

IMF Working Paper

© 1996 International Monetary Fund

This is a *Working Paper* and the author(s) would welcome any comments on the present text. Citations should refer to a *Working Paper of the International Monetary Fund*. The views expressed are those of the author(s) and do not necessarily represent those of the Fund.

WP/96/106

INTERNATIONAL MONETARY FUND

Research Department

**A Robust and Efficient Method for Solving
Nonlinear Rational Expectations Models**

Prepared by Michel Juillard and Douglas Laxton¹

Authorized for distribution by Peter Isard

September 1996

Abstract

The development and use of forward-looking macro models in policymaking institutions has proceeded at a pace much slower than predicted in the early 1980s. An important reason is that researchers have not had access to robust and efficient solution techniques for solving nonlinear forward-looking models. This paper discusses the properties of a new algorithm that is used for solving MULTIMOD, the IMF's multicountry model of the world economy. This algorithm is considerably faster and much less prone to simulation failures than to traditional algorithms and can also be used to solve individual country models of the same size.

JEL Classification Numbers:

C63, C88, D84, E17

¹This paper was presented at the Second International Conference on Computing and Economics and Finance, Geneva Switzerland, 26-28 June 1996. We thank Raouf Boucekkine, Jean-Pierre Laffargue, Pierre Malgrange, and Steve Symansky for encouraging us to do work in this area and Peter Hollinger for incorporating the solution algorithm into portable TROLL. We also thank Helen Hwang and Susanna Mursula for their excellent technical assistance. Juillard is assistant professor at the University Paris 8, and a researcher at CEPREMAP, Paris. Laxton is an economist in the Economic Modeling and External Adjustment Division of the Research Department. The views expressed in this paper are those of the authors and do not necessarily reflect those of the International Monetary Fund.

Contents

	Page
Summary	3
I. Introduction	4
II. The L-B-J Method For Solving Nonlinear Rational Expectations Models	6
A. The Basic Problem and Notation	6
B. Solving the System With the Newton-Raphson Method	8
C. Using the Sparsity of the Jacobian Blocks	11
III. An Illustrative Example of the L-B-J Method	11
IV. The Fair-Taylor Method	18
V. L-B-J Versus F-T on a Mark II Version of MULTIMOD	19
VI. Experiences With More Recent Versions of MULTIMOD	23
VII. Conclusions	23
Text Tables	
1. L-B-J Method Versus Fair-Taylor in TROLL	27
2. L-B-J Method Versus Fair-Taylor in TROLL	28
3. L-B-J Method Versus Fair-Taylor in TROLL	29
4. L-B-J Method Versus Fair-Taylor in TROLL	30
Charts	
1. Small Model Results for Inflation	14
2. Small Model Results for Short-Term Interest Rate	15
3. Small Model Results for Real GDP	16
4. Small Model Results for Long-Term Interest Rate	17
5. Multimod Results for the Value of the Deutsche Mark	21
6. Multimod Results for the Value of the Deutsche Mark	22
References	24

Summary

The development and use of forward-looking macro models in policymaking institutions has proceeded at a pace much slower than predicted in the early 1980s. An important reason is that researchers have not had access to robust and efficient solution techniques for solving nonlinear forward-looking models. The numerical complexity of solving a forward-looking macro model is considerably more onerous than solving a traditional backward-looking, reduced-form model of the same size. In many cases, researchers have been forced either to linearize their models or to focus their attention on very small models that could be solved easily with available technology. This paper discusses the development and implementation of a new algorithm based on a Newton-Raphson iterative method. It is used for solving MULTIMOD, the IMF's multicountry model of the world economy. This algorithm is considerably faster and much less prone to simulation failure than traditional algorithms and can also be used to solve individual country models of the same size.

I. INTRODUCTION

Advances in applied econometric modeling over the past four decades have been greatly facilitated by revolutionary changes in computer hardware and software. With these advances, policy authorities in many countries have come to rely on large-scale macroeconomic models as one of the standard sources of information from which they regularly seek guidance in attempting to maintain macroeconomic stability.

The generic characteristics of macroeconomic models have evolved over time in parallel with mainstream macroeconomic thinking. As economic theorists and policy makers have become increasingly enamored with the assumption that economic agents behave in a rational, forward-looking manner, a generation of backward-looking econometric models has been increasingly retired in favor of a new generation of models with forward-looking expectations. To date, the development of the new generations has been impeded by the numerical complexity of solving forward-looking models. This difficulty has been substantially reduced, however, with the recent introduction of more powerful solution algorithms.

The numerical complexity of solving a small scale rational expectations (RE) model is equivalent to solving very large scale backward-looking econometric models. In order to reduce the size of the problem to manageable proportions, traditional algorithms have been designed to break large simultaneous blocks into smaller pieces and then use an iterative procedure to ensure consistency across blocks until the full system has converged.

The traditional Fair-Taylor (F-T 1983) algorithm separated the problem into three types of iterations and a Gauss-Seidel iterative procedure was relied upon to solve for each layer of iteration. The problems with this approach are well-known; it can be a time consuming process and the method may not find a solution even though a well-defined saddle point stable solution may exist—see Armstrong et al (1995). In order to achieve a solution with these techniques, practitioners have been forced to rely heavily on certain tuning parameters (ordering, convergence tolerance limits, damping factors, divergence factors etc.) to help the algorithms achieve convergence in a reasonable amount of time. For certain classes of models, this problem has been so severe that these models simply could not be relied upon to provide solutions in a production-related environment.

In practice, this has meant that model builders have had to either linearize their models or restrict their attention to issues that could be dealt with more easily with traditional algorithms. With the enormous advance in computer technology over the last few years, it is now possible to design more robust methods for solving medium-sized nonlinear rational expectations models.

In this paper, we investigate the efficiency gains of using a Newton-Raphson approach for solving MULTIMOD, the IMF's multicountry model of the world economy. This new approach involves stacking the equations—or combining the Type I and Type II iterations in F-T algorithm—and then employing a method first proposed by Laffargue (1990), then developed by

Boucekkine (1995) and Juillard (1996), to exploit information about the repetitive and sparse structure of the full simultaneous problem.²

In order to provide an estimate of the potential efficiency gains we compare the simulation times of the Laffargue-Boucekkine-Juillard (L-B-J) method with those of the F-T algorithm. Because solution times depend on the tightness of the tolerance limits these comparisons are made with both loose and tight convergence tolerances for the F-T algorithm. The L-B-J and F-T techniques are both implemented in TROLL and are solved on an IBM RS/6000 work station.

Our results will be encouraging to economists who are interested in building medium-sized macro models designed for policy analysis. First, we find significant savings in terms of time when compared to the traditional F-T algorithm even when fairly loose conventional convergence tolerance limits are allowed for F-T. Second, we show that when the F-T convergence tolerance is tightened sufficiently to approximately replicate the L-B-J results, the relative time savings of the L-B-J algorithm become enormous. This result is surprising because all these tests were performed on an early version of MULTIMOD that was developed with the F-T algorithm. This model is approximately linear in the sense that it generally only takes a few L-B-J iterations to achieve full convergence for a large set of shocks. We also consider a few alternative cases where MULTIMOD development has been impeded because F-T has had more difficulty in finding solutions. Here the relative comparison becomes more difficult because it depends on how adept the user is at tuning certain parameters to achieve convergence and this depends on the particular shock and model that is under consideration.

The remainder of this paper is organized as follows. In Section II, we explain the basic problem and a new Newton-Raphson based solution method suggested by Laffargue (1990), Boucekkine (1995) and Juillard (1996). Section III then provides an illustrative example of how the L-B-J algorithm works for solving a small nonlinear model of the output-inflation process. Section IV provides a very brief discussion of the F-T algorithm. In Section V we use MULTIMOD, the Fund's macroeconomic model of the world economy, to compare the simulation performance of the L-B-J algorithm with the F-T algorithm. In Section VI, we provide examples of extensions to models like MULTIMOD that have been difficult to implement with the F-T algorithm. Section VI provides some conclusions.

²The original prototype for the L-B-J algorithm was written in GAUSS for a PC (Juillard, 1996). It has recently been integrated in a much more efficient manner into Portable TROLL by Peter Hollinger at Intex Solutions.

II. THE L-B-J METHOD FOR SOLVING NONLINEAR RATIONAL EXPECTATIONS MODELS

A. The Basic Problem and Notation

We consider nonlinear models with n endogenous variables, y_{it} , $i=1, \dots, n$, m exogenous variables, x_{jt} , $j=1, \dots, m$, and p parameters, θ_k , $k=1, \dots, p$. We use the notation y_t for the vector of dependent variables at time t , x_t the vector of exogenous variables, and θ the vector of parameters.

