Behavior Research Methods, Instruments, & Computers
2003, 35 (3), 310-317

An SAS/IML procedure for maximum likelihood factor analysis

RUSAN CHEN
Georgetown University, Washington, D.C.

Maximum likelihood factor analysis (MLFA), originally introduced by Lawley (1940), is based on a firm mathematical foundation that allows hypothesis testing when normality is assumed with large sample sizes. MLFA has gained in popularity since Jöreskog (1967) implemented an iterative algorithm to estimate parameters. This article presents a concise program using matrix language SAS/IML with the optimization subroutine NLPQN to obtain MLFA solutions. The program is pedagogically useful because it shows the step-by-step computational processes for MLFA, whereas almost all other statistical packages for MLFA are in "black boxes." It is also demonstrated that this approach can be extended to other multivariate methods requiring numerical optimizations, such as the widely used structural equation modeling. Researchers may find this program useful in conducting Monte Carlo simulation studies to investigate the properties of multivariate methods that involve numerical optimizations.

Factor analysis is one of the most widely used multivariate methods in the behavioral sciences. As one of the factor analysis methods, maximum likelihood factor analysis (MLFA) enables researchers to test the hypothesis that the number of specified common factors is sufficient. MLFA also allows the construction of confidence intervals for the estimated parameters when the multivariate normality assumption is satisfied and the sample size is large.

Suppose that the basic model of factor analysis holds in the population:

$$\mathbf{x} = \boldsymbol{\mu} + \Lambda\mathbf{f} + \boldsymbol{\varepsilon}, \tag{1}$$

where $\mathbf{x}$ is a column vector of p variables, $\boldsymbol{\mu}$ is the mean vector of $\mathbf{x}$, $\mathbf{f}$ is a column vector of k ($k < p$) common factors, Λ is a $p \times k$ matrix of factor loadings, and $\boldsymbol{\varepsilon}$ is a vector of p unique factors. The covariance matrices of $\mathbf{x}$, $\mathbf{f}$, and $\boldsymbol{\varepsilon}$ are denoted respectively as Σ, Φ, and Ψ. The matrix Ψ is a diagonal matrix with p diagonal elements known as unique variances. On the basis of the assumptions that $E(\mathbf{f}, \boldsymbol{\varepsilon}) = 0$, $E(\mathbf{f}) = 0$, $E(\boldsymbol{\varepsilon}) = 0$, and Φ is an identity matrix $\mathbf{I}_k$, the covariance matrix of $\mathbf{x}$ becomes

$$\Sigma = \Lambda\Lambda' + \Psi. \tag{2}$$

The parameters Λ and Ψ in Formula 2 must be estimated from a sample. Suppose that we draw a random sample of size n from the population and that we have also obtained the observed measures for the p variables from the n units. We can compute the sample covariance matrix $\mathbf{S}$. Since the observed $\mathbf{S}$ is an unbiased estimate of Σ, the remaining problem becomes to estimate Λ and Ψ with Formula 2 to fit Σ, which is estimated by $\mathbf{S}$. In most cases, the measurement scales of the p variables are arbitrary and standardized, and then $\mathbf{S}$ is a correlation matrix.

To restate the problem: Given $\mathbf{S}$ with $p(p + 1)/2$ known elements, the researcher needs to estimate Λ and Ψ with, altogether, $pk + p$ unknown elements. To determine a unique solution for Λ and Ψ within a factor analysis model, the matrix $\Lambda'\Psi^{-1}\Lambda$ is imposed arbitrarily as a diagonal matrix with distinctive elements on the diagonal. Lawley (1940, 1941) first applied the maximum likelihood method to estimate Λ and Ψ. Assuming that the p variables are multivariate normally distributed, the log likelihood function of Σ, as a function of the elements of Λ and Ψ, is

$$\log L = -(n/2)\,[\log|\Sigma| + \text{trace}\,(\mathbf{S}\Sigma^{-1})]. \tag{3}$$

The efficient estimation of Λ and Ψ is to maximize the value of L. Equivalently, it can be achieved more conveniently to minimize the function

$$F(\Lambda, \Psi) = \log|\Sigma| + \text{trace}\,(\mathbf{S}\Sigma^{-1}) - \log|\mathbf{S}| - p. \tag{4}$$

