COWLES FOUNDATION FOR RESEARCH IN ECONOMICS

AT YALE UNIVERSITY

Box 2125, Yale Station
New Haven, Connecticut 06520

COWLES FOUNDATION DISCUSSION PAPER NO. 537

Note: Cowles Foundation Discussion Papers are preliminary materials circulated to stimulate discussion and critical comment. Requests for single copies of a Paper will be filled by the Cowles Foundation within the limits of the supply. References in publications to Discussion Papers (other than mere acknowledgment by a writer that he has access to such unpublished material) should be cleared with the author to protect the tentative character of these papers.

AN ALGORITHM FOR FIML AND 3SLS ESTIMATION OF LARGE NONLINEAR MODELS

William R. Parke

October 31, 1979

Pg. 21
missing

AN ALGORITHM FOR FIML AND 3SLS ESTIMATION OF LARGE NONLINEAR MODELS*

by

William R. Parke

ABSTRACT

This paper presents a numerical algorithm for computing full information maximum likelihood (FIML) and nonlinear three-stage least squares (3SLS) coefficient estimates for large nonlinear macroeconometric models. The new algorithm, which is demonstrated by actually computing FIML and 3SLS coefficient estimates for two versions of the 97 equation Fair Model, is substantially more effective than other algorithms on FIML and 3SLS estimation problems.

*This paper is a revised version of a dissertation presented for the degree of Doctor of Philosophy at Yale University.

I would like to thank Ray Fair, Kim Peck, and Randy Olsen for helpful comments on various versions of this paper. In addition, Ray Fair supplied several computer programs and the data, and Yale University provided financial assistance over the course of this project. The research described in this paper was undertaken with grants from the National Science Foundation. Computer programs implementing this algorithm are available from the author.

1. Introduction

This paper presents a numerical algorithm for computing full information maximum likelihood (FIML) and nonlinear three-stage least squares (3SLS) coefficient estimates for large nonlinear models. (The proposed algorithm will be denoted algorithm A.) Although the theory of FIML estimation has been available for thirty years [20], FIML estimation has long been regarded as impossible on large nonlinear models.[1] The more recently proposed nonlinear 3SLS estimator [19] also poses difficulties for large models.[2] Using the algorithm presented in this paper, FIML and 3SLS are now feasible alternatives for estimation of large nonlinear models.

The principles behind algorithm A's efficiency can be summarized as follows. First, most of the algorithm's steps explicitly control the mean of each equation's residuals. Large changes in those means are avoided because the FIML and 3SLS estimation problems are extremely sensitive to the residuals' means. Second, the coefficients are grouped by equations and the groups are treated separately during most of each

[1] Using previously proposed algorithms, nonlinear FIML estimation has been possible only on small models. Newton's method [6, 7, 8] has successfully been applied to linear models with up to 14 equations [3, 5, 9], but that method is not feasible for nonlinear models. Two general purpose maximization algorithms [16, 23] and two algorithms suitable only for FIML and 3SLS estimation [4, 9] have been proposed for nonlinear estimation problems. The largest model that has been estimated using these algorithms has 19 equations and 61 unknown coefficients [2, 9, 12], but that model's nonlinear portion is recursive, making the Jacobian independent of time. In comparison, algorithm A has computed FIML and 3SLS estimates for two models, one with 48 equations and 88 unknown coefficients and one with 97 equations and 107 unknown coefficients.

[2] The largest nonlinear model previously estimated using 3SLS is the 19 equation model mentioned in the previous footnote [2].

iteration. This focus on individual equations is effective because coefficients within the same equation are generally more strongly related than are coefficients within different equations. Third, a hierarchical arrangement of the algorithm's steps allows certain steps to explore directions that have been traced out by a series of earlier steps. Fourth, no derivatives of the objective function are required, eliminating one obstacle to estimating nonlinear models.

The outline of this paper is as follows. Section II defines the FIML and 3SLS estimators, and Section III describes algorithm A. Section IV reviews the results of using algorithm A to estimate two versions of the 97 equation Fair Model. Two appendices discuss the algorithm's theoretical rate of convergence and a procedure for computing the FIML covariance matrix.

II. The FIML and 3SLS Estimators

Write the general structural model as:

$$f_i(y_t, x_t, \beta) = u_{it} , \quad (t = 1, \ldots, T) , \quad (i = 1, \ldots, N) , \tag{1}$$

where y_t is a vector of endogenous variables, x_t is a vector of predetermined variables, β is a vector of unknown coefficients, and u_{it} is a scalar residual. The first G equations are stochastic, and the remaining $N-G$ equations are identities.

The FIML coefficient estimates [20] maximize:

$$F_4(\beta) = -\frac{T}{2}\log(|S|) + \sum_{t=1}^{T}\log(|J_t|) . \tag{2}$$

S is the $G \times G$ covariance matrix of the stochastic equations' residuals,

$(S)_{ij} = \frac{1}{T}\sum_{t=1}^{T} u_{it}u_{jt}$. J_t is the $N \times N$ Jacobian matrix of partial derivatives of the residuals with respect to the endogenous variables, $(J_t)_{ij} = \partial u_{it}/\partial y_{jt}$.