The model is such that the n current values of y_t depend on previous and future values of the endogenous variables, the exogenous variables and the parameters. Exogenous variables may appear themselves with leads or lags, but this will not change the nature of the problem to be solved. For simplicity, we can consider only current exogenous variables without affecting the generality of the discussion.

Any model with endogenous variables appearing with multiple leads (y_{t+i} for $i > 1$) or lags can be transformed into an equivalent model with variables that appear only with single leads and lags through the use of defining auxiliary variables. For this reason, and without loss of generality, here we only consider models with leads and lags of one period.

For the system of current endogenous variables to be exactly determined there must be *an equation for each of the n current endogenous variables*. For convenience, we express this system in the following way:

$$\begin{aligned} g_1(y_{t-1}, y_t, y_{t+1}, x_t, \theta) &= 0 \\ &\vdots \\ g_n(y_{t-1}, y_t, y_{t+1}, x_t, \theta) &= 0 \end{aligned}$$

In most problems, not all of the n endogenous variables appear with a lead or a lag. However, a basic identification condition of the model requires that they all appear as a current variable. The fact that a variable does not appear with a lead or a lag can be exploited by the algorithm.

For a given sequence of exogenous variables, x_t , $t=1, \dots, T$, and a given set of parameters θ , assume that a unique trajectory exists for y for the time span between 1 to T . Obviously, this

particular trajectory will correspond to specific initial conditions y_0^* and terminal conditions y_{T+1}^* .³ Conditions for the existence of a single saddle-point stable trajectory are discussed in Blanchard and Kahn (1980) and Boucekine (1995). For the purpose of this paper, we will assume that these stability conditions are satisfied.

As in a particular simulation which takes the exogenous variables x_t , $t=1, \dots, T$ and the parameters θ as given, one can rewrite the equations of the model in the following compact form:

$$f(z_t) = \begin{bmatrix} g_1(y_{t-1}, y_t, y_{t+1}, x_t, \theta) \\ g_n(y_{t-1}, y_t, y_{t+1}, x_t, \theta) \end{bmatrix} \quad t=1, \dots, T$$

where $z_t = [y'_{t-1}, y'_t, y'_{t+1}]'$. For the initial and terminal conditions, we write:
This is a standard two-point boundary value problem which is best solved simultaneously for all

$$\begin{aligned} f_0 &= y_0 - y_0^* = 0 \\ f_{T+1} &= y_{T+1} - y_{T+1}^* = 0 \end{aligned}$$

the equations and all the periods at once.⁴ For this reason, we build Y , the vector of the values of the dependent variables for all the periods stacked-up as $Y' = [y'_0, y'_1, \dots, y'_T, y'_{T+1}]$.

The entire system with $(T+2) \times n$ equations can now be written as

³In this paper, we ignore the issue about how to solve for the true terminal conditions and just impose artificial base-line estimates. There are two general approaches to this problem. The first, which was suggested by Fair and Taylor (1983), just adds an additional iterative scheme to the basic algorithm and this iterative method is then used to eventually buildup the true model-consistent terminal conditions. The alternative is to derive a steady-state analogue model and then use this model to compute the true terminal conditions—for an example of this methodology see Black, et al (1994).

⁴Sections 4 and 5 discusses the problems associated with breaking the full simultaneous problem into smaller pieces—see also Armstrong et al (1995) and Pioro et al (1996).

$$F(Y) = \begin{bmatrix} f_0(y_0) \\ f_1(z_1) \\ \vdots \\ f_T(z_T) \\ f_{T+1}(y_{T+1}) \end{bmatrix} = 0$$

B. Solving the System With the Newton-Raphson Method

Conceptually the Newton-Raphson method is quite simple. It consists of taking analytical derivatives and then deriving linear approximations of the system $F(Y) = 0$. At each step, the vector Y is modified by an amount ΔY .

$$\Delta Y = - \left[\frac{\partial F}{\partial Y} \right]^{-1} F(Y)$$

The difficulty with this approach resides with the size of the Jacobian $\frac{\partial F(Y)}{\partial Y}$ which is a matrix of dimension $[n \times (T+2)] \times [n \times (T+2)]$, where n is the number of equations and T the number of simulation periods. Note that it is necessary to take into account, the space necessary for initial and terminal conditions. For a model with 500 equations, which is about the size of the various versions of MULTIMOD (including auxiliary equations necessary to handle multiple leads and lags), simulated over 100 periods, it would be a matrix 51,000 x 51,000.⁵ However, as is shown in Laffargue (1990), the structure of this matrix is such that its triangularization can be handled recursively and that there is no need to store the entire Jacobian matrix at any time. In fact, as we will show below, only a matrix 50,000 x 112 is required, i.e. a matrix with as many rows as there are equations multiplied by the number of periods in the simulations and as many columns as there are variables with lead (including auxiliary variables for multiple leads) in the model. Furthermore, the design of the algorithm makes it easy to store it on a hard disk and does not require excessive memory requirements for the algorithm to be functional.

The equation for the computation of the improvement of Newton-Raphson can be written in a way which illuminates the particular structure of the Jacobian:

⁵As shown is in Section IV, this method can efficiently handle problems that are substantially larger than this. Indeed, one enormous advantage of this algorithm compared to some existing algorithms is that solution speed is approximately a linear function of the simulation horizon.

$$\begin{bmatrix} I & & & & & & \\ L_1 & C_1 & F_1 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & L_t & C_t & F_t & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & L_T & C_T & F_T \\ & & & & & I & \end{bmatrix} \Delta Y = - \begin{bmatrix} 0 \\ f_1(z_1) \\ \vdots \\ f_t(z_t) \\ \vdots \\ f_T(z_T) \\ 0 \end{bmatrix}$$

where L_t , C_t and F_t are the partial Jacobians:

$$L_t = \frac{\partial f_t(z_t)}{\partial y_{t-1}}$$

$$C_t = \frac{\partial f_t(z_t)}{\partial y_t}$$

$$F_t = \frac{\partial f_t(z_t)}{\partial y_{t+1}}$$

As already mentioned, for most applications the partial Jacobians have many empty columns and we can take advantage of this special sparse structure to improve the efficiency of the algorithm.

The basic approach of the L-B-J algorithm is to eliminate the elements either below or above the main diagonal so that the model solution can proceed recursively either backwards or forwards. For illustrative purposes, we will eliminate the elements below the main diagonal. The handling of the first period is special because the existence of initial conditions. We have the equation:

$$L_1 \Delta y_0 + C_1 \Delta y_1 + F_1 \Delta y_2 = -f_1(z_1)$$

The initial conditions give us $\Delta y_0 = 0$ and we can solve the linear problem for Δy_1 :

$$\Delta y_1 + C_1^{-1} F_1 \Delta y_2 = -C_1^{-1} f_1(z_1)$$

After this first step, the system looks like:

$$\begin{bmatrix} I & & & & & \\ & I & M_1 & & & \\ & L_2 & C_2 & F_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & L_T & C_T & L_T \\ & & & & & I \end{bmatrix} \Delta Y = \begin{bmatrix} 0 \\ d_1 \\ -f_2(z_2) \\ \vdots \\ -f_T(z_T) \\ 0 \end{bmatrix}$$

where

$$\begin{aligned} M_1 &= C_1^{-1} F_1 \\ d_1 &= -C_1^{-1} f_1(z_1) \end{aligned}$$

For the second period, and all the following ones, the basic equation is:

$$L_t \Delta y_{t-1} + C_t \Delta y_t + F_t \Delta y_{t+1} = -f_t(z_t)$$

We retrieve the value of Δy_{t-1} from the equation for the previous period:

$$\Delta y_{t-1} + M_{t-1} \Delta y_t = d_{t-1}$$

By substitution and eliminating the term $L_t \Delta y_{t-1}$, we obtain

$$(C_t - L_t M_{t-1}) \Delta y_t + F_t \Delta y_{t+1} = -f_t(z_t) - L_t d_{t-1}$$

Then, solving for Δy_t ,

$$\Delta y_t + (C_t - L_t M_{t-1})^{-1} F_t \Delta y_{t+1} = -(C_t - L_t M_{t-1})^{-1} (f_t(z_t) + L_t d_{t-1})$$

After triangularization, the system looks like:

$$\begin{bmatrix} I & & & & \\ & I & M_1 & & \\ & & \ddots & \ddots & \\ & & & I & M_T \\ & & & & I \end{bmatrix} \Delta Y = \begin{bmatrix} 0 \\ d_1 \\ \vdots \\ d_T \\ 0 \end{bmatrix}$$

where $M_t = (C_t - L_t M_{t-1})^{-1} F_t$ and $d_t = -(C_t - L_t M_{t-1})^{-1} (f_t(z_t) + L_t d_{t-1})$. The value of ΔY are then easily obtained through backward substitution:

$$\Delta y_t = d_t - M_t \Delta y_{t+1}$$

Using this approach, the only blocks requiring storage are M_t and d_t $t = 1, \dots, T$. As already mentioned, further reduction in storage is obtained by taking into account the empty columns of the partial Jacobians.

