NBER WORKING PAPER SERIES

ESTIMATION OF ECONOMETRIC MODELS USING NONLINEAR FULL INFORMATION MAXIMUM LIKELIHOOD: PRELIMINARY COMPUTER RESULTS

David A. Belsley*
Kent D. Wall**

Working Paper No. 142

COMPUTER RESEARCH CENTER FOR ECONOMICS AND MANAGEMENT SCIENCE
National Bureau of Economic Research, Inc.
575 Technology Square
Cambridge, Massachusetts 02139

July 1976

*NBER Computer Research Center and Boston College. Research supported in part by National Science Foundation Grant SOC75-13626 to the National Bureau of Economic Research, Inc.

**NBER Computer Research Center. Research supported in part by National Science Foundation Grant SOC75-13626 to the National Bureau of Economic Research, Inc.

Abstract

This working paper provides some preliminary results on the computational feasibility of nonlinear full information maximum likelihood (NLFIML) estimation. Several of the test cases presented were also subjected to nonlinear three stage least square (NL3SLS) estimation in order to illustrate the relative performance of the two estimation techniques. In addition, certain other aspects central to practical implementation are highlighted. These include the effect of various computers on the efficiency of the code, as well as the relative merits of numerical and analytical generation of gradient information. Broadly speaking, NLFIML appears competitive in cost and superior in statistical properties to NL3SLS.

Acknowledgement

The successful completion of all the computer experiments owes much to the able research assistance of Mr. David Jones.

TABLE OF CONTENTS

1. INTRODUCTION

The potential benefits of nonlinear full information maximum likelihood (NLFIML) estimation in econometric modeling are many and enviable. Notable among these are its inherent flexibility and asymptotic consistency and efficiency. These advantages of NLFIML have, to date, been more than offset by its computational burdens. In the general nonlinear case the computational requirements are often considered extreme - indeed prohibitive. However, recent advances in modern digital computers and programming techniques appear to have mitigated this disadvantage. The significant increase in capability of third generation computers, the availability of optimizing compilers (such as the FORTRAN H compiler), and programming methodology which borders on the frontiers of the state-of-art can now be combined to alleviate (if not eliminate) the drawbacks of NLFIML estimation.

An awareness of present-day technology and methodology has guided the National Bureau of Economic Research, Computer Research Center, in its specification and construction of a NLFIML facility. The proof of the pudding, is, of course, in the eating; and to this end preliminary testing of the resulting facility has been undertaken. The results of these first tests, which are in part to be interpreted as a progress report on our approach to FIML estimation, we feel lend credence to the claim that there is indeed hope for NLFIML. NLFIML (its practical usefulness with an acceptable computational cost) appears to be feasible.

2. THE NLFIML ESTIMATOR

Full Information Maximum Likelihood estimates multiequation econometric models can be obtained by determining that value of the parameter vector, θ, which maximizes the concentrated log-likelihood function. This function may be derived as (see Eisenpress and Greenstadt [1966], Chow [1973]),

$$\ell^*(\theta) = \text{const.} - \frac{T}{2} \ln(\det \hat{S}) + \sum_{t=1}^{T} \ln(|\det J_t|) \tag{1}$$

where $\hat{S}$ is the estimated residual variance-covariance matrix:

$$\hat{S} = \hat{S}(\theta) = \frac{1}{T} \sum_{t=1}^{T} u_t(\theta)u_t'(\theta) \ , \tag{2}$$

and J_t is the Jacobian matrix evaluated at time t:

$$J_t = J_t(\theta) = \frac{\partial f_t(y_t,z_t,\theta)}{\partial \theta} \tag{3}$$

The generally nonlinear vector function $f_t(y_t,z_t,\theta)$ represents the model specification, and is related to the residuals, by definition, according to

$$f_t(y_t,z_t,\theta) = u_t \ . \tag{4}$$

Both f_t and u_t are G×1 vectors,

$$f_t = [f_{1t}, f_{2t} ,\ldots, f_{gt} ,\ldots, f_{Gt}]'$$
$$u_t = [u_{1t}, u_{2t} ,\ldots, u_{gt} ,\ldots, u_{gt}]' \ ,$$

representing a G equation model for each t. The G×T vector y_t contains the endogenous variables at time t, while the K×1 vector z_t represents all the predetermined variables of the model at time t. The sequences of random vectors $\{u_t \ ; \ 1 \leq t \leq T\}$ are assumed to form a jointly normal, zero mean stochastic process, independent over t. Therefore,

$$u_t \sim N(0,S)$$

The actual implementation of the NLFIML involves the estimation of θ via the solution to an equivalent nonlinear function <u>minimization</u> problem: It has proven more expedient to determine the estimate of θ as that value which minimizes the scaled negative log likelihood function,

$$(5) \qquad V(\theta) = \ell n(\det \hat{S}) - \frac{2}{T} \sum_{t=1}^{T} \ell n(|\det J_t|).$$

Most of the test models described below also were estimated using a version of nonlinear three-stage least squares (NL3SLS) due to Jorgenson and Laffont [1974]. This provided comparative information regarding the speed and computational demands of NLFIML estimation. Without indulging in all the details, (for which see the article cited), the Jorgenson-Laffont NL3SLS estimator is that value of θ which minimizes

$$(6) \qquad V(\theta) = F'(\theta)\,[\hat{S}^{-1} \otimes X(X'X)^{-1}X']\,F(\theta)$$

where,

$$F'(\theta) = [f_1(y_1,z_1,\theta),\ f_1(y_2,z_2,\theta)\ ,\ldots,\ f_G(y_1,z_1,\theta)\ ,\ldots\ f_G(y_T,z_T,\theta)]'$$

and X is a matrix of exogenous variables which need not include all the elements of z_t given in the original model specification. $\hat{S}$ is any consistent estimator of S.

