This PDF is a selection from an out-of-print volume from the National Bureau of Economic Research

Volume Title: Annals of Economic and Social Measurement, Volume 6, number 5

Volume Author/Editor: NBER

Volume Publisher:

Volume URL: http://www.nber.org/books/aesm77-5

Publication Date: December 1977

Chapter Title: ADAPTED METHODS FOR SOLVING AND OPTIMIZING QUASI-TRIANGULAR ECONOMETRIC MODELS

Chapter Author: Pierre Nepomiastehy, Alain Ravelli

Chapter URL: http://www.nber.org/chapters/c10543

Chapter pages in book: (p. 78 - 85)

Annals of Economic and Social Measurement, 6/5, 1978

ADAPTED METHODS FOR SOLVING AND OPTIMIZING QUASI-TRIANGULAR ECONOMETRIC MODELS

BY PIERRE NEPOMIASTCHY AND ALAIN RAVELLI

This paper sets out a Newton-like method for solving the model and a method for computing the gradient (adjoint variable technique) which are both adapted to econometric models with a quasi-triangular structure (after a possible renumbering of the equations). Comparisons are made with classical methods on the 130 equations Star model.

I. INTRODUCTION

Let us consider a non-linear and dynamic macroeconomic model with n endogeneous variables. To solve this model, we propose the following method. A set of s variables, called *loop variables*, and a set of s equations, called *loop equations* are selected such as: for given values of the loop variables, the remaining model with $n - s$ equations and $n - s$ variables is triangular and can be solved directly; then, an algorithm is chosen to iterate on the values of the loop variables in order to satisfy the loop equations.

The efficiency of this method obviously depends on the number of loop variables. We propose in [1] a method to renumber the equations and the variables in order to minimize this number.

If, after a possible renumbering, s is small compared to n, then the jacobian matrix of the model is lower-quasi-triangular (i.e. most of its non zero elements are under the main diagonal) and the model is called *quasi-triangular*. In the paper, we shall consider only this kind of model but it seems to be the case for many macroeconomic models. Here are a few examples:

Model	Number of equations (n)	Number of loop variables (s)
Andomini [2]	4	1
Pimpon [3]	14	2
Fair [4]	83	7
Star [5]	130	3
DMS [6]	about 1000	less than 100

The quasi-triangular structure was obtained by hand for Andomini and Pimpon, by the algorithm described in [1] for Fair, and was provided by the author of Star; for DMS, no renumbering was made.

In section III, we propose an adapted Newton method easy to use and we give numerical results using comparisons with Gauss-Seidel.

We have shown in [7] on the small Pimpon model that the adjoint variable method is more effective for computing the gradient than the generally used finite difference method proposed by Fair [8]. We propose in section IV an adaptation of this adjoint variable method to the quasi-triangular structure of the model and we give numerical results of comparisons with the finite difference method made using the Star model.

These comparisons provide most encouraging results to support the efficiency of the two proposed methods.

II. Structure of the Model

For sake of clarity of the presentation, we shall simplify the structure of the model in the sense that we are not going to distinguish between equations which do not depend on loop variables and which correspond to pre-determined variables, equations which depend on loop variables (heart of the model) and the triangular set of output equations. The methods proposed in sections III and IV can easily be adapted to take into account these differences between equations (see [1]).

Let us consider a discrete time dynamic model with periods t going from 1 to T. Let n be the number of endogeneous variables, x_t^i be the value of variable i at period t and x_t be the vector $\{x_t^1, \ldots, x_t^n\}$. Let r be the number of control variables, u_t^i be the vector $\{u_t^1, \ldots, u_t^r\}$. Let p and q be the maximum lag appearing in the model on (respectively) endogeneous and control variables. Then, the model is described by the following set of equations:

$$\text{(1a)} \qquad x_t^i = f_t^i(x_t, x_{t-1}, \ldots, x_{t-p}, u_t, u_{t-1}, \ldots, u_{t-q}),$$

$$i = 1, \ldots, n, \qquad t = 1, \ldots, T$$

$$\text{(1b)} \qquad x_t^i \text{ given for } i = 1, \ldots, n \quad \text{and} \quad t = -p+1, -p+2, \ldots, -1, 0$$

We shall assume that the given functions f_t^i are continuously differentiable and that, at least for any control taken in a reasonable range, the system (1) has a unique solution.