C. Using the Sparsity of the Jacobian Blocks

Because not all variables of the model have leads and lags, several columns of the Jacobian blocks will be empty and this information will generally be known before starting the computation.⁶ This information can then be exploited in the following way.

For one period, the argument of the function $f_t(\cdot)$ needs only the subvector of z_t containing lagged and leading variables actually used in the model. If the function $f_t(\cdot)$ is redefined in this manner, its corresponding Jacobian will necessarily just contain the nonempty columns of L_t , C_t and F_t . Then, the variables present in block M_t , are those present with a lead in the model. Furthermore, in most large macroeconomic models, the C_t matrix as well as $(C_t - L_t M_{t-1})$ can be very sparse, and there is great numerical advantage in using sparse matrix techniques—for a discussion of available sparse matrix techniques see Press et al (1992).

III. AN ILLUSTRATIVE EXAMPLE OF THE L-B-J METHOD

In order to illustrate the L-B-J method in practice, consider the following simple nonlinear model of the output-inflation process suggested by Laxton, Meredith and Rose (1995):

⁶It is possible that because of linearization that a zero element may suddenly become nonzero. This problem is easily dealt with by algorithms that are designed to exploit sparse systems.

$$\begin{aligned}
 PDOT_t &= .414*PDOT_{t+1} + (1-.414)*PDOT_{t-1} + .196*(g^2/(g-Y_t) - g) \\
 &\quad + .276*(g^2/(g-Y_{t-1}) - g) \\
 RR_t &= RS_t - .414*PDOT_{t+1} - (1-.414)*PDOT_{t-1} \\
 RS_t &= 3*PDOT_t - Y_t \\
 Y_t &= .304*Y_{t-1} - .098*RR_t - .315*RR_{t-1} + EY_t
 \end{aligned}$$

where PDOT is inflation, Y is the output gap (real GDP relative to potential output), RR is the real interest rate, RS is the short-term interest rate, EY is a shock term and g is a constant equal to 0.049. The model also contains an equation for the long-term interest rate that is determined by the expectation theory of the term structure; this variable is determined recursively and is excluded from the matrices to simplify the exposition.

This particular functional form has some interesting properties because it implies that the short-run trade-off between output and inflation—or the so-called Phillips curve—depends on the existing level of excess demand pressures. Indeed, at very high levels of excess demand pressure (for $Y > 0.03$ or 0.04 for example) an expansionary monetary policy would have enormous effects on inflation and only small positive effects on Y. This is an interesting example because the model contains forward-looking behavior as well as a significant nonlinearity that would arise in any well-specified structural model of the monetary transmission mechanism. We examine now the consequences of a shock of .02 on EY in period 1.

In order to make the procedure more explicit, we detail one part of the iterative procedure. During the first iteration of the Newton-Raphson procedure, after the triangularization for the third period, the relevant subsystem for periods 3 and 4 is:

$$\begin{bmatrix}
 1 & 0 & 0 & 0 & -0.48 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & -0.72 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & -1.3 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0.19 & 0 & 0 & 0 \\
 -0.59 & 0 & 0 & -0.28 & 1.0 & 0 & 0 & -0.20 & -0.41 & 0 & 0 & 0 \\
 0.59 & 0 & 0 & 0 & 0 & 1.0 & -1.0 & 0 & 0.41 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & -3.0 & 0.0 & 1.0 & -1.0 & 0 & 0 & 0 & 0 \\
 0 & 0.32 & 0 & -0.30 & 0 & 0.1 & 0 & 1.0 & 0 & 0 & 0 & 0
 \end{bmatrix}
 \begin{bmatrix}
 PDOT_3 \\
 RR_3 \\
 RS_3 \\
 Y_3 \\
 PDOT_4 \\
 RR_4 \\
 RS_4 \\
 Y_4 \\
 PDOT_5 \\
 RR_5 \\
 RS_5 \\
 Y_5
 \end{bmatrix}
 =
 \begin{bmatrix}
 0.007 \\
 0.011 \\
 0.015 \\
 -0.006 \\
 0 \\
 0 \\
 0 \\
 0
 \end{bmatrix}$$

After triangularization for period 4, the subsystem becomes:

$$\begin{bmatrix}
 1 & 0 & 0 & 0 & -0.48 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & -0.72 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & -1.3 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0.19 & 0 & 0 & 0 \\
 & & & & 1 & 0 & 0 & 0 & -0.48 & 0 & 0 & 0 \\
 & & & & 0 & 1 & 0 & 0 & -0.70 & 0 & 0 & 0 \\
 & & & & 0 & 0 & 1 & 0 & -1.2 & 0 & 0 & 0 \\
 & & & & 0 & 0 & 0 & 1 & 0.21 & 0 & 0 & 0
 \end{bmatrix}
 \begin{bmatrix}
 PDOT_3 \\
 RR_3 \\
 RS_3 \\
 Y_3 \\
 PDOT_4 \\
 RR_4 \\
 RS_4 \\
 Y_4 \\
 PDOT_5 \\
 RR_5 \\
 RS_5 \\
 Y_5
 \end{bmatrix}
 =
 \begin{bmatrix}
 0.007 \\
 0.011 \\
 0.015 \\
 -0.006 \\
 0.002 \\
 -0.005 \\
 0 \\
 -0.006
 \end{bmatrix}$$

In this example, only one column of the M blocks need to be stored. It is the one corresponding to the lead variable $PDOT_{t+1}$.

Charts 1 to 4 show how the trajectory of the endogenous variables change from iteration to iteration. The values reported in these charts are for inflation, the short-term interest rate, real GDP and the long-term interest rate. In total, it takes four iterations to reach convergence.⁷ However, only the first two iterations are distinguishable from one another in the charts and for most practical purposes we can say that system has converged after two iterations.

In the first iteration the model is linearized around $Y = 0$ so the procedure initially underestimates the inflationary effect of the Y shock. As a result the positive response of short-term and long-term interest rates is also underestimated in the first period of the shock. However, in the second iteration the model now has a better solution trajectory to linearize the model around and consequently the procedure will now produce much more reliable estimates of the inflationary effects of the EY shock. Of course, when the system is stacked the same properties will hold for all periods and any updating of the linearization process in the neighborhood of the true solution will converge rapidly. For example, this nonlinear model of the output-inflation process suggests that the output gap, or real GDP, will have to undershoot its equilibrium level—as can be seen in Chart 3, the negative effects on real GDP are also estimated fairly precisely in the second iteration. Again, this a key property of stacking the system and exploiting the quadratic convergence and robustness properties of a Newton-Raphson based algorithm.

⁷In this example, the convergency criterium is set so that $\max \|f(z_t)\| \leq 10^{-5}$.

Chart 1. Small Model Results for Inflation

PERIOD	BASE	I	II	III	IV
1	0	0	0	0	0
2	0	0.6496	0.9641	0.9688	0.9688
3	0	0.7754	1.198	1.208	1.208
4	0	0.1448	0.322	0.3358	0.3358
5	0	-0.05994	-0.02502	-0.01632	-0.0163
6	0	-0.00611	-0.004	-0.00259	-0.00258
7	0	0.01677	0.02084	0.02039	0.0204
8	0	0.003683	0.006576	0.006624	0.006624
9	0	-0.00218	-0.00181	-0.00164	-0.00164
10	0	-0.00054	-0.00073	-0.00071	-0.00071
11	0	0.000417	0.000437	0.000417	0.000417
12	0	0.000126	0.000189	0.000186	0.000186
13	0	-6.30E-05	-5.60E-05	-5.20E-05	-5.20E-05
14	0	-2.40E-05	-3.20E-05	-3.10E-05	-3.10E-05
15	0	0.000008	0.000008	0.000007	0.000007