The definitions for $V(\theta)$, given by (5) and (6), illustrate how, from a computational point of view, non-linear simultaneous-equation estimation reduces to a problem in function minimization - the only difference being the precise definition of the function to be minimized. Given $V(\theta)$, any suitable function minimization algorithm can be employed to compute the optimal θ.

3. NONLINEAR FUNCTION MINIMIZATION

The particular function minimization algorithm employed in this work is that due to Davidon [1959] as refined by Fletcher and Powell [1963], and popularly referred to as the Davidon-Fletcher-Powell (DFP) algorithm.[1] There are certainly several applicable algorithms; however, the DFP has so far proved superior in all around performance. A detailed discussion of the DFP algorithm is unnecessary here, but a brief account of the computations will aid in understanding the results of Section 5.

Given an initial guess, θ^o, the DFP algorithm proceeds by generating a sequence of continually improving estimates $\{\theta^k;\ k = 1,2,\ldots\}$ according to:[2]

(7) $$\theta^{k+1} = \theta^k + \partial\theta^k = \theta^k + \alpha^k H(\theta^k) g^k(\theta^k)$$

α^k = a scalar quantity, the optimal stepsize along the conjugate gradient search direction

$H(\theta^k)$ = a matrix, the k^{th} approximation to the inverse Hessian of $V(\theta)$.

$g(\theta^k)$ = a vector, the gradient of $V(\theta)$ evaluated at $\theta = \theta^k$.

The optimal step α^k in this algorithm is determined by cubic interpolation of $V(\theta)$ along the current search direction ($=H(\theta^k)\ g(\theta^k)$). The H matrix is generated automatically within the algorithm. The gradient vector g may be generated by simple finite differencing, but the implicit-function-differentiation capability of the TROLL system has been employed in order automatically to generate the analytic expressions to allow $g(\theta^k)$ to be computed exactly.

The gradient expressions for NLFIML estimation follow from the fact that the i^{th} element of $g(\theta)$ can be written as

$$\text{(8)} \qquad \frac{\partial V(\theta)}{\partial \theta_i} = \frac{2}{T} \sum_{t=1}^{T} \left[\sum_{g \varepsilon p_i} u_t' \, s^g \, \frac{\partial u_{gt}}{\partial \theta_i}k - \sum_{g \varepsilon p} \gamma_t^g \frac{\partial^2 u_{gt}}{\partial \theta_i \partial y_t} \right]$$

where

p_i = set of equation numbers in which θ_i appears,

s^g = g^{th} column of $\hat{S}^{-1}$,

γ_t^g = g^{th} row of J_t^{-1}.

The partial derivatives $\partial u_{gy}/\partial\theta_i$ and $\partial^2 u_{gt}/\partial\theta_i\partial y_t$ are computed using the definitions for g^{th} element of u_t given in (4). At run time the differentiation routine is called to scan (4) and generate the code for the required derivatives. This code is then compiled and saved for the actual execution of the program. Therefore, during estimation, gradients can be computed very rapidly. Note that the additional cost for obtaining exact gradients is minimal since the inverses of $\hat{S}$ and J_t effectively have been computed during the evaluation of the determinants in $V(\theta)$.

The generation of exact gradient expressions for NL3SLS follows in a completely analogous way once it is realized that, in this case,

$$\text{(9)} \qquad \frac{\partial V(\theta)}{\partial \theta_i} = 2 \left[\frac{\partial F(\theta)}{\partial \theta_i}\right]' \left[\hat{S}^{-1} \otimes X(X'X)^{-1}X'\right] F(\theta) \; .$$

The process of improving θ^k according to (7) continues until convergence is obtained. Convergence occurs whenever the predicted change in θ^k is less than some prespecified error tolerance. Thus, if

$$\text{(10)} \qquad |\delta\theta_i^k| < \varepsilon_i \; ,$$

the iterations terminate, and the current estimate θ^k is accepted as optimal. The user supplies the vector of error tolerances ε. Other, more practical, convergence criteria are currently under study. (See the discussion of the results in Section 5.)

Since there is no prior guarantee that convergenced will be achieved, the improvement process automatically terminates after a prespecified number of iterations have been reached.

4. THE TEST MODELS

In order to obtain as much information as possible concerning computational performance six test problems were considered ranging from the estimation of small, strictly linear models to large, nonlinear models. Each model is presented in detail in the Appendix, but a brief discussion of their essential characteristics is given here in order to provide the proper perspective for the discussion of computer results contained in Section 5.

Model 1

The first test problem involves the estimation of a small, three-equation model with a total of 10 unknown parameters. The model is linear in both the parameters and the variables, and no lagged endogenous variables appear. The exogenous variables are taken from actual economic time series, while the endogenous variates are simulated data generated by the model with known parameters and error structure. This model, then, exemplifies the simplest and cleanest class of estimation problems one can consider.

Model 2

The second test problem is a small two-equation model with five parameters.[3] It is highly nonlinear in the parameters and contains across-equation parameter constraints. No lagged endogenous variables appear and no special assumptions are made on the error structure. Although small in size, this model highlights the problems faced in nonlinear estimation.

Model 3

Case 3 is a six equation, 22 parameter sub-model of GNP components considered by Fair [1973]. The model is linear in both variates and parameters. The exogenous variables are taken directly from Fair, while the endogenous data are generated through simulation of Fair's model based on the assumption that his parameter estimates are true. This case represents a significant increase in the size (both in number of equations and parameters) of the estimation problem, thereby constituting a much more realistic (i.e., difficult) function minimization problem.