Lawley (1940, 1941) considered an iterative numerical procedure for minimizing the value of F, but the method was not very successful. Since then a number of iterative methods have been introduced for minimizing F or maximizing the likelihood function L, including those in Emmett (1949), Bargmann (1957), and Lawley and Maxwell (1963). In general, with these methods, a numerical solution to the equations $\partial F/\partial\Lambda = 0$, and $\partial F/\partial\Psi = 0$ has been sought. However, most of these iterative methods converged slowly, especially at the final steps of the iter-

Work on this article was partially supported by Grant DK 56975 from the National Institutes of Health. The author thanks Ching-Fan Sheu and Diana Suhr for their expert comments that substantially improved the manuscript. The author also thanks Randi Streisand for suggestions on writing style. Correspondence concerning this article should be addressed to R. Chen, 314 Car Barn Building, Center for New Designs in Learning and Scholarship, Georgetown University, N.W. Washington, DC 20057 (e-mail: chenrs@georgetown.edu).

ation process. For some data sets, the proposed methods even failed to converge; therefore, these methods were not very satisfactory.

Jöreskog's Numerical Solution to MLFA

Jöreskog (1966, 1967, 1977) and Jöreskog and Lawley (1968) introduced and implemented in Fortran a method that was completely successful in directly minimizing the function F. In this method, it was proposed that for a given Ψ, Λ could be determined by the formula

$$\Lambda = \Psi^{1/2}\,\Omega\,(\Theta - \mathbf{I})^{1/2}, \tag{5}$$

where Ω is a $p \times k$ matrix with columns as the first k latent vectors of the matrix $\Psi^{-1/2}\,\mathbf{S}\,\Psi^{-1/2}$ and Θ is a $k \times k$ diagonal matrix with the largest k latent roots of matrix $\Psi^{-1/2}\,\mathbf{S}\,\Psi^{-1/2}$ as its diagonal elements. Thus, the problem becomes to minimize F with respect to Ψ, instead of with respect to both Ψ and Λ. A quasi-Newton minimization method published by Fletcher and Powell (1963) was implemented in this method to achieve the F minimization process. The mathematical derivation for the theory and the computational method were very well presented in Lawley and Maxwell (1971).

Numerical algorithms that are different from Jöreskog's method have been proposed to minimize the F value to estimate Λ and Ψ within the MLFA model. Jennrich and Robinson (1969) recommended a Newton–Raphson algorithm, and Rubin and Thayer (1982) introduced an EM algorithm for MLFA solutions. Fuller (1987) described an algorithm that was a modification based on Jöreskog (1967) and Jennrich and Robinson (1969). Fuller's method is implemented and available in SAS/STAT PROC FACTOR with option METHOD=ML. However, since Jöreskog's solution is the first successful and the most popular method for MLFA solutions, the program presented in this article is based on Jöreskog's method.

SAS/IML Implementation of Jöreskog's Method for MLFA

The program listings in Appendix A demonstrate the step-by-step computational process for MLFA based on Jöreskog's method. As one of the procedures available in the SAS statistical package, SAS/IML is a high-level matrix language for programming purposes (SAS Institute, 2000). SAS/IML software includes a set of nonlinear optimization subroutines for estimation of constrained or unconstrained parameters through iterative processes. Specifically, the NLPQN subroutine that uses the quasi-Newton optimization method was selected in this program. One of the options in the NLPQN subroutine (Option [4] = 4) performs the original Davidon, Fletcher, and Powell (DFP) method to update the inverse Hessian matrix. The DFP method performs a line search in each iteration on the search direction with quadratic interpolation and cubic extrapolation. This iteration method is similar to the one originally introduced by Jöreskog (1967). The Module "FUN" computes and returns the function value $F(\Lambda,\Psi)$ using Formula 4, assuming that the unique variances (e.g., the diagonal elements of Ψ) are known. The Λ matrix is calculated using Formula 5, and the Σ is then fitted by Λ and Ψ.

The Module "GRAD" specifies the gradient function to compute the first-order derivative $\partial F/\partial\Psi = \text{diag}\,[\Psi^{-1}(\Sigma - \mathbf{S})\,\Psi^{-1}]$ (Jöreskog, 1967, p. 450). When this module is not specified, the NLPQN subroutine will numerically approximate the gradient vector by the finite differences method. However, it usually requires more calls to the function module for the iterative process to converge. For intensive computational tasks, such as Monte Carlo simulation studies, it is recommended that the gradient function be specified if it is available.