For solving the model at period t, the vector:

$$\text{(2)} \qquad e_t = \{x_{t-1}, x_{t-2}, \ldots, x_{t-p}, u_t, u_{t-1}, \ldots u_{t-q}\}$$

is known and the problem is to find the solution x_t of the problem:

$$\text{(3)} \qquad x_t^i = f_t^i(x_t, e_t), \quad i = 1, \ldots, n$$

After a possible renumbering of equations and variables, the loop equations are the s last equations of the model and $y_t = \{x_t^{n-s+1}, \ldots, x_t^n\}$ is the vector of the loop variables. Then, by definition of the loop variables, the system (3) is triangular for any given y_t and (3) can be written:

(4a) $\qquad x_t^i = f_t^i(x_t^1, \ldots, x_t^{i-1}, y_t, e_t), \quad i = 1, \ldots, n - s$

(4b) $\qquad x_t^i = f_t^i(x_t, e_t), \qquad i = n - s + 1, \ldots, n$

III. Solution of the Model

For any given y_t, the system (4a) is triangular and can be solved by simple evaluations of functions f_t^i, which gives us values of x_t^i for any $i \leq n - s$. Since $y_t = \{x_t^{n-s+1}, \ldots, x_t^n\}$, it can be seen that, taking (4a) into account, x_t can be considered as a function $x_t(y_t)$ of y_t. Of course, with an arbitrary value of y_t, $x_t(y_t)$ does not in general satisfy equations (4b), there is an error which depends on and only on y_t. We shall denote $\varphi_t(y_t)$ this error. The problem is to find y_t in such a way that this error is equal to zero:

(5) $\qquad \varphi_t^i(y_t) = 0, \quad i = 1, \ldots, s$

In theory, it is possible to eliminate variables x_t^i, $i < n - s$, from equations (4) and obtain the *analytical* expressions of functions $\varphi_t^i(y_t)$. For large models, this is obviously impossible for practical reasons but it is clear that, using (4), one is able to compute the numerical value of $\varphi_t(y_t)$ for any given numerical value of y_t. In this case, the partial derivatives of φ_t^i cannot be analytically computed and the simplest method for solving (5) is the well-known Gauss-Seidel algorithm.

It should be noted that this Gauss-Seidel algorithm, applied to the system (5) of dimension s ($s = 3$ for Star) and not to the total system (3) of dimension n ($n = 130$ for Star) is adapted to the quasi-triangular structure of the model. But, still, it has the usual disadvantages of the Gauss-Seidel methods, namely:

a) the convergence is slow;*

b) the convergence is strongly dependent on the ordering and normalization of the equations and on the weights chosen for the feedback, weights which can be determined for each model only by a large number of random tests.

We propose solving (5) by using the Newton method which has a convergence rate of 2** and no parameters to determine in the feedback rule. It should be noted that if the Newton method had been applied directly to the total model (3), then each iteration would have required the solution of a linear system of dimension n. Using a general package

*Its convergence rate is 1: the error at iteration k is proportional to the error at iteration $k - 1$.

**The error at iteration k is proportional to the square of the error at iteration $k - 1$.

to solve this linear system, we have obtained very poor results. This linear system is sparse and can be solved by adapted methods like those proposed by Drud [9] and we shall wait until Drud's routine is available to see if the "global Newton method" with the sparse technique for solving the linear system is more effective than our method or not.