Model 4

This case duplicates Model 3 except that the generating error structure contains first-order autocorrelation. Accordingly, in estimation, a Cochrane-Orcutt transformation has been employed in specifying the model. This introduces a multiplicative form of nonlinearity and six additonal parameters, bringing the number of parameters to 28. The model thus constitutes a mildly nonlinear estimation problem (as compared to the severe nonlinearities in Model 2) of moderate size.

Model 5

The fifth test problem deals with another variant of Model 3. Actual economic data, rather than simulated data are used for the endogenous variables, and dummy variables are included in many of the equations to take account of strikes. In fact, this specification is precisely that of the money-GNP sector of the model used by Fair [1973]. There are now 38 unknowns to estimate including the autoregressive parameters of the Cochrane-Orcutt transformation.

Model 6

The last estimation problem uses the entire Fair [1973] model of 16 equations and 61 parameters. Actual economic series are used for all the data. This problem is most closely representative of a realistic economic estimation problem - both in terms of size and nonlinearity.

5. THE RESULTS

The preceeding models are all estimated by the NLFIML facility within GREMLIN, and whenever possible, the same models are also estimated by NL3SLS and ERSF, another program capable of maximum-likelihood estimation. ERSF, however, does not permit for the estimation of models nonlinear in the variables and therefore could not be employed with Models 2 and 6. Nonetheless, its inclusion is of interest because for linear models both ERSF and NLFIML employ the same computations except in the way the gradients are determined: ERSF uses first finite-differences of $V(\theta)$ while NLFIML employs the exact analytically generated gradient expressions. Thus, it should be possible to determine whether analytic or numeric differentiation is the more efficient for this use. Table 1 summarizes the results.

TABLE 1. Convergence and Timing Summary: All Models

	Convergence	No. of iterations	No of function evaluations	Times in mins:sec Total	Times in mins:sec Virtual	Compiler	Starts
<u>Model 1</u>	(3 equation, 10 parameter, linear)						
NLFIML	yes	19	51	:14	NA	G	true
ERSF	yes	31	63	:18	:15	G	true
NL3SLS	yes	11	22	:06	:03	H	OLS
NLFIML	yes	31	69	:13	:08	H	OLS
<u>Model 2</u>	(2 equation, 5 parameter, nonlinear)						
NLFIML	yes	16	27	:18	:16	H	OLS
NL3SLS	yes	11	22	:09	:07	H	OLS
<u>Model 3</u>	(6 equation, 22 parameter, linear)						
NLFIML	yes	36	105	1:18	1:17	G	OLS
ERSF	no	64	NA	4:13	3:53	G	OLS
NL3SLS	yes	23	23	:36	:25	H	OLS
NLFIML	yes	36	105	:35	:32	H	OLS
<u>Model 4</u>	(6 equation, 28 parameter, nonlinear)						
NLFIML	yes	134	288	7:54	7:47	G	OLS
NLFIML	yes	134	288	1:49	1:42	H	OLS
NL3SLS	yes	43	123	:35	:25	H	OLS
<u>Model 5</u>	(6 equation, 38* parameter, nonlinear)						
NLFIML	yes	99	214	1:26	1:19	H	Huber
<u>Model 6</u>	(16 equation, 61 parameter, nonlinear)						
NLFIML	yes?	72	144	14:08	13:32	H	Fair

NA indicates quantity not available.

*Because of the observation interval employed, only 4 of the additional 10 coefficients were estimated together with the 28 previous coefficients.

Several points of interpretation need to be made:

1. Different starting values are used for the maximum likelihood estimators. OLS, of course, refers to ordinary least squares. TRUE refers to the true parameter values that are used in the generation of simulated data and are only employed in Model 1. Other names refer to specific estimates of parameters from other studies.

2. Compiler. Different compilers were used in the development of these packages, and they clearly result in differing degrees of computational speed. The H compiler is designed for fast "number crunching" and separate entries were included in order to preserve a proper comparison between NLFIML and ERSF which is compiled with the G-level compiler.

3. CPU Time. The total time is relevant to billing, but more meaningful comparisons of the relative computations efficiencies results from the use of the virtual figures. These figures refer to the IBM 370-168.

4. Number of iterations: comparison between NL3SLS and NLFIML is best made on the basis of virtual CPU time. The number of iterations for NL3SLS is that required <u>after</u> the preliminary NL2SLS estimation has been run on each equation separately. Number of iterations therefore does not indicate the full effort involved for NL3SLS relative to NLFIML.

5. Convergence. The unqualified "yes" indicates that convergence is obtained using the stringent gradient criterion utilized by the version of DFP currently employed, namely, stop the iterations if

$$|\partial\theta_i^k| \leq \varepsilon_i$$

for all elements of θ, where $\varepsilon_i = \varepsilon = 10^{-3}$. Convergence by a weaker but meaningful criterion usually occurs significantly earlier. This weaker criterion uses the computed elasticity of the likelihood function with respect to each element of θ. In essence, convergence is obtained whenever

$$\left| \frac{\theta_i^k \, g_i(\theta^k)}{V(\theta^k)} \right| \leq \varepsilon$$

for all i. The qualified "yes" (denoted by the presence of a question mark) indicates that a strict criterion was not met, but that the computed elasticities are all less than 0.01.

