychology*, 37, 62–83.
- Browne, M. W. (1992), "Circumplex Models for Correlation Matrices," *Psychometrika*, 57, 469–497.
- Browne, M. W. and Cudeck, R. (1993), "Alternative Ways of Assessing Model Fit," in *Testing Structural Equation Models*, eds. K. A. Bollen and S. Long, Newbury Park, CA: Sage Publications, Inc.
- Browne, M. W. and Du Toit, S.H.C. (1992), "Automated Fitting of Nonstandard Models," *Multivariate Behavioral Research*, 27, 269–300.
- Browne, M.W. and Shapiro, A. (1986), "The Asymptotic Covariance Matrix of Sample Correlation Coefficients under General Conditions," *Linear Algebra and its Applications*, 82, 169–176.
- Bunch, J.R. and Kaufman, K. (1977), "Some Stable Methods for Calculating Inertia and Solving Symmetric Linear Systems," *Mathematics of Computation*, 31, 162–179.
- Buse, A. (1982), "The Likelihood Ratio, Wald, and Lagrange Multiplier Tests: An Expository Note," *The American Statistician*, 36, 153–157.
- Chamberlain, R.M., Powell, M.J.D., Lemarechal, C., and Pedersen, H.C. (1982), "The Watchdog Technique for Forcing Convergence in Algorithms for Constrained Optimization," *Mathematical Programming*, 16, 1–17.
- Cramer, J.S. (1986), *Econometric Applications of Maximum Likelihood Methods*, Cambridge: Cambridge University Press.
- DeLeeuw, J. (1983), "Models and Methods for the Analysis of Correlation Coefficients," *Journal of Econometrics*, 22, 113–137.

- Dennis, J.E., Gay, D.M., and Welsch, R.E. (1981), "An Adaptive Nonlinear Least-Squares Algorithm," *ACM Trans. Math. Software*, 7, 348–368.
- Dennis, J.E. and Mei, H.H.W. (1979), "Two New Unconstrained Optimization Algorithms which use Function and Gradient Values," *J. Optim. Theory Appl.*, 28, 453–482.
- Dennis, J.E. and Schnabel, R.B. (1983), *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, New Jersey: Prentice-Hall.
- Dijkstra, T. K. (1992), "On Statistical Inference with Parameter Estimates on the Boundary of the Parameter Space," *British Journal of Mathematical and Statistical Psychology*, 45, 289–309.
- Everitt, B.S. (1984), *An Introduction to Latent Variable Methods*, London: Chapman & Hall.
- Fletcher, R. (1980), *Practical Methods of Optimization*, Vol. 1, Chichester: John Wiley & Sons, Inc.
- Fletcher, R. (1987), *Practical Methods of Optimization*, Second Edition, Chichester: John Wiley & Sons, Inc.
- Fletcher, R. and Powell, M.J.D. (1963), "A Rapidly Convergent Descent Method for Minimization," *Comput.J.*, 6, 163–168.
- Fletcher, R. and Xu, C. (1987), "Hybrid Methods for Nonlinear Least Squares," *J. Numerical Analysis*, 7, 371–389.
- Fuller, A.W. (1987), *Measurement Error Models*, New York: John Wiley & Sons, Inc.
- Gallant, A. R. (1987), *Nonlinear Statistical Models*, New York: John Wiley & Sons, Inc.
- Gay, D.M. (1983), "Subroutines for Unconstrained Minimization," *ACM Trans. Math. Software*, 9, 503–524.
- Gill, E.P., Murray, W., Saunders, M.A., and Wright, M.H. (1983), "Computing Forward-Difference Intervals for Numerical Optimization," *SIAM J. Sci. Stat. Comput.*, 4, 310–321.
- Gill, E.P., Murray, W., and Wright, M.H. (1981), *Practical Optimization*, London: Academic Press.
- Gill, E.P., Murray, W., Saunders, M.A., and Wright, M.H. (1984), "Procedures for Optimization Problems with a Mixture of Bounds and General Linear Constraints," *ACM Trans. Math. Software*, 10, 282–298.
- Goldfeld, S.M., Quandt, R.E., and Trotter, H.F. (1966), "Maximization by Quadratic Hill-Climbing," *Econometrica*, 34, 541–551.
- Guttman, L. (1953), "Image Theory for the Structure of Quantitative Variates," *Psychometrika*, 18, 277–296.

