Gradient methods in FIML estimation of econometric models

Calzolari, Giorgio and Panattoni, Lorenzo

IBM Scientific Center, Pisa, Italy

1985

GRADIENT METHODS IN FIML ESTIMATION OF ECONOMETRIC MODELS

Giorgio Calzolari and Lorenzo Panattoni
IBM Scientific Center, Pisa, Italy

1. Introduction

Efficient computational algorithms, to produce full information maximum likelihood estimates of the structural form coefficients in a system of simultaneous equations, have worried for a long time and are still worrying econometricians.

Several optimization techniques have been proposed in the last few years and experimented with on linear and nonlinear models of increasing size. While some techniques are search algorithms which do not make use of information on first and second derivatives (e.g., Parke, 1982), it is generally acknowledged that gradient methods, and more specifically *Newton-like* methods, which make use of such information, should be superior to the others, at least near the optimum. The drawback of *Newton-like* methods, as well pointed out in Belsley (1980, p. 222), lies in the excessive cost required in the calculation of the Hessian matrix. Therefore, methods have been proposed in the literature which replace the Hessian matrix with other matrices, like those adopted in Berndt, Hall, Hall and Hausman (1974), Amemiya (1977), or in Dagenais (1978).

Belsley's findings, after comparing the computational optimization performances of different matrices, placed the algorithm which uses the "*exact*" Hessian in a dominant position for optimization of the FIML objective function. On the other hand Dagenais' experiments showed that a gradient method in which the Hessian is replaced by a suitable approximation can be computationally more efficient that a *Newton-like* algorithm, at least as long as the *robustness* with respect to the initial guess of the coefficients is concerned.

In our Monte Carlo study, the performances of the *Newton-like* method are compared with the performances of gradient algorithms in which more easily obtainable matrices are used: the *outer product* matrix proposed in Berndt *et al.* (1974), and the *generalized least squares type* matrix discussed in Amemiya (1977, p. 963) and experimented with in Dagenais (1978).

A large set of Monte Carlo experiments is performed on models of different size and with different sample period lengths. A systematic average behavior is derived from the Monte Carlo experiments and evidenced in the paper.

Convergence with the outer product matrix is usually slow, at least as the number of iterations is concerned; simplicity in the computation of the matrix provides only a partial compensation in terms of computation time. This result is in agreement with Belsley (1980).

The convergence with the Hessian is usually faster near the optimum, again in agreement with Belsley (1980), while the *generalized least squares type* matrix works better far from it, and this is not only in terms of "*robustness*" (less chance of false convergence to a saddle point rather than a maximum), as already Dagenais (1978) noticed, but also in terms of "*gain*" inside the iterative gradient procedures. This result, which motivated this paper, was observed and measured across a large set of Monte Carlo replications on models with short sample period lengths (like models with annual data) and might be approximately quantified as follows. Whichever "*good*" starting point of the iterative maximization process was adopted, such as the point obtained from single equation estimation (least squares or instrumental variables), only when most of the distance (99% or more, on the average) between the initial point and the optimum had been covered, the convergence became faster using the Hessian.

This suggests first of all that, although the Hessian as expected performs better *near the optimum*, this "*near the optimum*" should be interpreted in a much mor restrictive sense than usually believed in practical applications.

On the other hand, the fact that the *generalized least squares type* matrix "*gains*" usually more than the Hessian near the starting point and less near the optimum might be quite useful for implementing FIML procedures. A good improvement of the computational efficiency has in fact been obtained by using a *mixed*

gradient algorithm based on the *generalized least squares type* matrix in the first iterations and on Hessian in the last iterations.

2. Three Gradient Methods

Let the system of simultaneous equations be represented as

$$f_i(y_t, x_t, a_i) = u_{it}, \quad i = 1, 2, \ldots, m; \quad t = 1, 2, \ldots, T, \quad (1)$$

where y_t is the $m \times 1$ vector of endogenous variables at time t, x_t is the vector of predetermined variables at time t and a_i is the vector of unknown structural coefficients in the ith equation. The $m \times 1$ vector of random error terms at time t, $u_t = (u_{1t}, u_{2t}, \ldots, u_{mt})'$, is assumed to be independently and identically distributed as $N(0, \Sigma)$, with Σ completely unknown, apart from being symmetric and positive definite. The complete $n \times 1$ vector of unknown structural coefficients of the system will be indicated as $a = (a_1', a_2', \ldots, a_m')'$.