Recently, Jorgenson and Laffont [19] have proposed a nonlinear generalization of the linear three-stage least squares estimator. Their 3SLS coefficients maximize:[3]

$$(3) \qquad F_3(\beta) = -u'[\Sigma^{-1} \otimes Q(Q'Q)^{-1}Q']u \ .$$

Q is a $T \times K$ matrix of first stage regressors, and Σ is a fixed, consistent estimate of the error terms' covariance matrix, derived perhaps from the 2SLS coefficient estimates. The vector u is a $TG \times 1$ vector of the residuals arranged as: $u = (u_{11}, \ldots, u_{1T}, \ldots, u_{G1}, \ldots, u_{GT})'$.

III. Algorithm A

A. General Description

Algorithm A, which is a relaxation algorithm, generates a hierarchical sequence of search directions. Each iteration includes 1 maximizing search for each coefficient, 2 maximizing searches for each equation, and up to 10 maximizing searches for the complete coefficient vector. The searches for each coefficient also adjust the constant terms to avoid disturbing the mean of any equation's residuals. The searches for each

[3] Several additional types of coefficient estimates can be found using the basic 3SLS setup. If Σ is taken to be any diagonal matrix, the resulting objective function, $F_2(\beta)$, is maximized by the nonlinear two-stage least squares coefficients [1]. If, in addition, an identity matrix is substituted for $Q(Q'Q)^{-1}Q'$, the resulting objective function, $F_1(\beta)$, is maximized by the nonlinear least squares coefficients.

equation combine the results of the individual coefficient searches, and the searches for the complete coefficient vector are generated from the results of past iterations.

Sections B through H describe the algorithm in detail. Section B establishes notation and explains the information about the model that must be supplied by the user. The order of the hierarchical search directions is discussed in Section C. Sections D, E, and F specify the three types of search directions. Section G gives the modifications necessary when there are constraints across coefficients. Section H gives the procedure for maximizing in any one of the search directions. (Several practical matters, such as the criteria for terminating a sequence of iterations, are discussed in Section IV, and Appendix One outlines the algorithm's theoretical rate of convergence.)

B. Notation and Input Requirements

Algorithm A maximizes an objective function, $F(\beta)$. The FIML objective function, $F_4(\beta)$, is the logarithm of the likelihood function (2). The 3SLS objective function, $F_3(\beta)$, is the negative of the least squares distance function (3). The present discussion assumes that a routine is available to evaluate one of these functions for any given coefficient vector.

Algorithm A also needs certain information about the model's structure. This information is supplied (by the user) in the form of several vectors that are of the same dimension as the coefficient vector (see Table 1). None of this information is used in the objective function evaluation, a task external to this algorithm.

The user supplied vectors describe each coefficient's place in

TABLE 1

Notation for Vectors of the Same Dimension as β

Vector Element	Definition
$\beta(i)$	The value of the i^{th} coefficient.
$t(i)$	The i^{th} coefficient's type.[a]
$n(i)$	The number of the equation containing $\beta(i)$.
$c(i)$	The index of the constant term of the equation containing $\beta(i)$. That is, $\beta(c(i))$ is the constant term of the equation containing $\beta(i)$.
$z_t(i)$	The value of the variable multiplied by $\beta(i)$ in forming $u_{n(i),t}$.[b] That is, the term $\beta(i)\cdot z_t(i)$ appears in equation $n(i)$.
$m(i)$	The mean of $z_t(i)$ over the sample period.
$v(i)$	The value of the entry in the search direction vector corresponding to $\beta(i)$. (Not supplied by the user.)

[a] $\beta(i)$ can be classified as either a constant term, a serial correlation coefficient, a coefficient of an explanatory variable, or none of the first three types. (See the discussion of Step 1 in Section D.)

[b] $z_t(i)$ is often some transformation, such as the logarithm, of one or more variables. Some coefficients (constant terms and serial correlation coefficients, for example) have no $z_t(i)$.

the model.[4] For coefficient $\beta(i)$, three integer-valued vectors, t, n, and c, give its coefficient type, the number of the equation it is in, and the index of that equation's constant term (if there is one).[5] The real-valued vector element $m(i)$ gives the mean of $z_t(i)$ over the sample period, where $\beta(i)$ is the coefficient for $z_t(i)$. Given these input vectors, algorithm A generates search directions (represented in Table 1 by the vector v) that have an entry for each coefficient.

C. Order of Coefficients and Steps

The coefficients are numbered as follows. All of the coefficients appearing in an equation are numbered consecutively, with the constant term last and the serial correlation coefficient, if there is one, next to last. If a lagged dependent variable appears in an equation, its coefficient should be the first one for that equation.

Given the coefficient numbering, the search directions are generated as shown in Table 2. Step 1 associates a search direction with each coefficient in numerical order, with one exception. After the last coefficient for a given equation (usually this will be the constant term), Step 2 creates a search direction for the whole equation. The algorithm then proceeds to Step 1 on the first coefficient of the next equation. After the coefficient list has been exhausted and Step 2 has been executed for the last equation, Step 3 creates search directions for the whole model. Steps 4 and 5 then repeat the search directions from Steps 2 and 3.