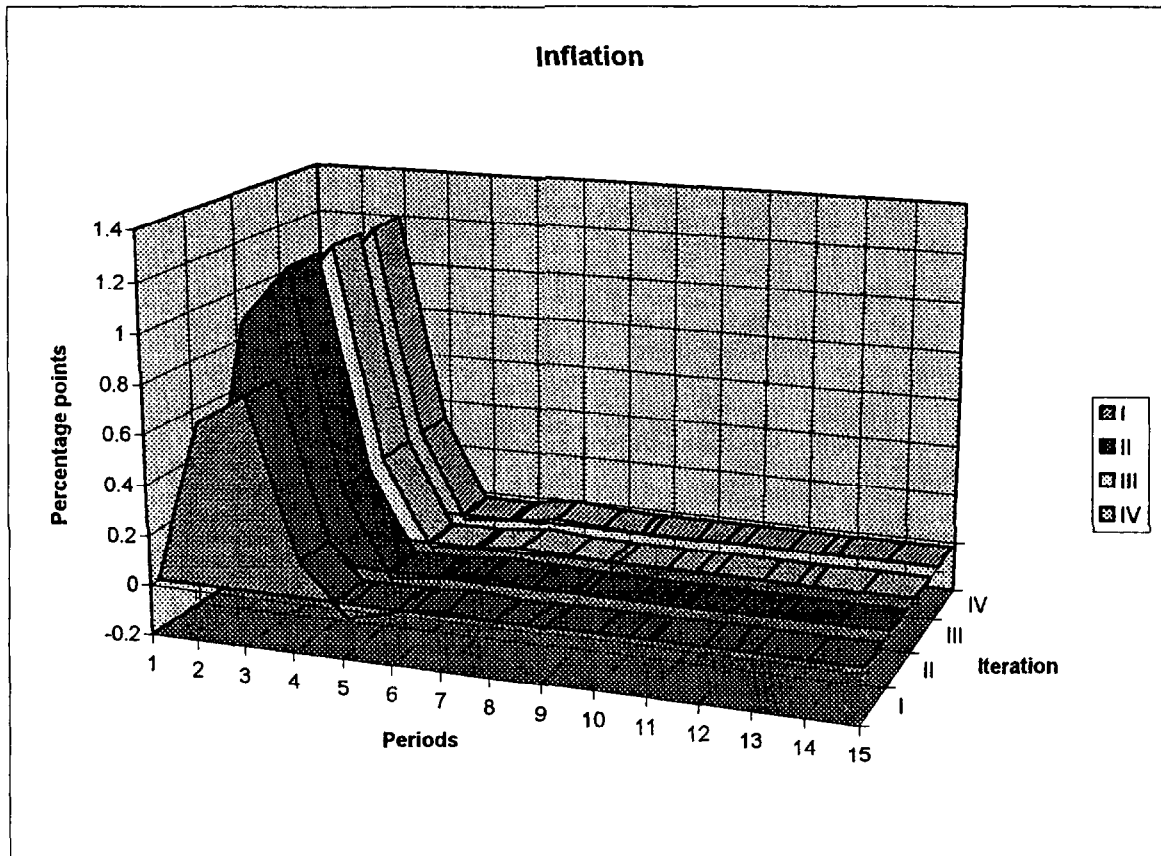


Chart 2. Small Model Results for Short-Term Interest Rate

PERIOD	BASE	I	II	III	IV
1	0	0	0	0	0
2	0	3.625	4.5	4.513	4.513
3	0	1.674	2.631	2.658	2.658
4	0	-0.1005	0.1208	0.1525	0.1525
5	0	-0.1524	-0.122	-0.1063	-0.1063
6	0	0.05571	0.06419	0.06358	0.0636
7	0	0.04206	0.05792	0.05702	0.05702
8	0	-0.00408	0.00121	0.001947	0.001946
9	0	-0.00625	-0.00682	-0.00645	-0.00645
10	0	0.000861	0.000514	0.000459	0.00046
11	0	0.001238	0.001513	0.001463	0.001463
12	0	-0.00007	0.000058	0.000071	0.000071
13	0	-0.00021	-0.00024	-0.00022	-0.00022
14	0	0.000005	-1.30E-05	-1.40E-05	-1.40E-05
15	0	0.000032	0.000037	0.000036	0.000036

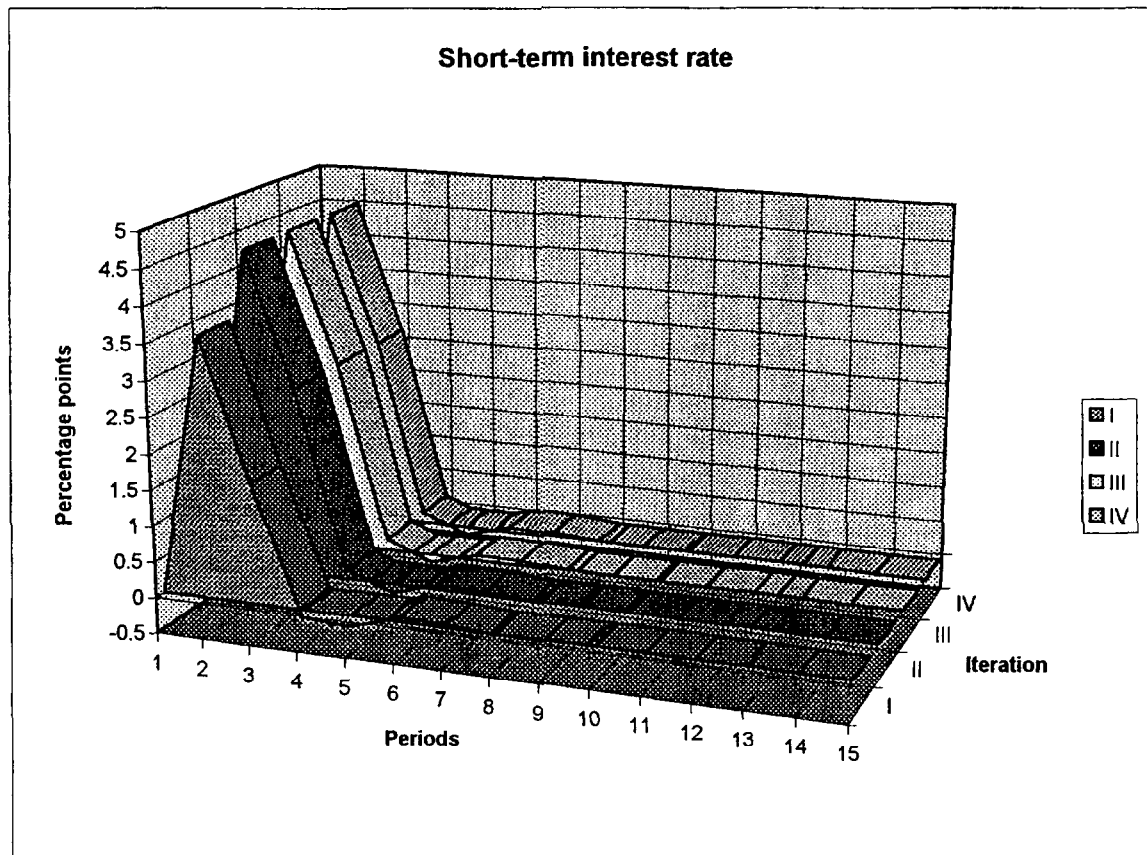


Chart 3. Small Model Results for Real GDP

PERIOD	BASE	I	II	III	IV
1	0	0	0	0	0
2	0	1.676	1.608	1.607	1.607
3	0	-0.652	-0.962	-0.9668	-0.9668
4	0	-0.5349	-0.8453	-0.8548	-0.8548
5	0	0.02739	-0.04691	-0.05739	-0.0574
6	0	0.07405	0.0762	0.07135	0.07134
7	0	-0.00824	-0.0046	-0.00416	-0.00417
8	0	-0.01513	-0.01852	-0.01793	-0.01793
9	0	0.000299	-0.00138	-0.00153	-0.00153
10	0	0.002475	0.00271	0.002574	0.002574
11	0	-1.40E-05	0.000202	0.000213	0.000213
12	0	-0.00045	-0.00051	-0.00049	-0.00049
13	0	-1.90E-05	-6.70E-05	-6.90E-05	-6.90E-05
14	0	0.000077	0.000083	0.000079	0.000079
15	0	0.000007	0.000014	0.000015	0.000015