Applying the Newton method to (5), at each iteration we have only to solve a linear system of dimension s (with $s = 3$ for Star). Consequently, the only problem is the computation of the partial derivatives $\partial\varphi_i^i/\partial y_t^j$. If an analytical derivation package (as Formac) is available, then the analytical expressions of the derivatives $\partial f_t^i/\partial x_t^i$ can be deduced from (4). By definition of $\varphi_t(y_t)$, we have:

$$\text{(6)} \qquad \varphi_t^i(y_t) = y_t^i - f_t^{n-s+i}[x_t(y_t), e_t], \quad i = 1, \ldots, s$$

with $y_t^i = x_t^{n-s+i}$; from (4a) we have:

$$\text{(7)} \qquad \frac{\partial x_t^i}{\partial y_t^j} = \sum_{k=1}^{i-1} \frac{\partial f_t^i}{\partial x_t^k}\frac{\partial x_t^k}{\partial y_t^j} + \frac{\partial f_t^i}{\partial y_t^j}, \quad i = 1, \ldots, n-s, \quad j = 1, \ldots, s$$

which for any j, is a triangular system in $\partial x_t^i/\partial y_t^j$, $i = 1, \ldots, n-s$.

Then, from derivation of (6), we have:

$$\text{(8)} \qquad \frac{\partial \varphi_t^i}{\partial y_t^j} = \delta_{ij} - \sum_{k=1}^{n-s} \frac{\partial f_t^{n-s+i}}{\partial x_t^k}\frac{\partial x_t^k}{\partial y_t^j} - \frac{\partial f_t^{n-s+i}}{\partial y_t^j}, \quad i, j = 1, \ldots, s$$

If no analytical derivation package is available, then finite difference approximations of the $\partial\varphi_t^i/\partial y_t^j$ are computed. The computation of $\varphi_t(y_t)$ for $s + 1$ values of vector y_t, that is $s + 1$ solutions of the triangular system (4), gives an approximation of these derivatives; for more details, see [1].

With programs written in Fortran H extended on an IBM 370-168 computer and with a unit of one millisecond of CPU time, Table 1 gives, for different values of the required accuracy, the computing time of one simulation over 10 periods of the 130 equations model Star.

TABLE 1

ϵ	GS	N1	N2
10^{-2}	14.7	9.6	12.0
10^{-4}	51.9	14.5	18.5
10^{-6}	129.0	20.2	20.6
10^{-8}	265.9	21.9	26.3

GS: Gauss-Seidel method.
N1: Newton method with analytical derivation of the model.
N2: Newton method with numerical derivation of the model.
ϵ: accuracy required.

Remember that the model is solved when $\varphi_t(y_t) = 0$, with $\varphi_t(y_t)$ given by formula (6). We have chosen as the test to stop the algorithm:

$$\sum_{i=1}^{s} \left| \frac{f_i^{n \cdot s+i} - y_i^i}{y_i^i} \right| \le \epsilon \tag{9}$$

All the methods tested were adapted to the quasi-triangular structure of the model. A run of the "global Newton method" discussed above with a standard package for the linear system produced disastrous results: 13 *seconds* for $\epsilon = 10^{-8}$ (50 times worse than GS).

The Gauss-Seidel method chosen for comparison was the one described above, namely Gauss-Seidel applied to (5), *after optimization* of the $s = 3$ weights of the feed-back rule. It is obvious that Gauss-Seidel applied to the total system (3), even after the optimization of the $n = 130$ weights of the feed-back rule, would have given worse results.

In methods N1 and N2, the jacobian $((\partial\varphi_t^i/\partial y_t^j))$ was computed for each iteration of the Newton methods. Computing it only sometimes gives a small improvement (for more details, see [1]).

From table 1, it appears obvious that when the Newton method is adapted to the quasi-triangular structure of the model, then this method is much more effective than the Gauss-Seidel, especially when a high accuracy is required, which is the case when the simulation algorithm is only a part of an optimization problem solved by sophisticated algorithms like Davidon-Fletcher-Powell.