Model 1. As expected, the first model posed no problem for any of the estimation methods. Convergence is obtained in each run, and computation times are all comparable except in the case of ERSF. Inaccurate gradient information, due to numerical approximation, caused ERSF to take many more iterations. It should be mentioned, however, that this phenomenon only arises near the solution point. Further, from the solution (where the gradients are larger and the percent error of the numerical gradient computation is smaller) both NLFIML and ERSF behaved quite similarly. The small size of the problem minimizes the advantages of the optimizing H compiler. With a problem of this size and complexity there do not appear to be any computational reasons for preferring one method to another. Table 2 presents the parameter estimates and true values for comparison.

Model 2. The highly nonlinear nature of this model failed to produce any real obstacles for either NLFIML or NL3SLS, both converging with little difficulty. Table 3 presents these estimated coefficients. There did appear, however, some difference in the computational costs with NLFIML requiring a

TABLE 2[a]

MODEL 1: True Parameter Values and Parameter Estimates

Coefficient	True Value	FIML Estimate[b]	Estimated Standard Errors[c]	
A1	10.00	12.1590	2.2175	2.2254
A2	0.15	0.0366	0.1304	0.1311
A3	0.70	0.7255	0.0827	0.0833
A4	5.00	4.8422	2.0584	2.0634
A5	0.60	0.6117	0.0452	0.0454
A6	-.20	-.0940	0.1325	0.1322
A7	-.25	-.2820	0.0437	0.0430
A8	-12.00	-12.7526	3.4915	3.4763
A9	0.30	0.3136	0.0769	0.0767
A10	0.40	0.4299	0.0491	0.0485

a. Estimation was over the period 1960(I) through 1972(IV).

b. The same final estimate was obtained whether using the true values or the OLS estimates as the initial guess.

c. The first column contains the standard errors using the true values as the initial guess while the second column gives the estimates standard errors resulting from OLS estimates as the starting values.

TABLE 3[a]

MODEL 2: NLFIML and NL3SLS Estimates

Coefficient	NL3SLS Estimates[b]	FIML Estimates[b]
C1	0.6364(0.0163)	0.5839(0.0163)
C2	0.0055(0.0008)	0.0058(0.0006)
C3	1.3665(0.1267)	1.3618(0.0911)
C4	-0.1248(0.1622)	0.4749(0.1891)
C5	0.3160(0.0328)	0.4470(0.0434)

a. Estimated using annual series from 1 to 41.

b. Estimates standard errors are given in parentheses.

little more than twice the time and twice the number of iterations than for NL3SLS. At face value it would appear that the latter has much to offer in terms of costs. This advantage must, however, be balanced with the better "quality" of the NLFIML estimates. First, the NLFIML estimates converged to exactly the same estimates obtained by Dr. Eisenpress using a full Newton-Raphson algorithm with analytically derived second partial derivatives.* The NL3SLS estimates are significantly different, especially in regard to C1, C4 (given the wrong sign by NL3SLS), and C5. Second, when compared to NLFIML, the NL3SLS estimates proved to be suboptimal both in terms of the negative log-likelihood function (-11.08 compared to -10.49) and its gradient. The gradient of the negative log-likelihood, evaluated at the NL3SLS estimates is,

$$\frac{\partial V}{\partial \theta}\Big|_{3SLS} = [0.093, -130.3, -1.46, -2.11, -1.66]',$$

while that associated with the NLFIML estimates is

$$\frac{\partial V}{\partial \theta}\Big|_{FIML} = [2.6 \times 10^{-5}, 1.0 \times 10^{-4}, 1.76 \times 10^{-6}, 2.8 \times 10^{-5}, -1.1 \times 10^{-4}]$$

This quality difference between 3SLS and FIML is more important in the nonlinear cases where 3SLS does not share FIML's asymptotic efficiency.

Model 3. Convergence proved a problem for ERSF, the routine using numerically generated gradients. The significant improvement in computation time resulting from the use of the H compiler is now quite evident, computation time being reduced by approximately 30% for NLFIML. This moderate-size, linear problem does not appear to provide any evidence for preferring NL3SLS

*While analytic first partial derivatives of $V(\theta)$ are used in NLFIML, only numerical approximations to $\partial^2 V/\partial\theta_i\partial\theta_j$ are employed in the DFP algorithm.

over NLFIML from a cost viewpoint - both methods take almost exactly the same time to arrive at a solution. Table 4 provides a comparison between the true coefficients and the NLFIML solution.

TABLE 4

MODEL 3: (a) True Coefficient Values

B11	-41.272	B12	0.109	B13	0.217
B14	0.102	B21	0.0547	B22	0.78
B23	0.07	B31	0.029	B32	0.915
B33	-0.023	B41	-7.7	B42	0.073
B43	0.515	B51	-2.476	B52	0.014
B53	0.055	B54	0.0878	B55	0.018
B61	-166.57	B62	0.557	B63	-0.521
B64	0.473				

(b) NLFIML Coefficient Estimates

COEF	VALUE	STD ERR	T-STAT	COEF	VALUE	STD ERR	T-STAT
B11	-40.7169	3.75491	-10.8436	B12	0.109797	0.00143	76.8039
B13	0.299665	0.058797	5.09659	B14	0.008743	0.062358	0.140204
B21	0.073768	0.012028	6.13309	B22	0.688904	0.051339	13.4188
B23	0.116294	0.018679	6.2261	B31	0.02811	0.006755	4.16135
B32	0.920227	0.025746	35.7419	B33	-0.026541	0.005062	-5.2434
B41	-6.7889	1.39416	-4.86953	B42	0.069682	0.006518	10.6911
B43	0.542429	0.067734	8.0082	B51	-3.12944	0.68493	-4.569
B52	0.01432	0.000463	30.9293	B53	0.067092	0.008458	7.9325
B54	0.056347	0.011978	4.70399	B55	0.040887	0.008347	4.89833
B61	-1.603E+02	11.5798	-13.8461	B62	0.464845	0.096369	4.82357
B63	-0.501609	0.035396	-14.1755	B64	0.527378	0.111714	4.7208