The concentrated log-likelihood function is

$$l_T = \sum_t \log \|\partial f_t / \partial y_t'\| - T/2 \log \left| T^{-1} \sum_t f_t f_t' \right|. \quad (2)$$

where $f_t = (f_{1t}, f_{2t}, \ldots, f_{mt})' = u_t$ and the Jacobian determinant $|\partial f_t / \partial y_t'|$ is taken in absolute value.

A gradient iterative procedure to maximize the log-likelihood function can be represented by the formula:

$$\hat{a}^{(k)} = \hat{a}^{(k-1)} + \lambda Q \partial l_T / \partial a, \quad (3)$$

where $\hat{a}^{(k-1)}$ is the estimate of the coefficients vector obtained after $k - 1$ iterations, Q is some $n \times n$ matrix, and λ is a real number (scalar).

Gradient methods differ in the way in which the matrix Q and the scalarλ are selected at each iteration. The selection of the matrix Q determines the choice of the direction along which the search for the maximization of the log-likelihood function will be made. The choice of λ determines the step size in this direction to obtain the new values of the coefficients.

As long as the choice of Q is concerned, three different approaches have been tried.

2.1

The matrix Q is given by the inverse of the Hessian of the log-likelihood function. The analytical expression of the i, jth block of the Hessian is given in Amemiya (1977, eq. 3.5)

$$
\begin{aligned}
-\partial^2 l_T/\partial a_i \partial a_j' \;=\; & -\sum_t \partial g_{ijt}/\partial u_{it} + T\left(\sum_t g_{ijt} f_t'\right)\left(\sum_t f_t f_t'\right)_i^{-1} \\
& + \left[\sum_t (\partial g_{it}/\partial u_{jt})(\partial g_{jt}'/\partial u_{it})\right] + T\left(\sum_t f_t f_t'\right)_{ij}^{-1}\left(\sum_t g_{it}\, g_{jt}'\right) \\
& - T\left(\sum_t g_{it} f_t'\right)\left(\sum_t f_t f_t'\right)_j^{-1}\left(\sum_t f_t f_t'\right)_i^{\prime -1}\left(\sum_t f_t g_{jt}'\right) \\
& - T\left(\sum_t f_t f_t'\right)_{ij}^{-1}\left(\sum_t g_{it} f_t'\right)\left(\sum_t f_t f_t'\right)^{-1}\left(\sum_t f_t g_{jt}'\right),
\end{aligned}
\tag{4}
$$

where $g_{it} = \partial f_{it}/\partial a_i$ (in practice the vector g_{it} contains the values of the explanatory variables appearing in the ith equation, if the model is linear in the coefficients), $\partial g_{it}/\partial u_{jt} = (\partial g_{it}/\partial y_t')(\partial f_t/\partial y_t')_j^{-1}$, $\partial g_{ijt}/\partial u_{it} = (\partial g_{ijt}/\partial y_t')(\partial f_t/\partial y_t')_i^{-1}$, and a single subscript i represents the ith column of the matrix. In this case the gradient method becomes a *Newton-like* algorithm.