[4] Vector elements corresponding to different coefficients are identified by an index within parentheses (e.g., $\beta(i)$ is the i^{th} coefficient). Different vectors will be identified by subscripts (e.g., β_n is the coefficient vector after n iterations).

[5] Some of these vector elements may not be defined for every coefficient. In that case, the undefined vector element will not be used in the algorithm.

TABLE 2

Selection of Search Directions*

I. For Each Equation in Turn:

Step 1a. Perturb the lagged dependent variable's coefficient, if any, and adjust the constant term.

Step 1b. Perturb the other variables' coefficients, and adjust the constant term.

Step 1c. Perturb the serial correlation coefficient, if any, without adjusting the constant term.

Step 1d. Perturb the constant term by itself.

Step 2. Subtract the coefficient vector at the start of Step 1 for this equation from the coefficient vector before Step 1d.

II. For the Whole Model:

Step 3a. Subtract the coefficient vector at the start of this iteration (before Step 1 for the first equation) from the current coefficient vector.

Step 3b. Subtract the starting coefficient vector, and three other previous coefficient vectors, from the current coefficient vector.

III. For Each Equation in Turn:

Step 4. Repeat the directions selected for Step 2.

IV. For the Whole Model:

Step 5. Repeat the directions selected for Step 3.

*Sections D through G give detailed rules for constructing these search directions. Section H gives the procedure for maximizing in any particular search direction.

D. Step 1

In Step 1, a search direction for each individual coefficient is constructed to satisfy the "means condition." That condition is met if, for any β along the search direction, the mean of each equation's residuals remains unchanged from its current value.[6] Since the right-hand side variables generally have nonzero means, the mean of an equation's residuals will almost certainly be changed if just one coefficient in that equation is changed. Therefore, in order to satisfy the means condition, the constant term (if the equation has a constant term) is changed along with each of the other coefficients. The search for the constant term itself changes only that one coefficient and does not satisfy the means condition.[7]

For an equation that is linear in its coefficients, a straight-line search satisfies the means condition. (Nonlinearities in the right-hand side variables do not matter because $m(i)$ and $z_t(i)$ are defined in terms of transformed variables.) The search direction vector for each coefficient, except the constant term, has two nonzero elements: $v(i) = \beta(i)$ and $v(c(i)) = -m(i) \cdot \beta(i)$.[8]

[6] The rationale for explicit attention to the residuals' means comes from simple linear regression. For ordinary least squares on equations including a constant term, one of the normal equations equates the mean of the dependent variable with the sum of the independent variables' means multiplied by their respective coefficients. In other words, the residuals will have a mean of zero by construction.

There is no such explicit statement about the residuals' means for full information estimation. Indeed, one would expect the mean of any one equation's residuals to be slightly nonzero in light of the covariance interactions between different equations. However, one would not expect the mean of the residuals to be far from zero.

[7] The searches for the constant terms alone allow the residuals' means to adjust to any value. In practice, the searches for the constant terms, which come last for each equation, make small adjustments in those means.

[8] For example, consider the following equation:

For an equation with nonlinearities in its coefficients, a more complicated procedure is necessary to satisfy the means condition. Suppose that Step 1 for $\beta(i)$ changes that coefficient to $\beta'(i)$. Then $\beta(c(i))$ must be changed to a value $\beta'(c(i))$ that satisfies the means condition. The relation between the changes in the two coefficients is, however, nonlinear, and a straight-line search will not satisfy the means condition. The desired relation between the changes in $\beta(i)$ and $\beta(c(i))$ must be determined from the specific functional form of the equation.

An equation that includes a serial correlation coefficient, but is otherwise linear in its coefficients, is amenable to straight-line search paths. The serial correlation coefficient is changed alone: $v(i) = \beta(i)$ and $v(c(i)) = 0$. The other coefficients are treated as if the equation were linear: $v(i) = \beta(i)$ and $v(c(i)) = -m(i)\cdot\beta(i)$. While these search directions only approximate the curved search paths that satisfy the means condition, the approximation is generally satisfactory.[9]

$$u_t = y_t - a - b\cdot x_t \ , \quad t = 1, \ldots, T \ .$$

The mean of the residuals, $\bar{u}$, is given by:

$$\bar{u} = \bar{y} - a - b\cdot\bar{x} \ , \quad \text{where } \bar{y} \text{ and } \bar{x} \text{ are the variables' means.}$$

If the change in coefficient a is $-\bar{x}$ times the change in coefficient b , there will be no change in $\bar{u}$.

[9] For example, consider the following equation:

$$u_t = y_t - a - b\cdot x_t - \rho\cdot(y_{t-1} - a - b\cdot x_{t-1}) \ , \quad t = 1, \ldots, T \ .$$

The mean of the residuals is given by:

E. Steps 2 and 4

Immediately after Step 1 has been completed for an equation's coefficients, Step 2 searches along a direction of increase in $F(\beta)$ for the whole equation.[10] The search direction vector is the difference between the coefficient vector before Step 1 for that equation and the coefficient vector just before the constant term search for that equation. (This difference vector has zero entries for the other equations' coefficients.) The directions used in Step 2 are saved and reused in Step 4, with no intervening occurrences of Step 1.