The convenient starting values of the iterative process for the unique variances can be computed from $(1 - k/(2p))(1/s^{ii})$, $(i = 1, 2, \ldots, p)$, where s^{ii} is the diagonal elements of the inverse matrix of $\mathbf{S}$. Jöreskog (1963) justified that these initial values worked well in practice. In this program, the starting values are stored in a vector $\mathbf{x}$ as input argument for Modules "FUN" and "GRAD."

The first two arguments for the NLPQN subroutine are the returned results when the iterative process terminates. The first argument, *rc* (return code), returns a number indicating how the iterative process is terminated. If $rc > 0$, the iterative process is terminated successfully with one of the specified criteria. A return code of $rc < 0$ indicates unsuccessful termination, and thus, the results are not reliable. The second argument of the NLPQN subroutine is a vector of length equal to the length of the input vector $\mathbf{x}$, containing the optimal values when $rc > 0$. In this program, the second argument, $\mathbf{xr}$, is a vector containing the estimated unique variances when the F value is minimized. The rest of the arguments are input information for the NLPQN subroutine. The "*grd*=" statement specifies the gradient module. This statement can be omitted if the gradient module is not specified in the program. In such cases, the NLPQN subroutine will compute the approximate gradient vector numerically by finite difference.

The input argument *con* is a matrix to specify parameter constraints within defined boundaries. In this program, the argument *con* is a $2 \times p$ matrix with all elements in the first row equal to 1×10^{-6} and all elements in the second row specified as missing values. The first row defines the lower bounds, and the second row defines the upper bounds for the p parameters. The missing values in the second row substitute the largest floating-point values. The unique variances in this program are thus constrained with a lower bound of 1×10^{-6}, to prevent their becoming zero or negative, an improper solution known as Heywood cases. The value 1×10^{-6} is arbitrary, but a value greater than 1×10^{-6} may lead to different estimates for the factor loadings when Heywood cases occur. It should be clarified that although the parameters are constrained within certain boundaries, the MLFA model presented in this program is termed *unrestricted*,

as compared with *restricted* MLFA models, because the number of parameters is not constrained; rather, all the parameters are free to be estimated.

One of the advantages of using MLFA is to allow the testing of the hypothesis that the specified number of factors is sufficient for the population. When the maximum-likelihood estimates of Λ and Ψ have been determined with F minimized, the value of F multiplied by a factor of $n - (2p + 5)/6 - 2k/3$ is nearly distributed as χ^2, with degrees of freedom = $[(p - k)^2 - (p + k)]/2$. Jöreskog and Lawley (1968) cautioned that this likelihood ratio test is valid only when the sample size n is reasonably large, with a safe rule of $n > 50$. To test this hypothesis the multivariate normality assumption is required, although the estimation of Λ and Ψ does not require the normality assumption.

In addition to the significance test with the approximate χ^2 distribution, researchers may also use Akaike's information criterion (AIC) to determine the number of factors to be retained. Akaike (1987) described the application of AIC for model selection in MLFA solutions. When F is minimized and χ^2 is computed as a function of the F value, AIC = $\chi^2 - 2\ (df)$. For models with different numbers of specified factors derived from the same correlation matrix, the rule is that the model with the smallest value of AIC is considered the best. The PROC FACTOR in SAS/STAT uses the χ^2 without Bartlett's (1951) correction, $\chi^2 = (n - (2p+5)/6)F$, to compute AIC. This program follows the same practice.

The sample correlation matrix listed in this program was also used in Emmett (1949) and Jöreskog (1967) to demonstrate their iterative methods for MLFA solutions. This correlation matrix with nine variables was originally obtained (Emmett, 1949) from 221 boys to explore the possibility of the existence of a space factor different from the other IQ factors, such as verbal, numerical, and memory. The nine variables are verbal intelligence, English, mechanical arithmetic, arithmetic problems, a matrix test, two nonverbal tests, and two tests of spatial ability. This correlation matrix is known to have an acceptable proper solution with a three-factor model. The results obtained from this program are consistent with those presented in Emmett (1949) and Jöreskog (1967). In order to compare the results from this program with that using PROC FACTOR, Appendix B presents a program, using the correlation matrix as input, for conducting MLFA with PROC FACTOR. The results show that the unrotated factor loadings, the communalities, the χ^2 test indices, and the AIC are the same from the two programs.