2.2

The matrix Q is given by the inverse of the *generalized least squares type* matrix introduced in Amemiya (1977, p. 963) and experimented with in Dagenais (1978). Such a matrix is obtained as follows. We first introduce the $T \times m$ matrix F, whose t, ith element is $f_i(y_t, x_t, a_j) = u_{it}$ (the matrix of residuals) and the matrix G_i, whose tth row is g_{it}' (in practice, for models linear in the coefficients, the matrix with the values of the explanatory variables appearing in the ith equation). We define, now,

$$
\hat{G}_i \;=\; G_i - T^{-1} F \sum_t (\partial g_{it}/\partial u_t')' \tag{5}
$$

and build the block diagonal matrix $\hat{G}$, whose m diagonal blocks are $\hat{G}_i$. The *generalized least squares* type matrix used in the gradient procedure is the inverse of the matrix

$$
[\hat{G}'(\hat{\Sigma}^{-1} \otimes I)\hat{G}] \tag{6}
$$

(for linear models, $\hat{G}$ has the form of the matrix used in Aitken–Zellner estimation, containing the values of the explanatory variables, but with the historical values of the endogenous variables replaced by the computed values).

2.3

The matrix Q is given by the inverse of the *outer product* matrix proposed in Berndt *et al.* (1974) whose i, jth block is

$$T^{-1} \sum_t \left[\partial g_{it}/\partial u_{it} - T(g_{it} f_t') \left(\sum_t f_t f_t' \right)_i^{-1} \right] \times \left[\partial g_{jt}/\partial u_{jt} - T(g_{jt} f_t') \left(\sum_t f_t f_t' \right)_j^{-1} \right]'. \tag{7}$$

The choice of the step size λ has been performed following an optimality criterion, i.e., trying to maximize the log-likelihood function by means of an univariate search in the selected direction (see also Eisenpress and Greenstadt, 1966, or Dagenais, 1978). Of course, the procedure is only based on heuristic considerations and there is no assurance that such a strategy for the selection of the value of λ is an optimal one; however, it appeared in practice to accelerate the calculations and to assure the convergence in most cases, and, therefore, it gave a good common basis for performing comparisons of the gradient algorithms using the three matrices.

For the univariate search we used a part of Powell's algorithm, as described in Pierre (1969, pp. 277–280), which does not involve the use of derivatives, but is quadratic convergent all the same. Particular care had to be used in the choice of the tolerance for the convergence in this univariate search because, although the maximization process improved the computational efficiency of the whole algorithm, this implied the evaluation of several values of the log-likelihood function. These computations, for medium and large size models, are rather time consuming and it can happen that with a too tight tolerance the algorithm requires a high number of such computations without a corresponding improvement in the efficiency of the whole algorithm. For the experimented models we found that values 0.01–0.001 of the relative tolerance on λ are usually good values for the overall computational efficiency of the maximization algorithm.

3. Experimental Comparison

Monte Carlo Experiments have been performed on four models of small medium size. Two models are linear, and two are nonlinear in variables.

(1) A multiplier-accelerator model, with three linear equations, two of which stochastic, and 6 unknown structural coefficients; the equations and empirical data can be found in Dhrymes (1970, pp. 533–534).

(2) A model for the Italian economy proposed in Sitzia and Tivegna (1975), consisting of 7 linear equations, 5 of which stochastic, and 19 unknown structural coefficients.

(3) A mildly nonlinear version of Klein-I model (six equations, three of which stochastic, and 12 unknown coefficients), obtained by replacing the linear equation for consumption with a log-linear equation (see Belsley, 1980, model 3B).

(4) The Klein-Goldberger model (Klein, 1969), which is nonlinear in variables and consists of 20 equations, 16 of which stochastic, with 54 unknown structural coefficients.

Monte Carlo experiments on all models are based on a few hundred replications, each of which has been performed as follows. Starting from the model with a given set of parameters ("*true*" coefficients and covariance matrix of the structural disturbances, held fixed in all replications), random values of the endogenous variables over the sample period are generated by means of stochastic simulation and are used for FIML estimation with the three methods.

To reproduce as much as possible the conditions under which FIML estimation is performed in practice, we choose a "*good*" starting point for each estimation by getting a preliminary single equation estimate (least squares or instrumental variables).