F. Steps 3 and 5

Following completion of Steps 1 and 2 for all of the equations, Step 3 is executed. Each search direction in Step 3 is the difference between the current coefficient vector and a past coefficient vector. (All of the elements of these difference vectors will in general be nonzero.) The past coefficient vectors consist of β_0 (the initial

$$\bar{u} = (1-\rho)\cdot(\bar{y} - a - b\cdot\bar{x}) + \rho\left\{\frac{y_T - y_0}{T} - b\frac{x_T - x_0}{T}\right\} ,$$

where y_0 , x_0 , y_T , and x_T are the variables' first and last observations. The mean $\bar{u}$ will not change substantially if the change in coefficient a is $-\bar{x}$ times the change in coefficient b because $\rho((x_T - x_0)/T)$ is usually much smaller than $(1-\rho)\cdot\bar{x}$. Changes in ρ alone will have little effect on $\bar{u}$ because $\bar{y} - a - b\cdot\bar{x}$ and $(y_T - y_0)/T -$ $- b((x_T - x_0)/T)$ are both likely to be small.

[10] Step 2 focuses on individual equations for two reasons. First, the strongest multicollinearity will generally be among coefficients in the same equation, rather than across equations. The multicollinearity will be associated with a ridge in the objective function, which may be revealed by the coefficient changes in Step 1. Second, Step 2 enjoys the advantage of using a different steplength for each equation's search direction. One steplength might not be optimal for all equations together.

coefficient vector,[11] β_{n-1} (the coefficient vector at the start of the current iteration), and, as they become available, β_{n-2}, β_{n-3}, and β_{n-4}.[12]

C. Constraints on the Coefficients

If one constraint is imposed across coefficients, then one previously unconstrained coefficient, say $\beta(j)$, is no longer explicitly in the coefficient vector over which $F(\beta)$ is maximized. The value for $\beta(j)$ is calculated implicitly by solving the constraint equation given the other coefficients' values. Since the implicit value of $\beta(j)$ changes whenever the value of any other coefficient involved in the constraint changes, the constant term adjustments in Step 1 are more complicated.

Consider first the most general case, a nonlinear constraint imposed on nonlinear equations. Suppose that Step 1 changes $\beta(i)$, which is involved in the constraint, to $\beta'(i)$, implying a change in $\beta(j)$ to $\beta'(j)$. Satisfying the means condition requires changing $\beta(c(i))$ to some value $\beta'(c(i))$ and, if $\beta(i)$ and $\beta(j)$ are in different equations, changing $\beta(c(j))$ to some value $\beta'(c(j))$. However, the search path that satisfies the means condition is in general curved and depends upon

[11] Other algorithms proposed for FIML estimation attempt to navigate the immensely complicated local neighborhood of β_n without ever considering this globally obvious direction. In practice, this one search can greatly enhance the effectiveness of an iteration.

[12] More past coefficient vectors, as well as coefficient vectors from other estimators, could be added to this list, particularly if Step 3 and 5 searches are not expensive relative to Step 1, 2, and 4 searches. For example, the 3SLS objective function involves a very large number of arithmetic operations when it is evaluated for a completely new set of coefficients, as it is in Steps 3 and 5. Consequently, the cost of a Step 3 or 5 search relative to the cost of a Step 1 or 2 search is greater for 3SLS than for FIML (see the top two lines in Table 4).

the specific functional forms of the constraint and the equations it affects.

If the equations involved in the constraint are linear in their coefficients, the constant term adjustments can be given in terms of the right-hand side variables' means. The change in $\beta(c(i))$ is $-m(i)\cdot(\beta'(i)-\beta(i))$ and the change in $\beta(c(j))$ is $-m(j)\cdot(\beta'(j)-\beta(j))$. If $\beta(i)$ and $\beta(j)$ are in the same equation, the change in $\beta(c(i))$ is $-m(i)\cdot(\beta'(i)-\beta(i)) - m(j)\cdot(\beta'(j)-\beta(j))$.

If the constraint is linear and the equations involved in the constraint are linear in their coefficients, then the search path is a straight line. The above simplification for linear equations can be used. In addition, $\partial\beta(j)/\partial\beta(i)$ is constant for a linear constraint so that $\beta'(j) - \beta(j)$ can be replaced by $[\partial\beta(j)/\partial\beta(i)]\cdot(\beta'(i)-\beta(i))$. If $\beta(i)$ and $\beta(j)$ are in different equations, the search direction vector has three nonzero elements: $v(i) = \beta(i)$, $v(c(i)) = -m(i)\cdot\beta(i)$, and $v(c(j)) = -m(j)\cdot[\partial\beta(j)/\partial\beta(i)]\cdot\beta(i)$. If $\beta(i)$ and $\beta(j)$ are in the same equation, there are two nonzero elements: $v(i) = \beta(i)$ and $v(c(i)) = (-m(i) - m(j)\cdot[\partial\beta(j)/\partial\beta(i)])\cdot\beta(i)$.

No changes in Steps 3 and 5 are necessary for any type of constraint. If a constraint is across coefficients within just one equation, no changes in Steps 2 and 4 are necessary. If a constraint is across two or more equations, Steps 2 and 4 should create one search direction for all of the coefficients changed during Step 1 for a given equation.