The output from a SAS/IML program is flexible and pedagogically useful. For example, the SAS/IML program user is able to observe the computational processes step by step for parameter estimates and test statistics obtained with MLFA. The researcher may also compute the newly published indices not available in PROC FACTOR. The program in Appendix A provides an indication of successful iteration processing, estimated factor loadings, unique variances, communalities, the χ^2 test results, and the AIC.

Extension to Structural Equation Modeling

The SAS/IML program presented in Appendix A is an example of using high-level matrix language with its ready-to-use optimization subroutines to perform modeling methods requiring numerical optimizations. This approach can also be applied to other statistical methods involving numerical optimizations, such as the widely used structural equation modeling (SEM). Cudeck, Klebe, and Henly (1993) introduced a SAS/IML implementation to conduct SEM with numerical approximation of matrix derivatives for the minimization process. The use of the NLPQN subroutine can substantially reduce the program listings and make the program very concise. A SAS/IML procedure using the NLPQN subroutine, modified from Cudeck et al.'s program, is presented in Appendix C. Cudeck et al. used the stability of alienation model (Wheaton, Muthen, Alwin, & Summers, 1977) as a numerical example to illustrate their implementation. To compare the results from Cudeck and colleagues' implementation and that from the program in Appendix C, the same example was used. The initial research interest of the stability and alienation model was to determine whether social attitudes, such as alienation and social distance toward minority groups, were relatively stable or highly volatile over time. The covariance matrix has six variables: anomia, measured in 1967 and 1971; powerlessness, measured in 1967 and 1971; education; and the occupational status index. The data were collected from 932 people in rural areas in Illinois.

The results from the program in Appendix C and those from Cudeck et al. (1993) are identical, including parameter estimates, standard errors for parameters, and the likelihood ratio test statistics. Readers familiar with PROC CALIS, the standard SAS procedure for SEM, may be interested in comparing the results from the program in Appendix C with that from PROC CALIS. Appendix D presents a program using SAS CALIS for estimating the stability of alienation model. It can be seen that the results from the program in Appendix C are identical to the results obtained using PROC CALIS.

The subroutine NLPFDD in Appendix C is used to produce the Hessian matrix required for the computation of standard errors for the parameters. It should be noted that the program does not include the details, such as to check whether the sample covariance matrix is positive definite.

Cudeck et al. (1993) presented examples to show that this general approach of using matrix language is useful when some models are difficult or even impossible to implement with most available programs. They also pointed out that the program is useful for incorporating the newly published methodological developments that are not available in commercial software. On the basis of these judgments, the program in Appendix C is likely to find a wide range of applications for behavior research.

This SAS/IML approach using the built-in maximization subroutines is useful when conducting Monte Carlo studies to investigate the properties of statistical methods involving numerical maximization. A broad range of statistical methods for the social and behavioral sciences

requires numerical optimization to obtain parameter estimates, including covariance structure modeling, random coefficient modeling, and latent class modeling (Arminger, Clogg, & Sobel, 1995). Researchers do not need to program and test their own maximization process, a technique that typically requires special training and skills. The program listings could be very concise, as can be seen in the comparison of the program in Appendix C with the one in Cudeck et al. (1993).

Furthermore, users of other matrix languages, such as S-Plus, R, Gauss and Matlab, may find similar computing environments in which to implement this approach.

REFERENCES

Akaike, H. (1987). Factor analysis and AIC. *Psychometrika*, **52**, 317-332.

Arminger, G., Clogg, C. C., & Sobel, M. E. (1995). *Handbook of statistical modeling for the social and behavioral sciences*. New York: Plenum.

Bargmann, R. (1957). *A study of independence and dependence in multivariate normal analysis* (Mimeo Series No. 186). Chapel Hill: University of North Carolina, Institute of Statistics.

Bartlett, M. S. (1951). The effect of standardization on an approximation in factor analysis. *Biometrika*, **38**, 337-344.