Model 4. Both NLFIML and NL3SLS obtained unqualified convergence, however, a difference in computation time is now evident between the two methods. The effect of the H compiler is also extremely pronounced in this case, improving the computational efficiency of NLFIML by approximately a factor of four. The computational advantage of NL3SLS (35 sec. total CPU time versus 109 sec total CPU time for NLFIML) is not overwhelming, and reference to Table 5 reveals some interesting characteristics concerning the quality of the three-stage estimates when measured against the NLFIML

TABLE 5A

MODEL 4: Likelihood Function and Gradients Evaluated at NL3SLS Estimates

ITERATION: 0 1 FCN EVALS FCN = 1.59585
GNORM = 24.8119

Coef	Value	Grad	Std. Err.	T-Stat
B11	-56.7122	-0.004355	9.51761	-5.95866
B12	0.11946	-0.518696	0.008252	14.4756
B13	0.282167	-0.511326	0.061576	4.5824
B14	0.1469	-0.027318	0.066011	2.22537
B21	0.129674	0.02415	0.015937	8.13655
B22	-0.07158	0.531824	0.103792	-0.68965
B23	0.189521	0.910057	0.064095	2.95686
B31	0.028855	-21.9063	0.001931	14.9435
B32	0.921487	-8.11427	0.009323	98.8385
B33	-0.032713	-4.33417	0.008348	-3.91874
B41	-17.3008	-0.005176	4.48739	-3.85543
B42	0.123908	2.46052	0.01627	7.61585
B43	0.219958	-0.356135	0.08813	2.49585
B51	3.18432	-0.006878	3.65055	0.872284
B52	0.005083	-4.45746	0.005886	0.863427
B53	0.066256	2.31987	0.009077	7.29893
B54	0.063322	2.49576	0.009361	6.76468
B55	0.02656	-0.02603	0.009331	2.84646
B61	-106.874	-0.002323	28.0256	-3.81343
B62	0.325044	0.257067	0.149754	2.17053
B63	-0.464352	-0.761012	0.064611	-7.18693
B64	0.283967	-0.045716	0.143323	1.9813
R1	0.676472	-0.080584	0.088928	7.60701
R2	0.898041	-0.245155	0.036053	24.909
R3	0.073135	0.172537	0.096324	0.759257
R4	0.661902	0.041769	0.08026	8.24694
R5	0.851894	-0.739133	0.037609	22.6512
R6	0.854066	-2.53483	0.033861	25.2224

TABLE 5B

MODEL 4: Likelihood and Gradients at NLFIML Solution

ITERATION: 0 1 FCN EVALS FCN = 1.32101
GNORM = 0.024121

Coef	Value	Grad	Std. Err.	T-Stat
B11	-55.6078	-2.629088E-06	8.87677	-6.26442
B12	0.116414	-0.001349	0.007869	14.7931
B13	0.290481	-0.000244	0.05934	4.89524
B14	0.144435	-0.000245	0.063472	2.27557
B21	0.160488	-0.00055	0.020529	7.8178
B22	-0.268705	-7.526274E-05	0.107083	-2.50932
B23	0.071895	-8.196232E-05	0.077345	0.929537
B31	0.029194	-0.022774	0.001797	16.2498
B32	0.916936	-0.006293	0.008616	106.425
B33	-0.027558	-0.004201	0.007874	-3.49998
B41	-7.4318	-2.568159E-06	6.93669	-1.07137
B42	0.082072	-0.001348	0.021534	3.81127
B43	0.435165	-0.000147	0.105942	4.10757
R51	3.74985	-2.300765E-06	3.46911	1.08092
R52	0.005064	-0.001206	0.005983	0.846286
B53	0.056254	-0.000273	0.008634	6.51575
B54	0.06387	-0.000273	0.00922	6.92703
B55	0.033041	-0.000261	0.00944	3.50002
B61	-93.011	-1.263592E-06	23.1194	-4.02307
B62	0.204178	-0.000162	0.148204	1.37768
B63	-0.392424	6.388946E-05	0.046676	-8.40734
B64	0.344255	-0.000159	0.108349	3.17728
R1	0.649792	-1.673411E-06	0.084132	7.72347
R2	0.943569	-0.00033	0.043264	21.8098
R3	-0.037683	-4.277731E-06	0.099158	-0.380027
R4	0.719822	5.960077E-06	0.082967	8.67602
R5	0.839265	-1.585828E-05	0.038928	21.5595
R6	0.836636	-6.237008E-05	0.040259	20.7811

results. First, note that the NL3SLS estimates produced a scaled log-likelihood value of 1.595 as opposed to a value of 1.321 for NLFIML. Next, note that the gradients of the likelihood function, when evaluated at the NL3SLS estimates, are considerably larger than those associated with the NLFIML estimates. In particular, the norm of the gradient for the NL3SLS estimator is 24.8119, while that of the NLFIML is 0.02412. The suboptimality of the NL3SLS estimates, judged in terms of the NLFIML loss function and the asymptotic efficiency of NLFIML, does seem to detract from the computational value of NL3SLS. Both estimates must, however, be subjected to further analysis, including their prediction capabilities, before a final verdict can be reached regarding the overall advantage of one method.