Several convergence criteria (on coefficients, on the likelihood and on the gradient) have been experimented with. While some differences have been encountered in several cases, the overall behavior did not change very much with the different criteria, apart from the obvious lengthening of convergence "*tails*" when adopting a very tight tolerance. The same can be said about the choice of the sample period length; the overall behavior did not change, apart from the obvious shortening of convergence "*tails*" with all methods when the sample period becomes longer. Again the overall

behavior did not change with the different choice of the predetermined variables in the sample period (exogenous variables have been either kept fixed in all experiments, or randomly generated with given means and covariance matrix, and lagged endogenous variables have also been kept fixed in all experiments, or randomly generated using dynamic stochastic simulation), and with the different choice of the "*true*" parameters of the model, on which Monte Carlo generations are based.

The simple computation of the number of iterations required to get convergence with the three matrices is not particularly illuminating (some more details can be found in Calzolari and Panattoni, 1983). The only sure indications which were obtained are the following.

(1) The use of the Hessian never requires very long tails for the convergence, while the other two matrices (the *outer product* matrix, in particular) often do.

(2) The Hessian, apart from the computational burden, rises more often than the other problems of false convergence to saddle points when it is used for the estimation of rather complex models (about one out of five cases with the Klein–Goldberger model with less than 50 observations).

Much more interesting considerations are obtained if we have a better insight in the convergence process. For each Monte Carlo replication, we first compute the maximum with a very high precision, then we measure the fraction of the distance between the starting point and the maximum covered at each iteration, with the three methods. The distance is measured both on the values of the log-likelihood and as length of the difference between the current and the final coefficient vectors. As before, in some cases the two measures give different results, but the overall behavior is practically the same. In Figure 1 results related to the distances measured on the values of the log-likelihood function are displayed on a log-scale. If we call $D(k)$ the distance which, after k iterations, still remains to get to the maximum, the value which is calculated is

$$d(k) = -\log [D(k)/D(0)]. \tag{8}$$

The value of this variable is equal to zero at the starting point, increases at any new (kth) iteration, as we move monotonically "*uphill*", and would be infinite at the optimum (in practice it