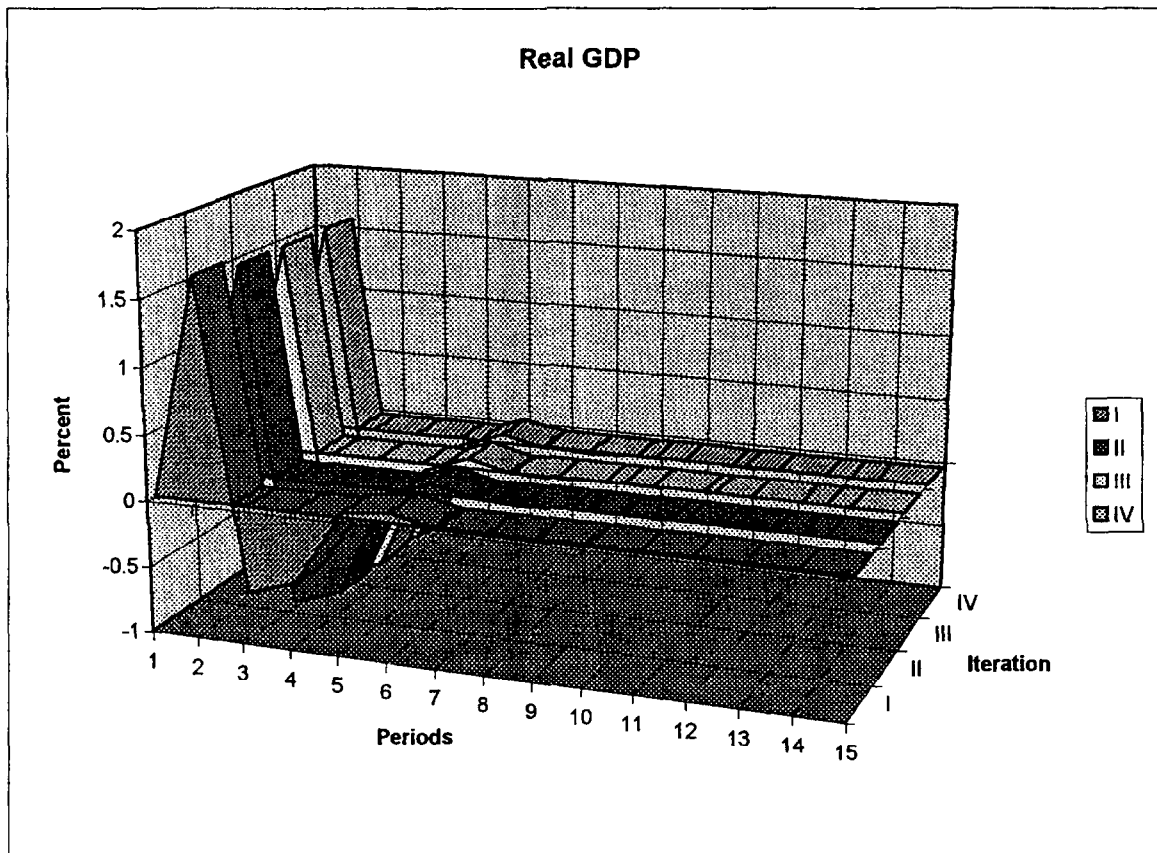
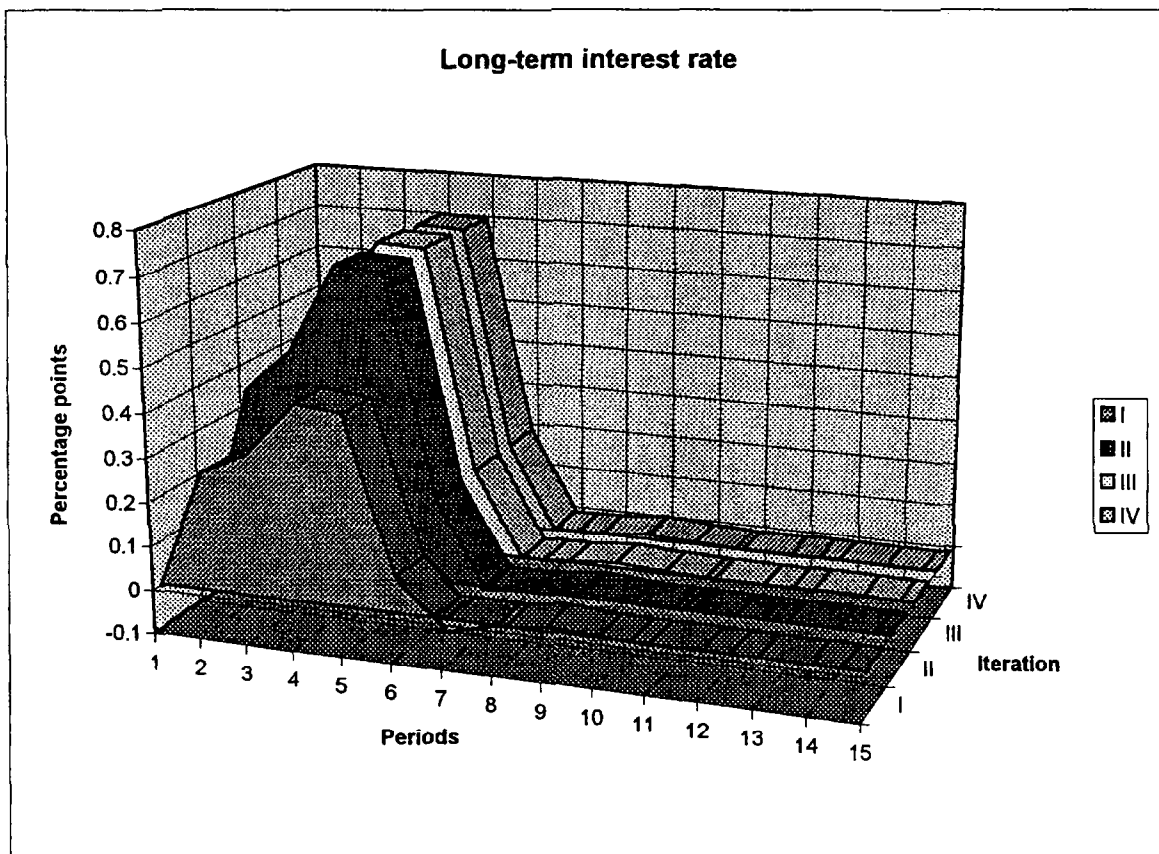


Chart 4. Small Model Results for Long-Term Interest Rate

PERIOD	BASE	I	II	III	IV
1	0	0	0	0	0
2	0	0.2689	0.3991	0.4011	0.4011
3	0	0.321	0.4958	0.5002	0.5002
4	0	0.4406	0.6983	0.7067	0.7067
5	0	0.4296	0.6914	0.7013	0.7013
6	0	0.08231	0.187	0.1957	0.1957
7	0	-0.02819	-0.00604	-0.00112	-0.00111
8	0	-0.00206	0.000377	0.001224	0.001231
9	0	0.008922	0.01146	0.01127	0.01127
10	0	0.001935	0.003551	0.00359	0.00359
11	0	-0.00111	-0.00088	-0.00079	-0.00079
12	0	-0.00026	-0.00035	-0.00034	-0.00034
13	0	0.000218	0.000233	0.000223	0.000223
14	0	0.000064	0.000098	0.000096	0.000096
15	0	-3.40E-05	-0.00003	-2.70E-05	-2.70E-05



IV. THE FAIR-TAYLOR METHOD

The Fair-Taylor (1983) method essentially breaks the full simultaneous problem discussed earlier into two parts and then relies on a Gauss-Seidel iterative scheme to obtain convergence. This process involves specifying guesses for the leads of the model—in the example discussed in Section III this would involve specifying guesses for lead values of PDOT in the inflation equation and the monetary policy reaction function—and then solving the model period by period over some horizon. This process is repeated until the expectational variables are consistent with the actual solution of the model up to some prespecified convergence criteria. Fair and Taylor referred to the solutions for each period where expectations are taken as predetermined as Type I iterations and the process of achieving consistency in expectations as Type II iterations. The Fair-Taylor algorithm also included Type III iterations. This third layer of iterations is necessary when the true terminal conditions are unknown. In practice, users can ignore this layer of iterations if they either know the steady state of the model or choose a simulation horizon that is sufficiently long that it will not have significant effects on the range of interest. In this paper, we ignore these issues and focus our attention on the properties of algorithms that combine Type I and Type II iterations versus the algorithms that do not.

In the early days of computer technology when memory was scarce and sparse matrix techniques were not available, it was impossible to solve large systems without breaking them into more manageable parts. Indeed, as late as 1983 Fair and Taylor were not only recommending that the full system be broken in terms of Type I and Type II iterations, they also recommended that Type I iterations be further broken down and solved with a Gauss-Seidel algorithm.⁸ The major problem with breaking the system into components is that, in some cases, these techniques may not find a solution even though a well-defined saddle point stable solution may exist.⁹ For example, Armstrong et al (1995) provide some examples of models that are saddle-point stable

⁸Much of the discussion in the literature of the relative merits of Newton-Raphson based methods versus Gauss-Seidel has ignored techniques to deal with the sparse and repetitive structure of the Jacobian—see, for example, Hughes Hallett and Fisher (1992). Obviously, for even small models, the matrix inversion problem for an Newton-Raphson based algorithm can become incredibly time consuming without some technique to exploit the sparse structure of the Jacobian. For a discussion of available sparse matrix techniques, see Press et al (1992).