Cudeck, R., Klebe, K., & Henly, S. (1993). A simple Gauss–Newton procedure for covariance structure analysis with high-level computer language. *Psychometrika*, **58**, 211-232.

Emmett, W. G. (1949). Factor analysis by Lawley's method of maximum likelihood. *British Journal of Mathematical & Statistical Psychology*, **2**, 90-97.

Fletcher, R., & Powell, M. J. D. (1963). A rapid convergent descent method for minimization. *Computer Journal*, **6**, 163-168.

Fuller, W. A. (1987). *Measurement error models*. New York: Wiley.

Jennrich, R. I., & Robinson, S. M. (1969). A Newton–Raphson algorithm for maximum likelihood factor analysis. *Psychometrika*, **34**, 111-123.

Jöreskog, K. G. (1963). *Statistical estimation in factor analysis*. Stockholm: Almqvist & Wiksell.

Jöreskog, K. G. (1966). *UMFLA: A computer program for unrestricted maximum likelihood factor analysis* (Research Memorandum 66-20). Princeton, NJ: Educational Testing Services.

Jöreskog, K. G. (1967). Some contributions to maximum likelihood factor analysis. *Psychometrika*, **32**, 443-482.

Jöreskog, K. G. (1977). Factor analysis by least square and maximum-likelihood methods. In K. Enslein, A. Ralston, & H. S. Wilf (Eds.), *Statistical methods for digital computers* (Vol. III, pp. 125-165). New York: Wiley.

Jöreskog, K. G., & Lawley, D. N. (1968). New methods in maximum likelihood factor analysis. *British Journal of Mathematical & Statistical Psychology*, **21**, 85-96.

Lawley, D. N. (1940). The estimation of factor loadings by the method of maximum likelihood. *Proceedings of the Royal Society of Edinburgh* (A), **60**, 64-82.

Lawley, D. N. (1941). Further investigation in factor estimation. *Proceedings of the Royal Society of Edinburgh* (A), **61**, 176-185.

Lawley, D. N., & Maxwell, A. E. (1963). *Factor analysis as a statistical method.* London: Butterworth.

Lawley, D. N., & Maxwell, A. E. (1971). *Factor analysis as a statistical method.* New York: Macmillan.

Rubin, D. B., & Thayer, D. T. (1982). EM algorithm for ML factor analysis. *Psychometrika*, **47**, 69-76.

SAS Institute (2000). *SAS/IML user's guide, Version 8*. Cary, NC: Author.

Wheaton, D. E., Muthen, B., Alwin, D. F., & Summers, G. F. (1977). Assessing reliability and stability in panel models. In D. R. Heise (Ed.), *Sociological methodology* (pp. 84-136). San Francisco: Jossey-Bass.

APPENDIX A

```
/*************************************************************/
/*   SAS/IML Procedure for MLFA using subroutine NLPQN       */
/*                                                           */
/*       n: sample size                                      */
/*       p: number of observed variables                     */
/*       k: number of factors   (k < p)                      */
/*       s: sample covariance matrix for p variables         */
/*   sigma: population covariance matrix for p variables     */
/*  lambda: p ( k factor loading matrix                      */
/*   omega: p ( k matrix defined in Equation 5               */
/*   theta: k ( k diagonal matrix defined in Equation 5      */
/*     psi: diagonal matrix for unique variance              */
/*************************************************************/
;

TITLE1 'SAS/IML Procedure for MLFA using subroutine NLPQN';
TITLE2 'The example was used in Emmett (1949) and J reskog
(1967)';

proc iml;
reset noname nocenter;

start mlfa;
```