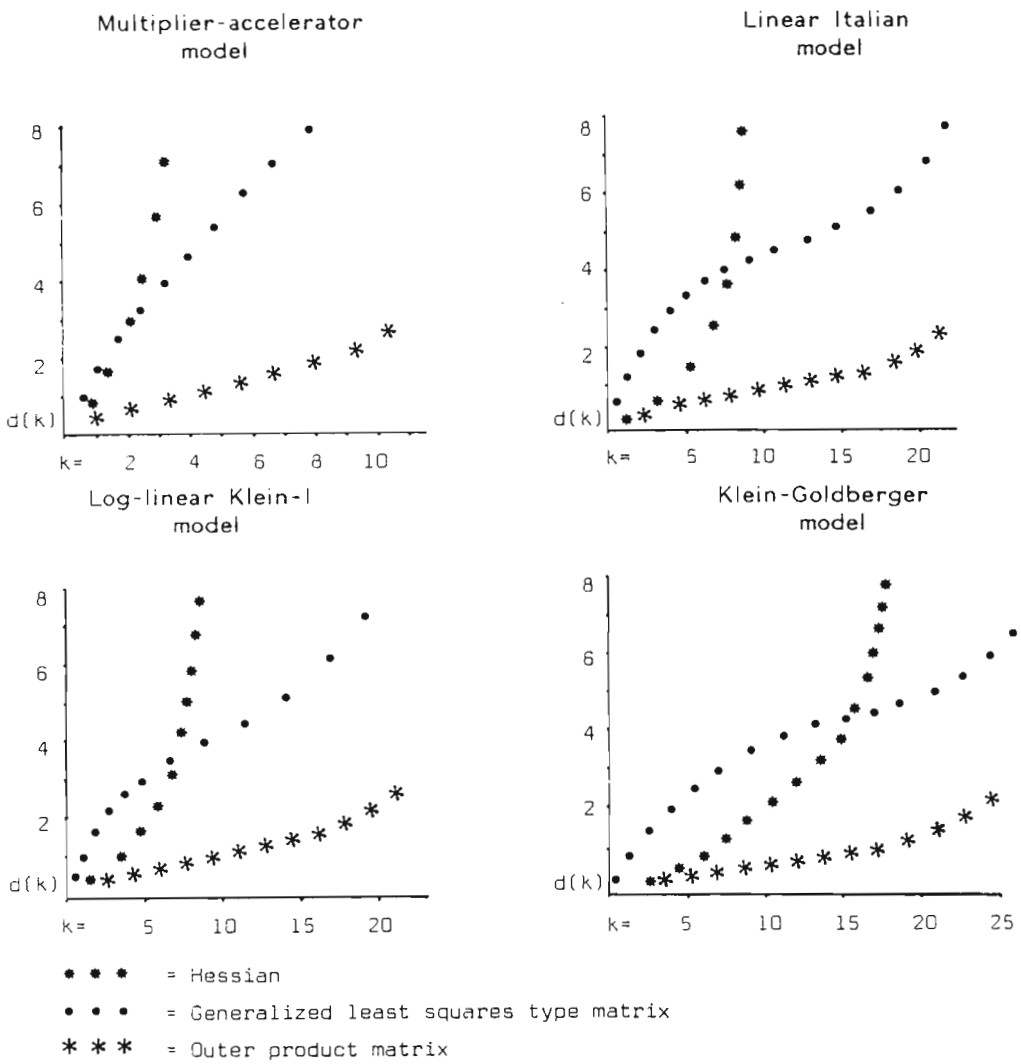

Figure 1. Average rate of convergence of the three gradient algorithms.

assumes a value of a few units, depending on the choice of the tolerance in the convergence criterion). For example, a value 4 means that the distance from the maximum of the point obtained after k iterations is 10^{-4} of the distance between the starting point and the maximum.

For each model, and for each iteration number (k), the value which is displayed in Figure 1 is the average value of all $d(k)$, across a few hundred Monte Carlo replications, obtained from using the three matrices.

An interesting systematic behavior of the three methods can be

observed for the models in Figure 1, where the length of the sample periods are those of the historical data originally proposed for the models themselves (only for the Klein–Goldberger model the sample had to be enlarged of a few observations). The gradient algorithm, which makes use of the *generalized least squares type* matrix is considerably faster in the first iterations and, on average, it allows to cover a good deal of the distance from a "*good*" starting point up to the maximum (more than 99.9% for these experiments based on rather short samples) in a smaller number of iterations that the same algorithm which makes use of the other two matrices. The dominance of the Hessian matrix becomes effective only in a very tight neighborhood of the optimum, where it allows a considerable reduction of the number of iterations.