Model 5. The addition of 4 dummy variable coefficients to the problem did not cause any significant problems for FIML. Nor did any difficulty arise because of the use of real data. Convergence was obtained in 99 iterations, requiring a total of only 79 seconds of virtual machine time. The scaled negative log-likelihood was reduced from a value of 0.565 to 0.420. The norm of the gradient at the final iteration was 0.0355. The final estimates together with their gradients are contained in Table 6.

Model 6. The entire Fair model, using actual data over the interval 1960(II) to 1973(I), achieved convergence in 72 iterations when the convergence criterion in (10) was relaxed by increasing ε_i from 10^{-4} to 10^{-3}. It is interesting to note, however, that convergence would have been achieved at the first iteration using the elasticity criterion $\max|e_i| \leq 0.1$! The initial loss function, using the Fair estimates, was computed at 36.21; at convergence its value had been reduced to 36.2066. Since the initial gradient norm was 56.333, while that at iteration 72 was 0.009, it is obvious that most of the

TABLE 6[a]

MODEL 5: Final Estimates and Gradients

Coef	Value	Grad
C31	-26.1935	-1.236810E-05
C32	0.107512	-0.009014
C33	0.118663	-0.001151
C34	0.053564	-0.001159
C41	0.043608	0.026933
C42	0.835926	0.007341
C43	0.039562	0.003475
C51	0.010347	-0.000285
C52	0.514176	7.267064E-06
C53	-0.006807	-0.000486
C61	-4.44574	-1.165192E-06
C62	0.075997	-0.000968
C63	0.421463	-7.436336E-05
C71	-4.32556	1.077380E-05
C72	0.016043	0.008187
C73	0.051321	0.00122
C74	0.087822	0.00121
C75	0.030281	0.001215
C81	-95.405	6.489034E-07
C82	0.648711	0.000185
C83	-0.350476	0.000152
C84	0.136498	-2.742165E-06
R3	0.452277	-3.345026E-06
R4	-0.263294	-1.550233E-07
R5	1.01869	0.017828
R6	0.861498	-2.835667E-06
R7	0.587505	5.713341E-07
R8	0.795925	1.057235E-06
C35	-1.94084	-2.799074E-07
C36	2.95738	-2.380445E-07
C85	-1.47457	2.030344E-08
C86	3.28608	-2.582679E-09

a. Estimated over the range 1959(IV) to 1972(IV). Therefore C37, C38, C64, C65, C87, and C88 were inactive and not estimated.

iterations were spent in minor adjustments to θ in order to reduce the gradient components to nearly zero. Practically speaking, it appears that the initial guess could have been accepted as the final solution - as indicated by the elasticity criterion and the relative change in the loss function.

6. CONCLUSIONS

The results summarized in Table 1, while revealing nothing startling concerning the effect of model size and starting values, indicate that NLFIML estimation is not nearly as costly in relation to 3SLS as is commonly thought. Indeed, for small estimation problems (Models 1 and 2) cost differences are negligible. Furthermore, in Model 2 NL3SLS produced an estimate for C4 that was of incorrect sign. For larger problems (Models 3,4, and 5), NLFIML always achieved convergence in under two minutes (or approximately $36 on the Cornell IBM370/158 computer). In cases where NL3SLS showed a noticeable savings in computation time (such as the larger nonlinear models like Model 4), such savings should be weighed against the loss in statistical efficiency that characterizes minimum distance relative to maximum likelihood in the nonlinear case.

Other conclusions of importance in terms of software development can also be drawn from Table 1. In particular, numerically generated gradients, i.e., resulting from finite differencing of the loss function, tend to degrade the performance of the optimization phase, especially near the solution where the percentage error in the differencing dominates the true value of the gradient. Analytically generated gradient expressions, while requiring more programming and debugging, offer such improved performance so as to far outweigh the initially high development costs. The importance of employing a sophisticated "optimizing" compiler (such as the FORTRAN-H compiler) is also emphasized in Table 1. The larger and more complex the problem, the greater are the savings in computation time over using the standard FORTRAN-G compiler.

Finally, it should be emphasized that even greater savings could be achieved by employment of more realistic and practical convergence criteria in determining when a solution to maximizing the likelihood function has been reached. Use of the elasticity criterion in Models 4,5, and 6 would have reduced the number of required iterations by 10% to 25%, thus further enhancing the attractiveness of NLFIML.

FOOTNOTES

1. The interested reader is referred to Dennis and More [1976] for an excellent discussion and review of nonlinear function minimization algorithms.

2. Any sequence $\{\theta^k; 1,2,\ldots\}$ for which $V(\theta^o) > V(\theta^1) > V(\theta^2) > \ldots$ is termed a continually improving sequence.

3. We are indebted to Dr. Harry Eisenpress of IBM for supplying this model and data.

REFERENCES

Chow, G. [1973] "On the Computation of Full-Information Maximum Likelihood Estimates for Nonlinear Equation Systems, The Rev. of Econ. Stats., Vol. LV, (February).

Davidon, W.C. [1958] "Variable Metric Method for Minimization, ANL-5990, AEC Res. Devel. Report.

Dennis, J. and Moŕe, J. [1976] "Quasi-Newton Methods, Motivation, and Theory", SIAM Rev. (forthcoming).

Eisenpress, H. and Greenstadt, J. [1966] "The Estimation of Nonlinear Econometric Systems," Econometrica, Vol. 34, no. 4, 6 (October).

Fair, R.C. [1973] "A Comparison of FIML and Robust Estimates of a Nonlinear Macroeconomic Model", NBER Working Paper Series, No. 15, (October).

Fletcher, R. and Powell, M.J.D. [1963] "A Rapidly Convergent Descent Method for Minimization, Comp. J. Vol. 6.