H. Maximization in a Given Search Direction

Given a search direction v , the objective function is evaluated at three points: β , $\beta+\delta v$, and $\beta-\delta v$, where δ is a steplength discussed below. The algorithm calculates the steplength s such that the coefficient vector $\beta + sv$ maximizes the quadratic polynomial interpolated through these three points. (If the second derivative of that polynomial is positive, s is taken to be 0 .) Of the four coefficient vectors: β , $\beta + \delta v$, $\beta - \delta v$, and $\beta + sv$, the one with the largest objective function value becomes the new coefficient vector. The algorithm then initiates the procedure to pick the next search direction.

The steplength δ is scaled to the size of the basic determinant of the search direction v . For a direction associated with a specific coefficient (Step 1), that coefficient is perturbed by 1% of its value (.01% for a constant term) and the other nonzero search vector entries are scaled proportionately. For the search directions that are differences between coefficient vectors (Steps 2 through 5), the steplength δ is 10%. Neither parameter choice is critical.

The total number of objective function evaluations per iteration can now be calculated in terms of the size of the model being estimated. Since each search in a given direction entails three evaluations, counting the search directions listed above yields the following formula:

$$(4) \qquad \frac{\text{F evaluations}}{\text{iteration}} = 3\cdot(\#\text{ coefficients} + 2\cdot\#\text{ equations} + 10) \ .$$

Equation (4) also determines the cost of each iteration of algorithm A because that cost is due almost entirely to the objective function evaluations.

IV. Computational Experience

Algorithm A has been used to compute FIML and 3SLS coefficient estimates for two versions of the Fair Model, denoted FM76 and FM78.[13] FM76 has 26 stochastic equations and 22 identities, or 48 total equations. FM78 has 29 stochastic equations and 68 identities, or 97 total equations. Of the 166 coefficients in FM76's stochastic equations, 88 are estimated using FIML and 3SLS.[14] For FM78, 107 of the 182 coefficients in the stochastic equations are estimated using FIML and 3SLS.[15] Both FM76 and FM78 include nonlinearities in their coefficients (due to serial correlation coefficients) and nonlinearities in their variables. In addition. FM76 includes a nonlinear constraint across the coefficients of two equations [14, pp. 11-13]. Although other nonlinear models have been estimated using algorithms mentioned in footnote 1, FM76 and FM78 are substantially larger than those other models.

Before applying algorithm A to these FIML and 3SLS estimation problems, the costs of calculating $F_4(\beta)$ and $F_3(\beta)$ were reduced in four ways. First, for search directions that change only coefficients in a single equation, only that equation's residuals are recomputed, and only one row and column in the residuals' covariance matrix is recomputed.

[13] FM76 is specified in [13]. Note especially pages 80-83. FM78 is a revised model documented in [14].

[14] Equations 1, 2, 3, 4, 8, 9, 10, 12, 13, 15, 16, 21, and 22 (FM76) and equations 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 15, 16, 24, and 90 (FM78) are estimated using FIML and 3SLS. The FIML objective functions do include all of the stochastic equations and identities, and the 3SLS objective functions do include all of the stochastic equations. Those coefficients that are not estimated by FIML and 3SLS are fixed at their 2SLS values.

[15] The FIML and 3SLS coefficient estimates for FM78 are given in [15].

(Note that Steps 1, 2, and 4 generally change only coefficients within a single equation.) Second, the calculations for $F_3(\beta)$ are broken down into blocks by pairs of equations, and the intermediate results are saved. Only a small percentage of the intermediate results need to be recalculated for coefficient changes within one equation. Third, the determinants of the Jacobian matrices are calculated using a determinant algorithm specially designed for matrices with relatively few nonzero elements [17].[16] Fourth, the determinants of the first and last Jacobians are calculated, and a linear interpolation is used to approximate the remaining Jacobians' determinants, which are not calculated.[17] The resulting costs of each iteration are given in the top half of Table 4.

Table 4 also shows the changes in $F(\beta)$ during the four sequences of iterations.[18] The FIML objective function's value increased by 64.27 during 12 iterations on FM76[19] and by 42.21 during 28 iterations on

[16] The Jacobians' determinants need not be recomputed for coefficient changes that do not alter their elements. (Consider, for example, Step 1 for the coefficient of a predetermined variable.) The present calculations do not take advantage of this possibility, which would have eliminated over half of the Jacobian determinant evaluations.

[17] For FM78, Fair and Parke [15] try using six Jacobians and show that using only two Jacobians has little effect on the FIML coefficient estimates. Since only one Jacobian is needed for a linear model, this empirical result is perhaps indicative of the degree of nonlinearity of the Fair Model. As noted in [15], one could use all of the Jacobians for the last few iterations to verify the final estimates.

In fact, given the rate of convergence of algorithm A on FIML estimation of FM76 and FM78, it would be feasible, but more expensive, to estimate those models using all 82 and 98 Jacobians respectively. FIML estimation of either model would not be feasible, even using only two Jacobians, with any other known algorithm.

[18] Since the econometric merits of 2SLS, 3SLS, and FIML are not an issue here, no attempt is made to contrast the economic implications of the various coefficient vectors. See [15] for additional discussion of the 2SLS, 3SLS, and FIML coefficient estimates of FM78.