⁹Hughes Hallett, Ma, and Yin (1996) provide an example of a nonlinear model of resource extraction where Newton-Raphson fails to converge. Of course, this is always a possibility in highly nonlinear models but we have never encountered any of these situations yet in our work with models that have a well-defined balanced growth path. However, users of MULTIMOD have found many situations where Fair-Taylor iterations would not converge without damping or using a different ordering of the model. Armstrong et al (1995) reports similar results for development work done on the Bank of Canada's Quarterly Projection Model.

where the Fair-Taylor algorithm is simply incapable of finding a solution¹⁰

V. L-B-J VERSUS F-T ON A MARK II VERSION OF MULTIMOD

There are a few reasons why the MARK II version of MULTIMOD, represents an interesting test case for a new algorithm.¹¹ First, because this version of the model was developed with the Fair-Taylor algorithm, the results are unlikely to be biased against it and in favor of the L-B-J method. Second, the structure of the model mimics in many respects the structure of many other modern forward-looking macro models that have been designed for policy analysis—see for example, Coletti et al (1996), Gagnon (1991), McKibbin and Sachs (1991), Helliwell et al (1990), Taylor (1988), and Edison et al (1987)—and for this reason the tests are likely to be representative.¹² Third, the model has been used extensively outside the Fund and in some cases it has been used as a benchmark to compare the performance of other algorithms—see for example, Pauletto (1995).

As mentioned earlier most of the development of MULTIMOD has taken place with Fair-Taylor. The specific implementation of the current version of the F-T algorithm is done as a macro in portable TROLL. Users have the option of solving the Type I iterations with either Newton-Raphson or Gauss-Seidel. In this paper we will only consider the former because we have been unable to perform MULTIMOD simulations reliably with Gauss-Seidel.¹³

The base-case set of results are reported in Table 1. These results compare the L-B-J method versus Fair-Taylor on a Mark II version MULTIMOD (MULTAQ) when the simulation horizon is set at 50 years and Type II convergence is set at 0.002.¹⁴ The table includes results for five shocks on the U.S. economy. These include shocks to: real government expenditures (*G*); the target level of nominal government debt (*BT*); potential output (*YCAP*); the target money supply

¹⁰By saddle-point stable, we mean that they satisfy that Blanchard-Kahn (1980) conditions.

¹¹Documentation for early versions of MULTIMOD can be found in Masson, Symansky, Haas and Dooley (1988) and Masson, Symansky and Meredith (1990).

¹²For an exhaustive review of the properties of these types of models see Bryant, Hooper and Mann (1993).

¹³The same problems have been experienced with other software—for example Pioro, McAdam and Laxton (1996) experienced these sorts of problems using SLIM.

¹⁴These simulations were run over night on an RS/6000 workstation that has been configured specifically for these applications. Hollinger (1996) reports some estimates on the same model using a Pentium with 32 megabytes of RAM.

(MT); and the risk premium term in the interest parity equation (ER). In addition, we also include the results for an increase in the target level of government debt and the money supply in all industrial countries.

There are several interesting results in this table. First, if we examine the results for the L-B-J method we can see that the model solves in about 3 to 4 L-B-J iterations. Indeed, if we turn to Charts 5 and 6, which reports the response of the Deutsche Mark for the first two shocks reported in the table, we can see that the model gets very close to the true solution after the first iteration—we choose to report an exchange rate here because in MULTIMOD exchange rates are key jumpers in the model that absorb the effects of all shocks. In terms of absolute time, the L-B-J method produces some very impressive results with all simulations completed in under two minutes.

If we now turn to the Fair-Taylor results we can see that the simulation times increase dramatically in some cases when a large number of Type II iterations are required for achieving convergence in expectations. The table also reports that maximum differences in the first period of the simulation between the L-B-J solution and the F-T solution for interest rates and the exchange rate. As can be seen in the table with this fairly loose level of Type II convergence tolerance (0.002) there can be fairly large differences between the L-B-J results and the F-T results. In fact, in the case of the shock to the U.S. interest parity relationship there can be errors in the exchange rate that can be as large as 7 percent.

In order to more closely replicate the precision in the L-B-J results, we recalibrated the Type II convergence criteria until the quantitative differences between the two methods were less than 0.1 in both absolute and percentage terms for all of the shocks. These results are reported in Table 2. As can be seen in the Table, it is possible to replicate the degree of precision in the L-B-J results but this takes fairly tight convergence criteria and many more Type II iterations.

For some shocks on MULTIMOD it can take more than 50 years to achieve a new steady-state equilibrium especially if the shock changes the desired net-foreign-asset position of the economy. Tables 3 and 4 redo the experiments considered in the first two tables with a simulation horizon increased from 50 years to 150 years. One of the great advantages of L-B-J is that solution times are approximately a linear function of the simulation horizon. Indeed, if we compare the L-B-J results in Table 3 with the L-B-J results in Table 1—or Table 4 with Table 2—we will see that the L-B-J solution times are approximately three times as large when we expand the simulation horizon by a factor of three. This is not the case for algorithms such as F-T. In some cases, solution time increases by less than a factor of three but in other cases by significantly more.

Although the exact magnitude of the efficiency gains of the L-B-J method relative to F-T depend on the shock, desired accuracy, and the simulation horizon these results strongly suggest that the L-B-J method is better suited for work that requires quick turnaround. However, as mentioned earlier, simulation speed is only one criteria for comparing alternative algorithms.

Chart 5. Multimod Results for the value of the Deutsche Mark

PERIOD	BASE	I	II	III
1	0.679684	0.674546	0.674598	0.674598
2	0.672895	0.668248	0.668297	0.668297
3	0.667468	0.663719	0.663758	0.663758
4	0.662476	0.659825	0.659853	0.659853
5	0.657789	0.65609	0.656109	0.656109
6	0.657789	0.656627	0.656645	0.656645
7	0.657789	0.656734	0.656751	0.656751
8	0.657789	0.656523	0.656537	0.656537
9	0.657789	0.656166	0.656178	0.656178
10	0.657789	0.655812	0.655821	0.655821