Jorgenson, D. and Jean-Jacques Laffont [1974], "Efficient Estimation of Non-linear Simultaneous Equations with Additive Disturbances", *Annals of Economic and Social Measurement*, Vol. 3, No. 4, October.

Appendix A. Models Used in Evaluating NLFIML and NL3SLS

Model 1

ENDOGENOUS:
Y1 Y2 Y3

EXOGENOUS:
X1 X2 X3

COEFFICIENT:
A1 A10 A2 A3 A4 A5 A6 A7 A8 A9

EQUATIONS

```
1:    Y1 = A1+A2*Y2+A3*X1
2:    Y2 = A4+A5*Y1+A6*Y3+A7*X2
3:    Y3 = A8+A9*Y1+A10*X3
```

Model 2

ENDOGENOUS:
Y1 Y2

EXOGENOUS:
Y11 Y3 Y9

COEFFICIENT:
C1 C2 C3 C4 C5

EQUATIONS

```
1:    C1*10**(C2*Y11)*(C5*Y1**(-C4)+(1-C5)*Y2**(-C4))**((-C3)/C4) = Y3
2:    (Y1/Y2)**(-C4-1)*(C5/(1-C5)) = Y9
```

Model 3

ENDOGENOUS:
CD CN CS IH IP V

DEFINITION:
GNP

EXOGENOUS:
CD.1 CN.1 CS.1 EX G HSQ HSQ.1 HSQ.2 IMP MOOD.1 MOOD.2 PE2 UNIT V.1

<u>Model 3</u>: (Con't)

POLICY:
MOOD

COEFFICIENT:
B11 B12 B13 B14 B21 B22 B23 B31 B32 B33 B41 B42 B43 B51 B52
B53 B54 B55 B61 B62 B63 B64

EQUATIONS

```
1:   GNP = = CD+CN+CS+IP+IH+DEL(1 : V)+(EX-IMP+G)
2:   CD = B11*B12*GNP+B13*MOOD(-1)+B14*MOOD(-2)
3:   CN = B21*GNP+B22*CN(-1)+B23*MOOD(-2)
4:   CS = B31*GNP+B32*CS(-1)+B33*MOOD(-2)
5:   IP = B41+B42*GNP+B43*PE2
6:   IH = B51+B52*GNP+B53*HSQ+B54*HSQ(-1)+B55*HSQ(-2)
7:   DEL(1 : V) = B61+B62*(CD+CN)+B63*V(-1)+B64*(CD(-1)+CN(-1))
```

<u>Model 4</u>

ENDOGENOUS:
CD CN CS IH IP V

DEFINITION:
GNP

EXOGENOUS:
EX G HSQ IMP MOOD PE2 UNIT

COEFFICIENT:
B11 B12 B13 B14 B21 B22 B23 B31 B32 B33 B41 B42 B43 B51 B52 B53
B54 B55 B61 B62 B63 B64 R1 R2 R3 R4 R5 R6

EQUATIONS

```
1:   GNP == CD+CN+CS+IP+IH+DEL(1 : V)+(EX-IMP+G)

2:   CD = R1*CD(-1)+B11*(1-R1)+B12*(GNP-R1*GNP(-1))+B13*(MOOD(-1)-R1*MOOD(-2))
          +B14*(MOOD(-2)-R1*MOOD(-3))

3:   CN = R2*CN(-1)+B21*(GNP-R2*GNP(-1))+B22*(CN(-1)-R2*CN(-2))
          +B23*(MOOD(-2)-R2*MOOD(-3))

4.   CS = R3*CS(-1)+B31*(GNP-R3*GNP(-1))+B32*(CS(-1)-R3*CS(-2))
          +B33*(MOOD(-2)-R3*MOOD(-3))

5.   IP = R4*IP(-1)+B41*(1-R4)+B42*(GNP-R4*GNP(-1)+B43*(PE2-R4*PE2(-1))

6.   IH = R5*IH(-1)+B51*(1-R5)+B52+(GNP-R5*GNP(-1))+B53*(HSQ-R5*HSQ(-1))
          +B54*(HSQ(-1)-R5*HSQ(-2))+B55*(HSQ(-2)-R5*HSQ(-3))

7.   DEL(1 : V) = R6*DEL(1 : V(-1))+B61*(1-R6)+B62*(CD+CN-R6*(CD(-1)+CN(-1)))
          +B63*(V(-1)-R6*V(-2))+B64*(CD(-1)+CN(-1)-R6*(CD(-2)+CN(-2)))
```

Model 5

ENDOGENOUS:
CD CN CS IH IP V

DEFINITION:
GNP

EXOGENOUS:
D644 D651 D704 D711 EX G HSQ IMP MOOD PE2

COEFFICIENT:
C31 C32 C33 C34 C35 C36 C37 C38 C41 C42 C43 C51 C52 C53 C61
C62 C63 C64 C65 C71 C72 C73 C74 C75 C81 C82 C83 C84 C85 C86
C87 C88 R3 R4 R5 R6 R7 R8