TABLE 4

FIML and 3SLS Estimation of FM76 and FM78

Estimator	FIML	FIML	3SLS	3SLS
Model	FM76	FM78	FM76	FM78
CPU Time per $F(\delta)$ Evaluation (Seconds)[a]				
Coefficient Changes within One Equation	n.a.[b]	.20	.12	.40
Changes in the Entire Coefficient Vector	.41	.64	2.18	2.94
$F(\delta)$ Evaluations per Iteration (minimum)[c]	354	432	354	432
CPU Time per Iteration (minutes)	2.3	1.9	1.4	3.8
Cost per Iteration (without discounts)	$28	$24	$17	$47
Total Change in $F(\delta)$	64.27	42.21	116.59[d]	48.33
Number of Iterations Executed	12	28	6	28
Change in F on Iteration 1	42.80	19.66	107.37	21.90
2	7.54	8.16	4.07	9.52
3	3.54	3.31	2.98	3.79
4	3.40	1.91	.92	1.89
5	1.96	1.11	.77	1.85
6	1.54	1.56	.48	.86
7	.92	.91		.63
8	.71	.90		1.11
9	.45	.65		.72
10	.48	.50		.63
11	.52	.79		.72
12	.41	.61		.55
13		.36		.56
14		.25		.37
15		.17		.39
16		.07		.27
17		.16		.36
18		.19		.23
19		.21		.25
20		.13		.23
21		.13		.20
22		.14		.19
23		.10		.21
24		.06		.20
25		.04		.22
26		.05		.25
27		.03		.09
28		.05		.14

[a] These figures refer to Yale University's IBM 370/158. All computations were performed using double precision Fortran.

[b] The shortcuts involving coefficient changes within one equation were not used in the FIML estimation of FM76.

[c] This figure increases slightly as search directions are added to Steps 3 and 5.

[d] All series of iterations start from the 2SLS estimates, except the 3SLS estimation of FM76, which starts from the FIML estimates.

FM78.[20] The 3SLS objective function's value increased by 116.59 during 6 iterations on FM76 and by 48.33 during 28 iterations on FM78.

All four sets of figures exhibit an excellent rate of convergence, given the sizes of the problems. The first iteration of each sequence achieved about half or more of the total increase from the starting coefficients to the final estimates. Over three-fourths of the total increase was secured, in all four cases, by the first four iterations.

The pattern of changes in the objective function affects the decision to terminate a series of iterations.[21] The algorithm may eventually, as indicated in Appendix One, be converging so that $F(\beta_n) - F(\beta_{n-1})$ is on average proportional to $F(\beta^*) - F(\beta_{n-1})$. The pattern of increases over the sequence of iterations would then give an indication of the reduction in $F(\beta^*) - F(\beta_n)$. The figures in Table 4 for FIML and 3SLS estimation of FM78 suggest that $F(\beta^*) - F(\beta_{28})$ is a very small fraction of $F(\beta^*) - F(\beta_0)$.

[19] Using Powell's algorithm for 24 iterations, Fair [13, p. 83] obtained an increase of 17.43 in the FIML objective function for FM76. Since that increase is only slightly over one-quarter of the increase with algorithm A, it is clear that Powell's algorithm was in fact converging at a hopelessly slow rate. Note that algorithm A increased $F(\beta)$ by over twice as much on the very first iteration as the whole sequence of iterations using Powell's algorithm.

[20] The Davidon-Fletcher-Powell algorithm [18] with constant term adjustments similar to Step 1 of algorithm A increased the FIML likelihood function for a revised version of FM78 by .24 in 4 iterations. (That is, DFP was applied to the α coefficient vector described in Appendix One.) The increase in $F(\beta)$ on the final iteration was so small that the double precision rounding errors dominated the Hessian updating formula, and the algorithm was forced to terminate.

[21] Algorithm A can easily be restarted after any number of iterations. In practice, a number of iterations are executed, the results are examined, and then a decision is made on executing additional iterations. Although automatic criteria for stopping could be incorporated, a discretionary process is convenient given the expense of a series of iterations and the limited experience in estimating large nonlinear models by FIML and 3SLS.

Successive changes in the coefficients are a second criterion for diagnosing adequate convergence. Since β^* is only an estimate of some true coefficient vector β^0, the degree of convergence may be adequate if $\beta_n - \beta^*$ is small relative to $\beta^* - \beta^0$. The difference $\beta^* - \beta^0$ can be measured by the coefficients' estimated standard errors, and $\beta_n - \beta^*$ can be roughly approximated by $\beta_n - \beta_{n-1}$. After 28 iterations on the FIML and 3SLS objective functions for FM78, the coefficients' estimated standard errors were much larger than the changes between iterations.[22] Additional iterations may thus have yielded somewhat more accurate computations of β^*, but would appear to have held little promise of significantly improving the estimate of β^0.

[22] On the 28th iteration for FIML, 48 coefficients changed by less than .01 standard error, 56 coefficients changed by between .01 and .1 standard error, and 2 coefficients changed by more than .1 standard error. For the 28th iteration for 3SLS, these figures are 66, 33, and 1 respectively.