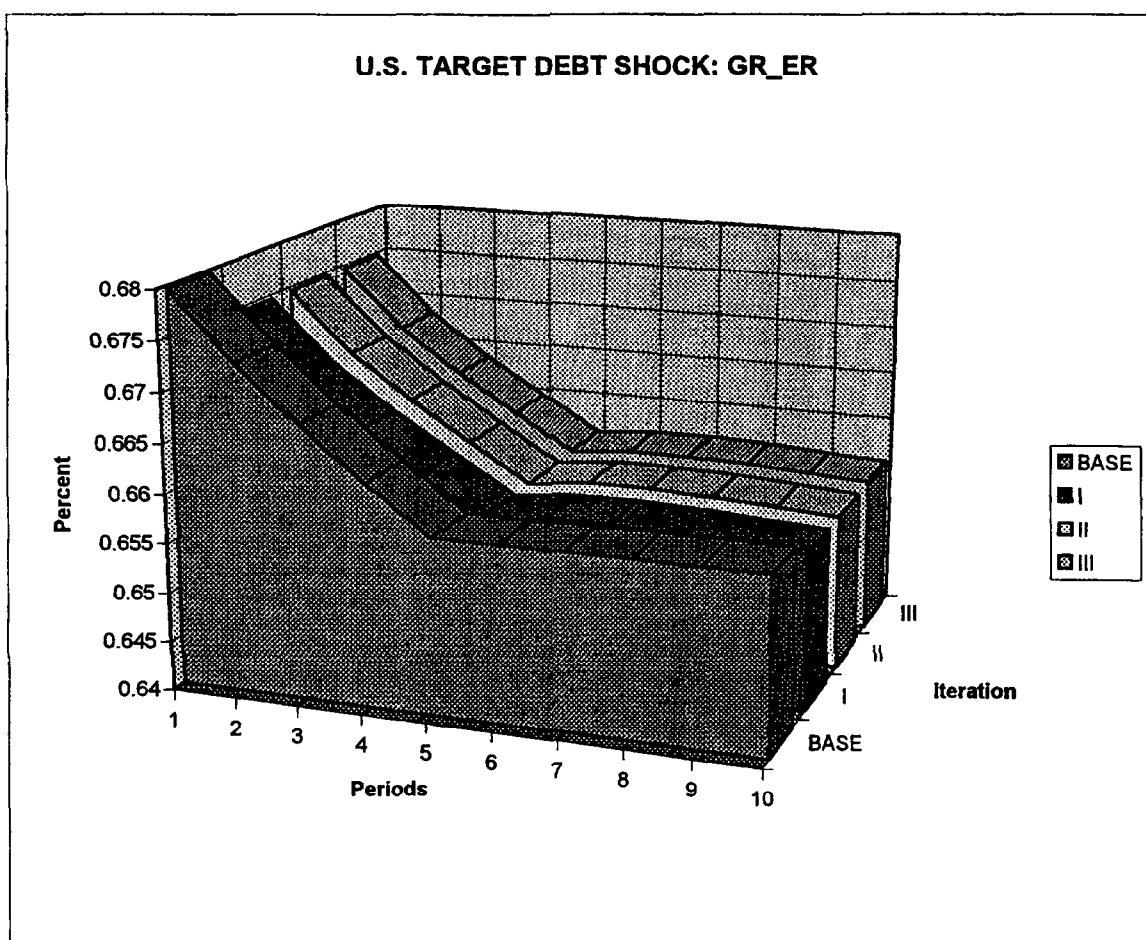
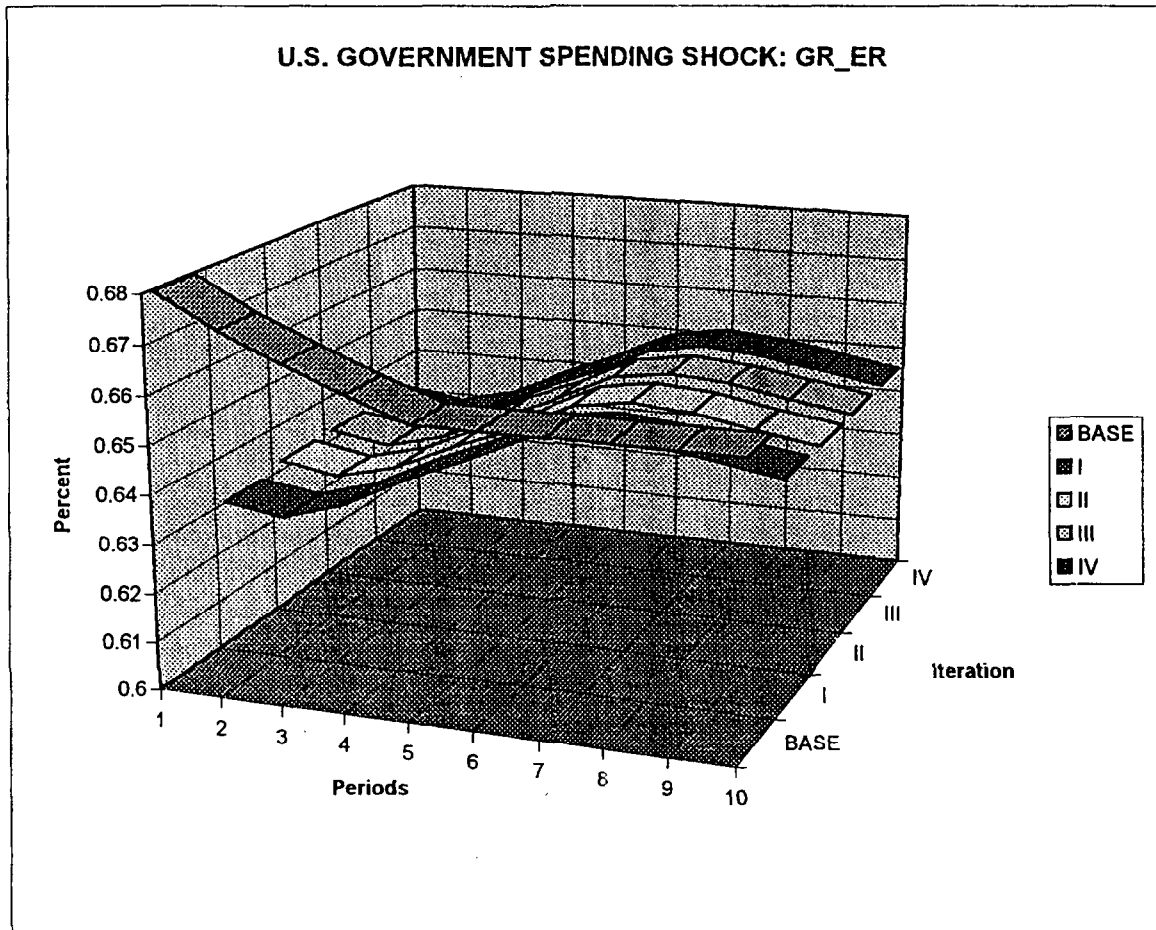


Chart 6. Multimod Results for the value of the Deutsche Mark

PERIOD	BASE	I	II	III	IV
1	0.679684	0.630221	0.631989	0.632099	0.632099
2	0.672895	0.628101	0.629728	0.629844	0.629844
3	0.667468	0.631934	0.632966	0.63308	0.63308
4	0.662476	0.638445	0.638737	0.638836	0.638836
5	0.657789	0.643867	0.643719	0.643791	0.643791
6	0.657789	0.649934	0.649734	0.649781	0.649781
7	0.657789	0.651852	0.651762	0.651796	0.651796
8	0.657789	0.650828	0.650786	0.650822	0.650822
9	0.657789	0.64863	0.64851	0.648557	0.648557
10	0.657789	0.646671	0.646406	0.646464	0.646464



VI. EXPERIENCES WITH MORE RECENT VERSIONS OF MULTIMOD

A more important criteria for active model development work is that the simulation algorithm should require minimal intervention by the user. In other words, a desirable property of any algorithm is that it should be robust in situations where there are well-defined saddle-point stable solutions. As mention earlier, there is no guarantee that algorithms such as F-T will converge even in situations where well-defined saddle point stable solutions exist. Indeed, before Newton-Raphson based methods were developed it was more difficult to tell in some situations if convergence failure was a consequence of an unstable model or if it was a consequence of an unstable algorithm.

Although even Newton-based techniques may not converge in highly nonlinear models, our experience thus far with introducing moderate nonlinearities into new versions of MULTIMOD suggest that the L-B-J technique is considerably more robust than F-T. Indeed, in several situations in the past, MULTIMOD development has been impeded because F-T had difficulty finding a solution. In some cases, these problems were resolved by finding appropriate damping factors—for an example, of a version of MULTIMOD where damping was necessary—see Bartolini, Razin, and Symansky (1995). In other cases, these problems were so severe that it would have been impractical to carry on with existing tools. This was the case for example in a recent version of the model that examined the implications of endogenous total factor productivity see Bayoumi, Coe, and Helpman (1996).

VII. CONCLUSIONS

The development and use of forward-looking macro models in policymaking institutions has proceeded at a much slower pace than what was predicted in the early 1980s. An important reason for this is that researchers have not had access to robust and efficient solution techniques for solving nonlinear forward-looking models. This paper discusses the properties of a new algorithm that is used for solving MULTIMOD. Relative to traditional algorithms in use today, we find that this algorithm is considerably faster and much less prone to simulation failure. Because the algorithm can be used to solve individual country models of the same size, the results in this paper are likely to be of interest to anyone who is also engaged in the development of national models.