APPENDIX A (Continued)

```
/*---- Objective Function Module -----------------------*/
start fun(x) global(s, k, p, xpsy, sigma, lambda, omega, f);
  xpsy = diag(1/sqrt(x));
  A = xpsy*s*xpsy;
  omega = (eigvec(A))[,1:k];
  theta = (eigval(A))[1:k];
  lambda = diag(sqrt(x))*omega*diag(sqrt(abs(theta-1)));
  sigma = lambda*t(lambda)+diag(x);
  f = log(det(sigma))+trace(s*inv(sigma))-log(det(s))-p;
  return(f);
finish;

/*--- Gradient Function Module -------------------------*/
  start grad(x) global(s, sigma);
  invpsy = inv(diag(x));
  g = (vecdiag(invpsy*(sigma-s)*invpsy))`;
  return(g);
  finish;

/*---Starting Values for Iteration----------------------*/
  p = ncol(s);
  x = (1-k/(2#p))#(1/vecdiag(inv(s)));

/*---Set up Options and Constraints---------------------*/
  option = {0 0 . 4};
  con = J(1,p,0.000001)//J(1,p,.);

call nlpqn(rc, xr, "fun", x, option, con) grd = "grad";

/*---Compute Likelihood Ratio Test and AIC -------------*/
  chisq = (n-(2#p+5)/6-2#k/3)#f;
  df =((p-k)##2-(p+k))/2;
  pvalue = 1 - probchi(chisq, df);
  AIC = (n - (2#p+5)/6)#f - 2#df;
  psi = xr`;
  comm = (1 - psi);
finish;

/*---- Sample Correlation Matrix -----------------------*/
  s ={ 1.000 .5232 .3950 .4706 .3455 .4262 .5761 .4338 .6393,
       .5232 1.000 .4792 .5060 .4181 .4619 .5469 .2829 .6445,
       .3950 .4792 1.000 .3554 .2701 .2536 .4524 .2185 .5038,
       .4706 .5060 .3554 1.000 .6906 .7909 .4427 .2852 .5050,
       .3455 .4181 .2701 .6906 1.000 .6794 .3825 .1488 .4091,
       .4262 .4619 .2536 .7909 .6794 1.000 .3721 .3138 .4721,
       .5761 .5469 .4524 .4427 .3825 .3721 1.000 .3846 .6801,
       .4338 .2829 .2185 .2852 .1488 .3138 .3846 1.000 .4700,
       .6393 .6445 .5038 .5050 .4091 .4721 .6801 .4700 1.000};
  n = 221;

/*---Specify Number of Factors and Run the Program --------*/
  k = 3;
run mlfa;

/*--- print out the results -------------------------------*/
```

APPENDIX A (Continued)

```
   if rc>0 then print
   '**The iterative process terminates successfully**';
   else print '**Warning: Unsuccessful Termination**';
   print
   lambda(|colname="Factor Loadings" format = 7.3|)
   psi   (|colname="Unique Variance" format = 6.3|)
   comm  (|colname="Communality" format=6.3|),
   chisq (|colname="Chi-Square" format=10.3|)
   df    (|colname="DF" format=5.0|)
   pvalue(|colname="P Value" format=8.3|),
   AIC   (|colname="Akaike Information Criterion"|);

quit;
```

APPENDIX B

```
/* Matrix Input Using SAS PROC FACTOR for MLFA  */
data Emmett (type=corr);
infile cards missover;
_type_ = 'corr';
if _n_ = 1 then _type_ = 'N';
input _name_ $ Verbal English Arith1 Arith2 Matrix NonVerb1
       NonVerb2 Space1 Space2;
cards;
n 221
Verbal    1.000  .5232 .3950 .4706 .3455 .4262 .5761 .4338 .6393
English    .5232 1.000 .4792 .5060 .4181 .4619 .5469 .2829 .6445
Arith1     .3950 .4792 1.000 .3554 .2701 .2536 .4524 .2185 .5038
Arith2     .4706 .5060 .3554 1.000 .6906 .7909 .4427 .2852 .5050
Matrix     .3455 .4181 .2701 .6906 1.000 .6794 .3825 .1488 .4091
NonVerb1   .4262 .4619 .2536 .7909 .6794 1.000 .3721 .3138 .4721
NonVerb2   .5761 .5469 .4524 .4427 .3825 .3721 1.000 .3846 .6801
Space1     .4338 .2829 .2185 .2852 .1488 .3138 .3846 1.000 .4700
Space2     .6393 .6445 .5038 .5050 .4091 .4721 .6801 .4700 1.000
;
 proc factor n=3 method = ml;
 run;
```