APPENDIX ONE

A Theoretical Bound on the Rate of Convergence

Step 1 of algorithm A is equivalent to univariate relaxation on a transformed coefficient vector α . The vector α is defined by:

$$(5) \qquad \beta = M\cdot\alpha ,$$

where the square matrix M is composed of an identity matrix plus some additional nonzero elements representing the constant term adjustments. Unless there are constraints across equations, M is block diagonal. The only nonzero element in the column of M corresponding to a constant term or a serial correlation coefficient is the diagonal element.

For example, consider a linear equation including two right-hand side variables with means of zero:

$$(6) \qquad y_t(4) = \alpha(1)\cdot y_t(1) + \alpha(2)\cdot y_t(2) + \alpha(3) + u_t ,$$

The complementary equation in terms of the β coefficients has right-hand side variables with nonzero means:

$$(7) \qquad y_t(4) = \beta(1)\cdot(y_t(1)+m(1)) + \beta(2)\cdot(y_t(2)+m(2)) + \beta(3) + u_t .$$

The associated diagonal block of M is:

$$(8) \qquad \begin{bmatrix} \beta(1) \\ \beta(2) \\ \beta(3) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -m(1) & -m(2) & 1 \end{bmatrix} \cdot \begin{bmatrix} \alpha(1) \\ \alpha(2) \\ \alpha(3) \end{bmatrix} .$$

Note that each change in $\alpha(1)$ or $\alpha(2)$ alone produces a change in the corresponding β coefficient plus a constant term adjustment.

Define an algorithm's rate of convergence to be the smallest constant R such that, for some constant K and for n greater than some n_0 :

$$(9) \qquad -(\beta_n - \beta^*)' \cdot H^{-1}(\beta^*) \cdot (\beta_n - \beta^*) \leq R^n \cdot K ,$$

where $H(\beta^*)$ is the Hessian matrix at the maximizing coefficient estimates β^* . Then, $k = -\log(2)/\log(R)$ iterations are needed to ensure that $F(\beta^*) - F(\beta_n)$, which is approximated by the left-hand side of (9), will be no more than one-half of $F(\beta^*) - F(\beta_{n-k})$. Convergence at rate R also implies that $F(\beta_n) - F(\beta_{n-1})$ is on average a fraction 1-R of $F(\beta^*) - F(\beta_{n-1})$.

Although algorithm A's complexity precludes an analytic expression for its rate of convergence, that rate should be somewhat faster than the rate of convergence for an algorithm composed of Step 1 alone. The latter rate of convergence can be given as follows. Let $r(H)$ be the spectral radius of $L^{-1} \cdot U$, where L and U are the lower triangular and strictly upper triangular decomposition of H . Univariate relaxation on the α coefficient vector, which is equivalent to Step 1 on the β coefficient vector, converges at rate $r(H(\alpha^*))$, where $H(\alpha^*)$ is the Hessian of the α coefficient vector at $\alpha^* = M \cdot \beta^*$ [21]. That Hessian is also equal to $M' \cdot H(\alpha^*) \cdot M$. The rate of convergence of Step 1 on the β coefficient vector is thus equal to $r(H(\alpha^*))$ or, equivalently, $r(M' \cdot H(\beta^*) \cdot M)$.[23]

[23] This discussion assumes that algorithm A does in fact converge. Convergence will occur under the standard assumptions for relaxation algorithms [24], the most important of which is convexity of $F(\beta)$ in an appropriate neighborhood of β^* . The FIML and 3SLS objective functions, which may

APPENDIX TWO

The Estimated FIML Covariance Matrix

The estimated covariance matrix for the FIML coefficient estimates is not involved in algorithm A, but its computation has proven to be an interesting and difficult problem. After the FIML coefficient estimates were computed, a variety of strategies for numerically differentiating the likelihood function all failed to yield a positive definite covariance matrix. This problem, if not solved, would severely restrict opportunities to analyze the FIML coefficient estimates. Fortunately, a procedure related to Step 1 of algorithm A does yield a positive definite covariance matrix estimate.

The standard formula for the asymptotic FIML covariance matrix is:

$$(11) \qquad V_4(\beta) = -\left[\frac{\partial^2 F_4(\beta)}{\partial\beta \; \partial\beta'}\right]^{-1} .$$

The second derivatives can be numerically approximated by:

$$(12) \qquad \left.\frac{\partial^2 F(\beta)}{\partial\beta(i) \; \partial\beta(j)}\right|_{\beta} = [F(\beta + \delta_i\beta(i)e_i + \delta_j\beta(j)e_j) - F(\beta + \delta_i\beta(i)e_i) - F(\beta + \delta_j\beta(j)e_j) + F(\beta)]/(\delta_i\beta(i)\delta_j\beta(j)) ,$$

where each δ_i is a scalar perturbation parameter and e_i is the i^{th} coordinate vector. Unfortunately, $V_4(\beta)$ failed to be positive definite for any choices of the perturbation parameters and for several other numerical differentiation formulas.[25] The elements of $V_4(\beta)$ also were very sensitive to the differentiation strategy.