- Guttman, L. (1957), "Empirical Verification of the Radex Structure of Mental Abilities and Personality Traits," *Educational and Psychological Measurement*, 17, 391–407.
- Häggglund, G. (1982), "Factor Analysis by Instrumental Variable Methods," *Psychometrika*, 47, 209–222.
- Hartmann, W.M. (1992), *The NLP Procedure: Extended User's Guide*, Cary: SAS Institute Inc.
- Hartmann, W. M. and Hartwig, R. E. (1995), "Computing the Moore-Penrose Inverse for the Covariance Matrix in Constrained Nonlinear Estimation," accepted for publication in *SIAM Journal on Optimization*.
- Hartmann, W. M. and So, Y. (1995), "Nonlinear Least-Squares and Maximum-Likelihood Estimation Using PROC NLP and SAS/IML," Computer Technology Workshop, American Statistical Association, Joint Statistical Meeting, Orlando, 1995.
- Hoelter, J.W. (1983), "The Analysis of Covariance Structures: Goodness-of-Fit Indices," *Sociological Methods and Research*, 11, 325–344.
- James, L.R., Mulaik, S.A., and Brett, J.M. (1982), *Causal Analysis: Assumptions, Models, and Data*, Beverly Hills: Sage Publications, Inc.
- Jennrich, R.I. (1987), "Tableau Algorithms for Factor Analysis by Instrumental Variable Methods," *Psychometrika*, 52, 469–476.
- Jöreskog, K.G. (1963), *Statistical Estimation in Factor Analysis*, Stockholm: Almqvist & Wicksell.
- Jöreskog, K.G. (1969), "Efficient Estimation in Image Factor Analysis," *Psychometrika*, 34, 51–75.
- Jöreskog, K.G. (1973), "A General Method for Estimating a Linear Structural Equation System," in *Structural Equation Models in the Social Sciences*, eds. A.S. Goldberger and O.D. Duncan, New York: Academic Press.
- Jöreskog, K.G. (1978), "Structural Analysis of Covariance and Correlation Matrices," *Psychometrika*, 43, 443–477.
- Jöreskog, K.G. (1982), "Analysis of Covariance Structures," in *A Second Generation of Multivariate Analysis*, ed. C. Fornell, New York: Praeger Publishers.
- Jöreskog, K.G. and Sörbom, D. (1979), *Advances in Factor Analysis and Structural Equation Modeling*, Cambridge MA: Abt Books.
- Jöreskog, K.G. and Sörbom, D. (1985), *LISREL VI: Analysis of Linear Structural Relationships by Maximum Likelihood, Instrumental Variables, and Least Squares*, Uppsala: University of Uppsala.
- Jöreskog, K.G. and Sörbom, D. (1988), *LISREL 7: A Guide to the Program and Applications*, Chicago, Illinois: SPSS Inc.
- Keesling, J.W. (1972), "Maximum Likelihood Approaches to Causal Analysis," Ph.D. dissertation, Chicago, 1972.

- Kmenta, J. (1971), *Elements of Econometrics*, New York: Macmillan Publishing Co.
- Krane, W.R. and McDonald, R.P. (1978), "Scale Invariance and the Factor Analysis of Correlation Matrices," *British Journal of Mathematical and Statistical Psychology*, 31, 218–228.
- Lawley, D.N. and Maxwell, A.E. (1971), *Factor Analysis as a Statistical Method*, New York: American Elsevier Publishing Company.
- Lee, S.Y. (1985), "On Testing Functional Constraints in Structural Equation Models," *Biometrika*, 72, 125–131.
- Lee, S.Y. and Jennrich, R.I. (1979), "A Study of Algorithms for Covariance Structure Analysis with Specific Comparisons Using Factor Analysis," *Psychometrika*, 44, 99–113.
- Loehlin, J.C. (1987), *Latent Variable Models, An Introduction to Factor, Path, and Structural Analysis*, Hillsdale, NJ: L. Erlbaum Associates.
- Long, J.S. (1983), *Covariance Structure Models, an Introduction to LISREL*, Beverly Hills, CA: SAGE Publications, Inc.
- MacCallum, R. (1986), "Specification Searches in Covariance Structure Modeling," *Psychological Bulletin*, 100, 107–120.
- Marsh, H.W., Balla, J.R. and McDonald, R.P. (1988), "Goodness-of-Fit Indices in Confirmatory Factor Analysis. The Effect of Sample Size," *Psychological Bulletin*, 103, 391–410.
- McArdle, J.J. (1980), "Causal Modeling Applied to Psychonomic Systems Simulation," *Behavior Research Methods & Instrumentation*, 12, 193–209.
- McArdle, J.J. (1988), "Dynamic but Structural Equation Modeling of Repeated Measures Data," in *The Handbook of Multivariate Experimental Psychology*, eds. J.R. Nesselroade and R.B. Cattell, New York: Plenum Press.
- McArdle, J.J. and Boker, S.M. (1986), *RAMpath - Path Diagram Software*, Denver: DATA Transforms, Inc.
- McArdle, J.J. and McDonald, R.P. (1984), "Some Algebraic Properties of the Reticular Action Model," *British Journal of Mathematical and Statistical Psychology*, 37, 234–251.
- McDonald, R.P. (1978), "A Simple Comprehensive Model for the Analysis of Covariance Structures," *British Journal of Mathematical and Statistical Psychology*, 31, 59–72.
- McDonald, R.P. (1980), "A Simple Comprehensive Model for the Analysis of Covariance Structures: Some Remarks on Applications," *British Journal of Mathematical and Statistical Psychology*, 33, 161–183.
- McDonald, R.P. (1984), "Confirmatory Models for Nonlinear Structural Analysis," in *Data Analysis and Informatics, III*, eds. E. Diday et al., North Holland: Elsevier Publishers.
- McDonald, R.P. (1985), *Factor Analysis and Related Methods*, Hillsdale, NJ: Lawrence Erlbaum Associates.