APPENDIX C

```
Title1 'SAS/IML Implementation of Stability of Alienation Model';
Title2 'Cudeck, Klebe & Henly (1993)';

proc iml;
reset noname nocenter;

start sem;

/*----Function Module--------------------------------*/
start fun (x)  global (s, sigma);
  nvar=ncol(s);
  ly = (1 // x(|1|) // 0 // 0) || (0 // 0 // 1 // x[2]);
  lx = 1 // x(|3|);
  bi = inv((1||0) // (x(|4|) || 1));
  ga = x(|5:6|);
  ph = x(|7|);
  ps = i(2) # x(|8:9|);
  td = I(2) # x(|10:11|);
  te = i(4) # x(|12:15|);
  te(|3,1|)=x(|16|); te(|1,3|) = x(|16|);
 q1=ly * bi;
 syy = q1 * (ga * ph * t(ga) + ps) * t(q1) + te;
 syx = q1 * ga * ph * t(lx);
```

APPENDIX C (Continued)

```
 sxx = lx * ph * t(lx) + td;
 sigma = (syy || syx) // (t(syx) || sxx);
 invsig = inv(sigma);
 f = log(det(sigma)) + trace(s * invsig) - log(det(s)) - nvar;
 return(f);
finish;

/*---- Starting Values for Iteration ------------------*/
 x = {1, 1, 5, -.6, -.6, -.2, 7, 4, 4, 3,   3, 5, 5, 5, 5, 2};

/*---- Set up Options and call the subroutines ----------*/

 optn = {0 0 . 2};
 call nlpqn(rc, xr, "fun", x, optn);
 call nlpfdd(f, grad, hess, "fun", xr);
/*---compute information matrix and the LR test ---*/
   infomat = (2 / (n-1))* inv(hess);
   se = sqrt (vecdiag (infomat));
   chisq = (n - 1) * f;
   df = ncol(s) * (ncol(s) + 1) * 0.5 - npar;
   pvalue = 1 - probchi(chisq, df);

finish;

  s = { 11.834  0.000    0.000    0.000 0.000   0.000,
         6.947  9.364    0.000   0.000  0.000   0.000,
         6.819  5.091   12.532   0.000  0.000   0.000,
         4.783  5.028    7.495   9.986  0.000   0.000,
        -3.839 -3.889   -3.841  -3.625  9.610   0.000,
        -2.190 -1.883   -2.175  -1.878  3.552   4.503};
   s = s +t(s)-diag(s);
   npar = 16; n = 932;

run sem;

/*--- label and print out the results ------------------*/

parlab = {"ly 2,1" "ly4,2" "lx 2,1" "  beta" " gam 1" " gam 2"
           "  phi" "   ps1" "   ps2" " td 1" "  td 2" "te 1,1"
           "te 2,2" "te 3,3" "te 4,4" "te 3,1" };
rowlab = {"estimate" " std err"};
par = xr//(se`);
print par (|colname=parlab rowname=rowlab format=8.3|);
print    "estimated covariance matrix",
   sigma[format=8.3],,
        "Likelihood Ratio Test Statistic",
   chisq  (|colname="Chi-Square" format=10.3|)
   df     (|colname="DF"         format=5.0|)
   pvalue (|colname="Prob"       format=8.3|);
```

APPENDIX D

```
TITLE "Stability of Alienation Model";
DATA Wheaton(TYPE=COV);
   _TYPE_ = 'COV'; INPUT _NAME_ $ V1-V6;
   LABEL V1='Anomia (1967)'  V2='Powerlessness (1967)'
         V3='Anomia (1971)'  V4='Powerlessness (1971)'
         V5='Education' 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   -2.190   -1.883   -2.175   -1.878   3.552   4.503
;

proc calis cov data=Wheaton tech=nr edf=931 pall;
 Lineqs
      V1 =           F1                + E1,
      V2 =   ly21    F1                + E2,
      V3 =           F2                + E3,
      V4 =   ly42    F2                + E4,
      V5 =           F3                + E5,
      V6 =   lx21(5) F3                + E6,
      F1 = Gam1(-.6) F3                + D1,
      F2 = Beta(-.6) F1 + Gam2(-.2) F3 + D2;
 Std
      E1-E6 = te1-te4 (4 * 5)  td1-td2 (2 * 3),
      D1-D2 = Ps1-Ps2 (2 * 4.),
      F3    = Phi (7.) ;
 Cov
      E1 E3 = te31 (2);
RUN;
```

(Manuscript received May 29, 2001;
revision accepted for publication August 7, 2002.)