EQUATIONS

```
1:   GNP == CD+CN+CS+IP+IH+DEL(1 : V)-IMP+EX+G
2:   CD = R3*CD(-1)+C31*(1-R3)+C32*(GNP-R3*GNP(-1))+C33*(MOOD(-1)-R3*MOOD(-2))
          +C34*(MOOD(-2)-R3*MOOD(-3))+C35*(D644-R3*D644(-1))+C36*(D651-R3*D651(-1)
          +C37*(D704-R3*D704(-1))+C38*(D711-R3*D711(-1))
3:   CN = R4*CN(-1)+C41*(GNP-R4*GNP(-1))+C42*(CN(-1)-R4*CN(-2))
          +C43*(MOOD(-2)-R4*MOOD(-3))
4:   CS = R5*CS(-1)+C51*(GNP-R5*GNP(-1))+C52*(CS(-1)-R5*CS(-2))
          +C53*(MOOD(-2)-R5*MOOD(-3))
5:   IP = R6*IP(-1)+C61*(1-R6)+C62*(GNP-R6*GNP(-1))+C63*(PE2-R6*PE2(-1))
          +C64*(D704-R6*D704(-1))+C65*(D711-R6*D711(-1))
6:   IH = R7*IH(-1)+C71*(1-R7)+C72*(GNP-R7*GNP(-1))+C73*(HSQ-R7*HSQ(-1))
          +C74*(HSQ(-1)-R7*HSQ(-2))+C75*(HSQ(-2)-R7*HSQ(-3))
7:   DEL(1 : V) = R8*DEL(1 : V(-1))+C81*(1-R8)+C82*(CD(-1)+CN(-1)-R8*(CD(-2)
          +CN(-2)))+C83*(V(-1)-R8*V(-2))+(-C84)*(DEL(1 : CD+CN)-R8*DEL(1 : CD(-1)
          +CN(-1)))+C85*(D644-R8*D644(-1))+C86*(D651-R8*D651(-1))+C87*
          (D704-R8*D704(-1))+C88*(D711-R8*D711(-1))
```

Model 6

ENDOGENOUS:
CD CN CS D IH IP LF1 LF2 M PD V

DEFINITION:
E GAP2 GNP GNPR MH UR V1 Y

EXOGENOUS:
AF ALPHA D644 D651 D704 D711 EX G GG GNPRP HSQ IMP MA MCG MOOD
PE2 P1 P2 T YA YG

FUNCTION:
LOG

COEFFICIENT:
C121 C122 C161 C162 C163 C164 C165 C166 C167 C168 C169 C171 C172
C173 C191 C192 C201 C202 C203 C31 C32 C33 C34 C35 C36 C37 C38
C41 C42 C43 C51 C52 C53 C61 C62 C63 C64 C65 C71 C72 C73 C74
C75 C81 C82 C83 C84 C85 C86 C87 C88 R16 R17 R19 R20 R3 R4 R5
R6 R7 R8

EQUATIONS

```
 1:   V1 == V-V(-1)
 2:   E == M+MA+MCG-D+300
 3:   UR == 1-E/(LF1+LF2-AF)
 4:   GNP == CD+CN+CS+IP+IH+DEL(1 : V)-IMP+EX+G
 5:   GNPR == 100*(GNP-GG)/PD)+YG
 6:   GAP2 == GNPRP-GNPR(-1)-DEL(1 : GNP)
 7:   Y == GNPR-YA-YG
 8:   MH == 1/ALPHA*Y*1000000
 9:   CD = R3*CD(-1)+C31*(1-R3)+C32*(GNP-R3*GNP(-1))+C33*(MOOD(-1)-R3*MOOD(-2))
           +C34*(MOOD(-2)-R3*MOOD(-3))+C35*(D644-R3*D644(-1))+C36*(D651-R3*D651(-1))
           +C37*(D704-R3*D704(-1))+C38*(D711-R3*D711(-1))
10:   CN = R4*CN(-1)+C41*(GNP-R4*GNP(-1))+C42*(CN(-1)-R4*CN(-2))+C43*(MOOD(-2)
           -R4*MOOD(-3))
11:   CS = R5*CS(-1)+C51*(GNP-R5*GNP(-1))+C52*(CS(-1)-R5*CS(-2))+C53*(MOOD(-2)
           -R5*MOOD(-3))
12:   IP = R6*IP(-1)+C61*(1-R6)+C62*(GNP-R6*GNP(-1))+C63*(PE2-R6*PE2(-1))
           +C64*(D704-R6*D704(-1))+C65*(D711-R6*D711(-1))
13:   IH = R7*IH(-1)+C71*(1-R7)+C72*(GNP-R7*GNP(-1))+C73*(HSQ-R7*HSQ(-1))
           +C74*(HSQ(-1)-R7*HSQ(-2))+C75*(HSQ(-2)-R7*HSQ(-3))
14:   DEL(1 : V) = R8*DEL(1 : V(-1))+C81*(1-R8)+C82*(CD(-1)+CN(-1)-R8*(CD(-2)
           +CN(-2)))+C83*(V(-1)-R8*V(-2))+(-C84)*(DEL(1 : CD+CN)-R8*DEL(1 : CD(-1)
           +CN(-1)))+C85*(D644-R8*D644(-1))+C86*(D651-R8*D651(-1))
           +C87*(D704-R8*D704(-1))+C88*(D711-R8*D711(-1))
15:   DEL(1 : PD) = C121+C122*SUM(I = -19 TO 0 : GAP2(I))/20
16:   DEL(1 : LOG(M)) = R16*DEL(1 : LOG(M(-1)))+C161*(1-R16)+C162*(T-R16*T(-1))
           +C163*(LOG(M(-1))-LOG(MH(-1))-R16*(LOG(M(-2))-LOG(MH(-2))))
           +C164*(DEL(1 : LOG(Y(-1)))-R16*DEL(1 : LOG(Y(-2))))+C165*(DEL(1 : LOG(Y))
           -R16*DEL(1 : LOG(Y(-1))))+C166*(D644-R16*D644(-1))+C167*(D651-R16*D651(-1))
           +C168*(D704-R16*D704(-1))+C169*(D711-R16*D711(-1))
```