REFERENCES

- Armstrong, J., R. Black, D. Laxton, and D. Rose (1995), "A Robust Method for Simulating Forward-Looking Models," in *The Bank of Canada's New Quarterly Projection Model, Part 2*, Technical Report No. 73 (Ottawa, Canada: Bank of Canada).
- Bartolini, L., A. Razin, and S. Symansky (1995), "G-7 fiscal restructuring in the 1990s: macroeconomic effects," in *Economic Policy*, Vol. 20, pp. 111-146.
- Bayoumi, Tamim, D.T. Coe, and E. Helpman (1996), "R&D Spillovers and Global Growth," IMF Working Paper No. WP/96/47 (Washington: International Monetary Fund).
- Black, R., D. Laxton, D. Rose and R. Tetlow (1994), "The Steady State Model: SSQPM," in *The Bank of Canada's New Quarterly Projection Model, Part 1*, Technical Report No. 72 (Ottawa, Canada: Bank of Canada).
- Blanchard, O. J., and C. M. Kahn (1980), "The Solution of Linear Difference Models Under Rational Expectations" in *Econometrica*, Vol. 48, pp. 1305-11.
- Boucekkine, R. (1995), "An Alternative Methodology for Solving Nonlinear Forward-Looking Models" in *Journal of Economic Dynamics and Control*, Vol. 19, No. 4, pp. 711-34.
- Bryant, R.C., P. Hooper, and C.L. Mann (1993), "*Evaluating Policy Regimes: New Research in Empirical Macroeconomics*," The Brookings Institution.
- Coletti, D., B. Hunt, D. Rose and R. Tetlow (1996), "The Dynamic Model: QPM," in *The Bank of Canada's New Quarterly Projection Model, Part 3*, Technical Report No. 75 (Ottawa, Canada: Bank of Canada).
- Edison, H.J., J.R. Marquez, and R.W. Tryon (1987), "The Structures and Properties of the Federal Reserve Board Multicountry Model," *Economic Modeling*, Vol. 4, pp. 115-315.
- Fair, R.C. and J.B. Taylor (1983), "Solution and Maximum Likelihood Estimation of Dynamic Nonlinear Rational Expectations Models," *Econometrica*, Vol. 51, No. 4, pp. 1169-185.
- Gagnon, J.E. (1991), "A Forward-Looking Multicountry Model for Policy Analysis: MX3," *Economic and Financial Computing*, Vol. 1, pp. 311-61.
- Helliwell, J.F., G. Meredith, P. Bagnoli, and Y. Durand (1990), "INTERMOD 1.1: A G-7 Version of the IMF's Multimod," *Economic Modelling*, Vol. 7, pp. 3-62.

- Hollinger, P. (1996), "The Stacked-Time Simulator in TROLL: A Robust Algorithm for Solving Forward-Looking Models," Paper presented at the Second International Conference on Computing in Economics and Finance, Geneva, Switzerland, June 26-28, 1996.
- Hughes Hallett, A. J. and P. J. Fisher (1990), "On Economic Structures and Model Solution Methods: Or Should Econometricians Use Newton Methods for Model Solution?," *Oxford Bulletin of Economics and Statistics*, Vol. 52, No 3, pp. 317-30.
- , Y. Ma, and Y. P. Yin, (1996), "Hybrid Algorithms with Automatic Switching for Solving Nonlinear Equations Systems in Economics," in *Journal of Economic Dynamics and Control*, Vol. 20, p. 1051-71.
- Juillard, M. (1996), "DYNARE: A Program for the Resolution and Simulation of Dynamic Models With Forward Variables Through the Use of a Relaxation Algorithm," CEPREMAP Working Paper No. 9602 (Paris, France: CEPREMAP).
- Laffargue, J.P. (1990), "Résolution d'un Modèle Macroéconomique Avec Anticipations Rationnelles," in *Annales d'Economie et Statistique*, Vol. 17, pp. 97-119.
- Laxton, D., G. Meredith and D. Rose (1995), "Asymmetric Effects of Economic Activity on Inflation," IMF Staff papers, Vol 42, No 2 (Washington: International Monetary Fund, June 1995), pp. 344-74.
- McKibbin, W.J., and J.D. Sachs (1991), "*Global Linkages: Macroeconomic Interdependence and Cooperation in the World Economy*," The Brookings Institution.
- Masson, P.R., S.A. Symansky, R. Haas, and M. Dooley (1988), "MULTIMOD: A Multi-Region Econometric Model," in *Staff Studies for the World Economic Outlook* (Washington: International Monetary Fund), pp. 50-104.
- , S.A. Symansky, and G. Meredith (1990), "MULTIMOD MARK II: A Revised and Extended Mode," Occasional Paper No. 71 (Washington: International Monetary Fund).
- Meredith, G. (1989), "INTERMOD 2.0: Model Specification and Simulation Properties," Working Paper 89-7, Working Group on International Macroeconomics (Ottawa, Canada: Department of Finance).
- Pioro, H., P. McAdam, and D. Laxton (1996), "Solving MULTIMOD with First-Order and Newton-Based Techniques," forthcoming University of Strathclyde Discussion Paper.
- Pauletto, G. (1995), "Solution and Simulation of Macroeconometric Models," Thesis, Department of Econometrics, University of Geneva.

Press, W., S. Teukolsky, W. Vetterling, and B. Flannery (1992), *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Second Edition 1992.

Taylor, J.B. (1988), "The Treatment of Expectations in Large Multicountry Econometric Models," In Bryant, Henderson, and others, pp. 161-82.

Table 1. L-B-J Method Versus Fair-Taylor in TROLL

Multimod Version: MULTAQ

Simulation Period: 50

N-R Convergence Criteria: 0.001

F-T Type 2 Convergence Criteria: 0.002

	L-B-J		Fair-Taylor		Maximum Difference		
	Time (Minutes)	Type 1 Iterations	Time (Minutes)	Type 2 Iterations	RS (% pts)	RL (% pts)	ER (%)
US-G shock (5%)	1.52	4	9.72	89	0.019	0.011	0.970
US-BT (5%)	1.17	3	0.82	7	0.019	0.018	0.814
US-YCAP shock (5%)	1.52	4	9.50	87	0.016	0.007	0.974
US-MT shock (10%)	1.52	4	6.37	58	0.008	0.007	0.347P
US-ER shock (0.5%)	1.50	4	1.68	15	0.110	0.146	7.258
BT shock for all countries (5%)	1.17	3	0.92	8	0.036	0.027	1.963
MT shock for all countries (10%)	1.52	4	7.67	70	0.015	0.014	0.710

Table 2. L-B-J Method Versus Fair-Taylor in TROLL

Multimod Version: MULTAQ

Simulation Period: 50

N-R Convergence Criteria: 0.001

F-T Type 2 Convergence Criteria: 0.00001

	L-B-J		Fair-Taylor		Maximum Difference		
	Time (Minutes)	Type 1 Iterations	Time (Minutes)	Type 2 Iterations	RS (% pts)	RL (% pts)	ER (%)
US-G shock (5%)	1.52	4	19.25	177	0.000	0.000	0.005
US-BT (5%)	1.17	3	12.25	113	0.000	0.000	0.001
US-YCAP shock (5%)	1.52	4	19.17	176	0.000	0.000	0.005
US-MT shock (10%)	1.53	4	15.23	140	0.000	0.000	0.006
US-ER shock (0.5%)	1.52	4	12.53	115	0.000	0.000	0.003
BT shock for all countries (5%)	1.17	3	13.72	126	0.000	0.000	0.003
MT shock for all countries (10%)	1.52	4	17.53	161	0.000	0.000	0.004

Table 3. L-B-J Method Versus Fair-Taylor in TROLL

Multimod Version: MULTAQ
Simulation Period: 150
N-R Convergence Criteria: 0.001
F-T Type 2 Convergence Criteria: 0.002

	L-B-J		Fair-Taylor		Maximum Difference		
	Time (Minutes)	Type 1 Iterations	Time (Minutes)	Type 2 Iterations	RS (% pts)	RL (% pts)	ER (%)
US-G shock (5%)	4.38	4	83.98	263	0.014	0.012	0.654
US-BT (5%)	3.33	3	2.30	7	0.044	0.037	2.081
US-YCAP shock (5%)	4.37	4	82.85	260	0.009	0.010	0.403
US-MT shock (10%)	4.38	4	17.58	55	0.039	0.040	1.627
US-ER shock (0.5%)	5.45	5	50.45	148	0.003	0.008	0.568
BT shock for all countries (5%)	5.10	4	2.97	8	0.103	0.074	5.453
MT shock for all countries (10%)	4.85	4	79.77	240	0.010	0.008	0.494

Table 4. L-B-J Method Versus Fair-Taylor in TROLL

Multimod Version: MULTAQ

Simulation Period: 150

N-R Convergence Criteria: 0.001

F-T Type 2 Convergence Criteria: 0.00001

	L-B-J		Fair-Taylor		Maximum Difference		
	Time	Type 1	Time	Type 2	RS	RL	ER
	(Minutes)	Iterations	(Minutes)	Iterations	(% pts)	(% pts)	(%)
US-G shock (5%)	4.42	4	163.72	499	0.000	0.000	0.003
US-BT (5%)	3.35	3	103.97	319	0.000	0.000	0.002
US-YCAP shock (5%)	4.40	4	156.60	486	0.000	0.000	0.004
US-MT shock (10%)	4.43	4	126.25	348	0.000	0.000	0.001
US-ER shock (0.5%)	5.45	5	103.67	324	0.000	0.000	0.005
BT shock for all countries (5%)	4.38	4	110.53	346	0.000	0.000	0.001
MT shock for all countries (10%)	4.43	4	143.37	449	0.000	0.000	0.004