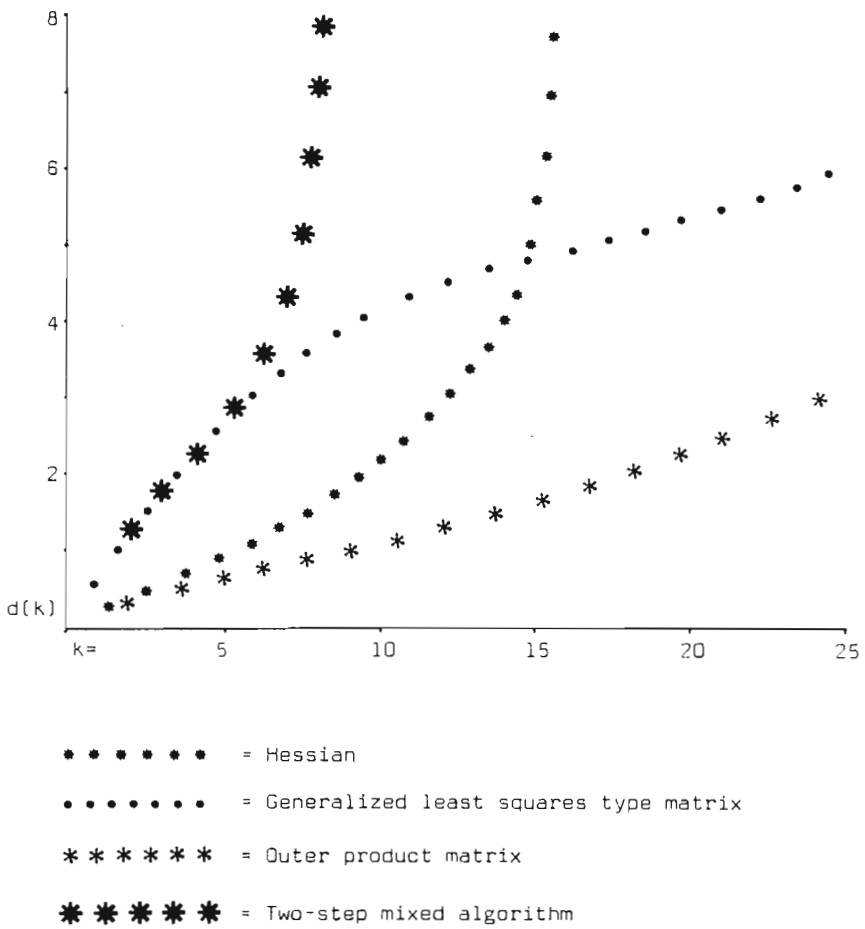

Figure 2. Average rate of convergence of the three gradient algorithms and of the mixed gradient algorithm on the Klein–Goldberger model.

This average behavior, which systematically occurs with minor variations in all the models and under all the different conditions experimented with, might be interesting for improving the computational efficiency of FIML algorithms. The use of the *generalized least squares type* matrix seems recommendable in the first iterations (it becomes even more recommendable when considering that its computation is rather simple and fast even for medium-large size models and is, in any case, considerably simpler and faster than computation of the Hessian). After a few iterations, the use of the Hessian should be preferred.

For example, since the slope of the curve related to the Hessian in Figure 1 becomes the highest when, on the average, $d(k) = 2$ (10^{-2} of the total distance still remains to get the maximum), a *two-step mixed* iterative algorithm would produce a good improvement of the computational efficiency. We first adopt a convergence criterion with a wide tolerance (for example a relative tolerance 10^{-2} on coefficients). Starting from a "*good*" initial value of the coefficients vector, we first apply iteratively the gradient method using the *generalized least squares type* matrix, until convergence is reached. We then adopt a tighter tolerance for the convergence criterion and apply iteratively the gradient method using the Hessian. A *mixed* gradient method of this kind, applied to the Klein–Goldberger model with a sample period of 50 observations, gave, on the average, the improvement of the computational efficiency evidenced in Figure 2.

References

Amemiya, T. (1977): "The Maximum Likelihood and the Nonlinear Three-Stage Least Squares in the General Nonlinear Simultaneous Equation Model", *Econometrica* **45**, 955–968.

Belsley, D. A. (1980): "On the Efficient Computation of the Nonlinear Full-Information Maximum-Likelihood Estimator", *Journal of Econometrics* **14**, 203–225.

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 Measurements* **3**, 653–665.

Calzolari, G. and L. Panattoni (1983): "Hessian and Approximated Hessian Matrices in Maximum Likelihood Estimation: A Monte Carlo Study". Pisa: Centro Scientifico IBM, paper presented at the *European Meeting of the Econometric Society*, August 29–September 2.

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.

Dhrymes, P. J. (1970): *Econometrics: Statistical Foundations and Applications.* New York: Harper & Row.

Eisenpress, H. and J. Greenstadt (1966): "The Estimation of Nonlinear Econometric Systems", *Econometrica* **34**, 851–861.

Klein, L. R. (1969): "Estimation of Interdependent Systems in Macroeconometrics", *Econometrica* **37**, 171–192.

Parke, W. R. (1982): "An Algorithm for FIML and 3SLS Estimation of Large Nonlinear Models", *Econometrica* **50**, 81–95.

Pierre, D. A. (1969): *Optimization Theory with Applications.* New York: John Wiley & Sons.

Sitzia, B. and M. Tivegna (1975): "Un Modello Aggregato dell'Economia Italiana 1952–1971", in *Contributi alla Ricerca Economica* No. 4. Roma: Banca d'Italia, 195–223.