- McDonald, R.P. (1989), "An Index of Goodness-of-Fit Based on Noncentrality," *Journal of Classification*, 6, 97–103.
- McDonald, R.P. and Hartmann, W. (1992), "A Procedure for Obtaining Initial Values of Parameters in the RAM Model," *Multivariate Behavioral Research*, 27, 57–176.
- McDonald, R.P. and Marsh, H.W. (1988), "Choosing a Multivariate Model: Noncentrality and Goodness of Fit," distributed paper.
- McDonald, R.P., Parker, P.M., and Ishizuka, T. (1993), "A Scale-Invariant Treatment of Recursive Path Models," *Psychometrika*, 58, 431–443.
- More, J.J. (1978), "The Levenberg-Marquardt Algorithm: Implementation and Theory," in *Numerical Analysis—Dundee 1977*, ed. G.A. Watson, Lecture Notes in Mathematics 630, Berlin: Springer-Verlag.
- More, J.J. and Sorensen, D.C. (1983), "Computing a Trust-Region Step," *SIAM J. Sci. Stat. Comput.*, 4, 553–572.
- Mulaik, S.A., James, L.R., Van Alstine, J., Bennett, N., Lind, S. and Stilwell, C.D. (1989), "Evaluation of Goodness-of-Fit Indices for Structural Equation Models," *Psychological Bulletin*, 105, 430–445.
- Muthén, B.O. (1987), *LISCOMP: Analysis of Linear Structural Relations Using a Comprehensive Measurement Model*, Mooresville IN: Scientific Software, Inc.
- Polak, E. (1971), *Computational Methods in Optimization*, New York: Academic Press, Inc.
- Powell, J.M.D. (1977), "Restart Procedures for the Conjugate Gradient Method," *Math. Prog.*, 12, 241–254.
- Powell, J.M.D. (1978a), "A Fast Algorithm for Nonlinearly Constraint Optimization Calculations," in *Numerical Analysis, Dundee 1977, Lecture Notes in Mathematics 630*, ed. G.A. Watson, Berlin: Springer Verlag, 144–175.
- Powell, J.M.D. (1978b), "Algorithms for Nonlinear Constraints that Use Lagrangian Functions," *Mathematical Programming*, 14, 224–248.
- Powell, M.J.D. (1982a), "Extensions to Subroutine VF02AD," in *Systems Modeling and Optimization, Lecture Notes In Control and Information Sciences 38*, eds. R.F. Drenick and F. Kozin, Berlin: Springer Verlag, 529–538.
- Powell, J.M.D. (1982b), "VMCWD: A Fortran Subroutine for Constrained Optimization," *DAMTP 1982/NA4*, Cambridge, England.
- Powell, J.M.D. (1992), "A Direct Search Optimization Method that Models the Objective and Constraint Functions by Linear Interpolation," *DAMTP/NA5*, Cambridge, England.
- Saris, W. (1982), "Linear Structural Relationships," in *A Second Generation of Multivariate Analysis*, ed. C. Fornell, New York: Praeger Publishers.

- Schmid, J. and Leiman, J.M. (1957), "The Development of Hierarchical Factor Solutions," *Psychometrika*, 22, 53–61.
- Schwarz, G. (1978), "Estimating the Dimension of a Model," *Annals of Statistics*, 6, 461–464.
- Sclove, L.S. (1987), "Application of Model-Selection Criteria to Some Problems in Multivariate Analysis," *Psychometrika*, 52, 333–343.
- Steiger, J.H. and Lind, J.C. (1980), "Statistically Based Tests for the Number of Common Factors," paper presented at the annual meeting of the Psychometric Society, Iowa City, IA.
- Swaminathan, H. (1974), "A General Factor Model for the Description of Change," Report LR-74-9, Laboratory of Psychometric and Evaluative Research, University of Massachusetts.
- Tucker, L.R. and Lewis, C. (1973), "A Reliability Coefficient for Maximum Likelihood Factor Analysis," *Psychometrika*, 38, 1–10.
- Wheaton, B., Muthén, B., Alwin, D.F., and Summers, G.F. (1977), "Assessing Reliability and Stability in Panel Models," in *Sociological Methodology*, ed. D.R. Heise, San Francisco: Jossey Bass.
- Wiley, D.E. (1973), "The Identification Problem for Structural Equation Models with Unmeasured Variables," in *Structural Equation Models in the Social Sciences*, eds. A.S. Goldberger and O.D. Duncan, New York: Academic Press.
- Wilson, E.B. and Hilferty, M.M. (1931), "The Distribution of Chi-Square," *Proceeding of the National Academy of Science*, 17, 694.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/STAT® User's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS/STAT® User's Guide, Version 8

Copyright © 1999 by SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-494-2

All rights reserved. Produced in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of the software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

The Institute is a private company devoted to the support and further development of its software and related services.