Appendix One suggests that the estimated coefficients' covariance matrix could be nearly singular as a result of the nonzero means of the right-hand side variables. The transformation (5) defining the α coefficients ($\beta = M\cdot\alpha$) neutralizes the effects of those nonzero means. The α coefficients' covariance matrix estimate is:

$$(13) \qquad V_4(\alpha) = -\left[\frac{\partial^2 F_4(M\cdot\alpha)}{\partial\alpha\ \partial\alpha'}\right]^{-1} .$$

The β coefficients' covariance matrix estimate is then:

$$(14) \qquad V_4(\beta) = M\cdot V_4(\alpha)\cdot M' .$$

Formulas (13) and (14) are less susceptible to the problem noted above than is (11). On FM78, the elements of $V_4(\beta)$ computed using (13) and (14) were not sensitive to the values of the δ_i's , and the matrix $V_4(\beta)$ was positive definite for the whole range of δ_i's that were tested. The estimated FIML covariance matrix also looked quite reasonable compared to the estimated 3SLS covariance matrix.

[25] The true information matrix at the final estimates is undoubtedly positive definite, but is nearly singular. In that case, very small errors in the numerical approximations to the second derivatives are sufficient to make the matrix of numerical second derivatives fail to be positive definite.

BIBLIOGRAPHY

[1] Amemiya, T. (1974), "The Nonlinear Two-Stage Least Squares Estimator," *Journal of Econometrics*, 2, 105-110.

[2] Belsley, D. A. (1979), "On the Computational Competitiveness of Full Information Maximum Likelihood and Three Stage Least Squares in the Estimation of Nonlinear, Simultaneous-Equations Models," *Journal of Econometrics*, forthcoming.

[3] __________ (1979), "On the Efficient Computation of the Nonlinear Full-Information Maximum Likelihood Estimator," mimeo.

[4] Berndt, E. K.; B. H. Hall; R. E. Hall; and J. A. Hausman (1974), "Estimation and Inference in Nonlinear Structural Models," *Annals of Economic and Social Measurement*, 3/4, 653-665.

[5] Chapman, D. R. and R. C. Fair (1972), "Full-Information Maximum Likelihood Program: User's Guide," Econometric Research Program Research Memorandum No. 137, Princeton University.

[6] Chow, G. C. (1968), "Two Methods of Computing Full-Information Maximum Likelihood Estimates in Simultaneous Stochastic Equations," *International Economic Review*, 9, 100-112.

[7] __________ (1973), "On the Computation of Full-Information Maximum Likelihood Estimates for Nonlinear Equation Systems," *Review of Economics and Statistics*, 104-109.

[8] __________ and R. C. Fair (1973), "Maximum Likelihood Estimation of Linear Equation Systems with Auto-Regressive Residuals," *Annals of Economic and Social Measurement*, 2/1, 17-28.

[9] Dagenais, M. G. (1978), "The Computation of FIML Estimates as Iterative Generalized Least Squares Estimates in Linear and Nonlinear Simultaneous Equations Models," *Econometrica*, 46, 1351-1362.

[10] Daniel, J. W. (1971), *The Approximate Minimization of Functionals*. Engelwood Cliffs, N.J.: Prentice-Hall, Inc., 1971.

[11] Dennis, J. E., Jr., and J. J. Moré (1977), "Quasi-Newton Methods, Motivation and Theory," *SIAM Review*, 19, 46-89.

[12] Fair, R. C. (1974), "On the Robust Estimation of Econometric Models," *Annals of Economic and Social Measurement*, 3/4, 667-677.

[13] __________ (1976), *A Model of Macroeconomic Activity. Volume II: The Empirical Model*. Cambridge: Ballinger Publishing Co.

[14] __________ (1978), "The Fair Model as of November 1, 1978," mimeo.

[15] __________ and W. R. Parke (1979), "Full Information Estimates of a Nonlinear Macroeconometric Model," Journal of Econometrics, forthcoming.

[16] Fletcher, R. and M. J. D. Powell (1963), "A Rapidly Convergent Descent Method for Minimization," Computer Journal, 6, 163-168.

[17] IBM (1971), SL-MATH, User's Guide, First Edition.

[18] IMSL (1979), IMSL Library, Edition 7, Houston, Texas, IMSL.

[19] Jorgenson, D. W. and J. Laffont (1974), "Efficient Estimation of Nonlinear Simultaneous Equations with Additive Disturbances," Annals of Economic and Social Measurement, 3/4, 615-640.

[20] Koopmans, T. C.; H. Rubin; and R. B. Leipnik (1950), "Measuring the Equation System of Dynamic Economics," in T. C. Koopmans (ed.), Statistical Inference in Dynamic Economic Models, Cowles Commission Monograph 10. New York: John Wiley & Sons, Ch. 2.

[21] Ortega, J. M. and M. L. Rockoff (1966), "Nonlinear Difference Equations and Guass-Seidel Type Iterative Methods," SIAM Journal of Numerical Analysis, 3, 497-513.

[22] Parke, William R. (1979), "An Algorithm for Full Information Estimation," Doctoral Dissertation, Yale University.

[23] Powell, M. J. D. (1964), "An Efficient Method of Finding the Minimum of a Function of Several Variables without Calculating Derivatives," Computer Journal, 7, 155-162.

[24] Schechter, S. (1962), "Iterative Methods for Nonlinear Problems," Trans. Amer. Math. Soc., 104, 179-189.

[25] Theil, H. (1971), Principles of Econometrics. New York: John Wiley & Sons.