IV. Optimization of the Model

We have to minimize the following loss function:

$$j(x, u) = \sum_{t=1}^{T} j_t(x_t, x_{t-1}, \ldots, x_{t-p}, u_t, u_{t-1}, \ldots, u_{t-q}) \tag{10}$$

where x and u are linked by the model (1). We shall assume that the functions j_t are continuously differentiable.

In this paper, we are only concerned by the search for an efficient algorithm to compute the gradient $J'(u)$ of $J(u) = j[x(u), u]$, where $x(u)$ is the solution of (1) associated with the control u.

The finite difference method [8] is a very simple method for gradient computation, but it requires rT model simulations for one gradient computation, where r is the control dimension and T the number of periods. We propose using the adjoint-variable method. This method is well known in optimal control theory (see, for example [10]) but, as far as we know, was applied to the optimization of a macroeconomic model for the first time by us [7]. In [7], we have shown how it can be applied to the 14-equations Pimpon model: here we show how it can be adapted to the quasi-triangular structure of the model and we give results for the 130-equations Star model.

To describe the adjoint-variable method, let us introduce several notations:

(11) $$\delta_t = \begin{cases} 1 & \text{if } t \le T \\ 0 & \text{if } t > T \end{cases}$$

(12) $$y_t^i = \sum_{k=0}^{p} \delta_{t+k} \frac{\partial j_{t+k}}{\partial x_t^i}, \quad i = 1, \ldots, n, \quad t = 1, \ldots, T$$

(13) $$w_t^i = \sum_{k=0}^{q} \delta_{t+k} \frac{\partial j_{t+k}}{\partial u_t^i}, \quad i = 1, \ldots, r, \quad t = 1, \ldots, T$$

(14) $$F_t^k = \left(\left(\frac{\partial f_{t+k}^i}{\partial x_t^j}\right)\right); \quad G_t^k = \left(\left(\frac{\partial f_{t+k}^i}{\partial u_t^j}\right)\right)$$

Let $\psi = \{\psi_1, \ldots, \psi_T\}$, with $\psi_t \in R^n$, be the solution of:

(15) $$\psi_t = \sum_{k=0}^{p} \delta_{t+k} (F_t^k)^* \psi_{t+k} + y_t, \quad t = 1, \ldots, T$$

where $(F_t^k)^*$ is the transposed matrix of F_t^k. The equation (15) is called the adjoint system of the optimization problem and its solution ψ is called the adjoint-variable. It can be proven (see [1]) that the gradient can be deduced from the value of ψ by the formula:

(16) $$J_t'(u) = w_t + \sum_{k=0}^{q} \delta_{t+k} (G_t^k)^* \psi_{t+k}, \quad t = 1, \ldots, T$$

For the gradient computation, one must solve the system (15) for $t = T$, then for $t = T - 1$, and so on till $t = 1$. Indeed, when ψ_t is computed, the values of $\psi_{t'}$, for any $t' > t$ are known, hence the vector:

(17) $$c_t = y_t + \sum_{k=1}^{p} \delta_{t+k} (F_t^k)^* \psi_{t+k}$$

is known (from (11) we see that, for $t = T$, $c_t = y_t$) and computing ψ_t is reduced to the solution of the linear system:

(18) $$\psi_t = (F_t^0)^* \psi_t + c_t$$

From the notation (14) and the structure of the model given by (4), it can be seen that (18) can be written:

(19a) $$\psi_t^i = \sum_{j=i+1}^{n} \frac{\partial f_t^j}{\partial x_t^i} \psi_t^j + c_t^i, \quad i = 1, \ldots, n - s$$

(19b) $$\psi_i^i = \sum_{j=1}^{n} \frac{\partial f_i^j}{\partial x_i^j} \psi_i^j + c_i^i, \quad i = n - s + 1, \ldots, n$$

Using the same argument as for solving the model, let $\lambda = \{\psi_i^{n-s+1}, \ldots, \psi_i^n\}$ be the vector of the adjoint loop variables. For any given λ, using (19a), it is possible to compute the corresponding values of ψ_i^{n-s}, then ψ_i^{n-s-1} and so on till ψ_i^1 and check whether these values satisfy (19b). Consequently, we are able to compute the errors:

(20) $$E_i(\lambda) = \lambda_i - \sum_{j=1}^{n} \frac{\partial f_i^j}{\partial x_i^j} \psi_i^j(\lambda) - c_i^{n-s+i}, \quad i = 1, \ldots, s$$

and the only problem is to find λ such as $E(\lambda) = 0$. From (19), it is clear that $\lambda \rightarrow E(\lambda)$ is a linear mapping which can be denoted by $E(\lambda) = A\lambda - b$. The computation of $E(\lambda)$ for $\lambda = 0$ gives the vector b. It can be proven* that the matrix A is the transposed of the jacobian matrix of the system (5), hence A is known. Consequently, the system $A\lambda = b$, which is only of dimension s, has to be solved, then, the adjoint variable ψ_i is computed with the help of (19a) and, finally, the gradient is obtained by the formula (16).

The adjoint variable method requires, for one gradient computation, $2T$ evaluations of equations (19) and T solutions of a linear system of dimension s: on the other hand, the finite difference method requires $rT^2/2$ solutions of the non-linear system (3) of dimension n**. Consequently, it is clear that the adjoint variable method is much more effective as shown on Table 2 which gives, for the model Star, the computing time*** of a gradient computation using the two methods:

TABLE 2

T	$M1$	$M2$	$M1/M2$
10	0.65 sec.	0.016 sec.	40.6
20	2.39 sec.	0.031 sec.	77.0
30	4.90 sec.	0.046 sec.	106.5

T : number of simulation periods.
$M1$: gradient computing time using the finite difference method.
$M2$: gradient computing time using the adjoint variable method.
$M1/M2$: ratio of computing times.

*this result was kindly suggested to us by one of the referees of this paper. For proof, see [1].

**for Star, we have $r = 10$, $n = 130$, $s = 3$ and $T > 10$.

***with programs in Fortran H extended on an IBM 370-168 computer.

In this comparison, the matrices F_t^k and G_t^k were obtained by analytical derivation of the equations of the model. A finite difference approximation of these matrices multiply the computing time of method $M2$ by a factor 10 (independently of T) and the adjoint variable method still remains much powerful that the finite difference method.

IRIA-LABORIA
Rocquencourt, 78150 Le Chesnay
France

References

[1] D. Gabay; P. Nepomiastchy; M. Rochdi; A. Ravelli: "Etude, résolution et optimisation de modèles macroéconomiques", Rapport IRIA, 1978.
[2] K. Wall: *NBER Cambridge Memo*, June 26, 1975.
[3] A. Coutiere: "Un modèle du système monétaire français," *Statistiques et Etudes Financières*, INSEE (Paris), 1975/17.
[4] R. C. Fair: "*A Model of Macroeconomic Activity*," Vol. II. The Empirical Model, Ballinger Publishing Company, 1976.
[5] J. Mistral: "STAR, Modèle de prévision à court/moyen terme," *Séminaire sur les modèles macroèconomiques* (CNRS-NSF), Gif/s/Yvette (France), Nov. 22 26, 1976.
[6] D. Fouquet: "Les multiplicateurs de DMS," Publ. INSEE, Jan. 24, 1977.
[7] P. Nepomiastchy: "Méthodes d'optimisation adaptées aux modèles macroèconomiques," *International Symposium on New Trends in System Analysis*, Versailles (France), Dec. 13 17, 1976.
[8] R. C. Fair: "On the Solution of Optimal Problems as Maximization Problems," *Annals of Economic and Social Measurement*, Vol. 3, No. 1, Jan. 1974, pp. 135 154.
[9] A. Drud: "Method for Control of Complex Dynamic Systems," *IMSOR*, No. 27, Lyngby (Danemark), 1976.
[10] A. Bensoussan; E. Hurst; B. Naslund: "*Management Applications of Modern Control Theory*," North-Holland, 1974.