Journal of Economic Dynamics and Control
22 (1998) 1291–1318

JOURNAL OF
Economic
Dynamics
& Control

An algorithm competition: First-order iterations versus Newton-based techniques

Michel Juillard[a], Douglas Laxton[b,*], Peter McAdam[c], Hope Pioro[d]

[a] *University Paris 8 – CEPREMAP, France*
[b] *International Monetary Fund, Research Department, Washington DC 20431, USA*
[c] *University of Strathclyde, Glasgow, Scotland, UK*
[d] *Bank of Canada, Canada*

Abstract

The development and use of forward-looking macromodels in policy making 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 alternative algorithms for solving MULTIMOD, the IMF's multi-country model of the world economy. Relative to traditional first-order algorithms in use today, we find that Newton-based techniques are considerably faster and much less prone to simulation failure. © 1998 Elsevier Science B.V. All rights reserved.

JEL classification: C63; C88; D84; E17

Keywords: Rational expectations; Newton algorithms; Macroeconomic models

1. 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

* Corresponding author. E-mail: dlaxton@imf.org

PII S0165-1889(98)00013-X

to rely on large-scale macroeconometric 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 macroeconometric 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 model is equivalent to a 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 – see Fair and Taylor (1983), Hall (1985), Fisher et al. (1986) and Fisher (1992).

The traditional Fair–Taylor (F–T) (1983) algorithm separated the problem into three types of iterations and a first-order 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) (hereafter called L–B–J) to exploit information about the repetitive and sparse structure of the full simultaneous problem.[1]

In order to provide an estimate of the potential efficiency gains we compare the simulation times of the 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 retuning the F–T parameters to achieve convergence and this depends on the particular shock.

The remainder of this paper is organized as follows. In Section 2, we explain the basic problem and a new Newton–Raphson based solution method suggested by Laffargue (1990), Boucekkine (1995) and Juillard (1996). Section 3 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 4 provides a brief discussion of the F–T algorithm. In Section 5 we use MULTIMOD to compare the simulation performance of the L–B–J algorithm with the F–T algorithm. In Section 6, we provide examples of extensions to models like MULTIMOD that have been difficult to implement with the F–T algorithm. Section 7 provides some conclusions.

[1] The 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.

2. The L–B–J method for solving nonlinear rational expectations models

2.1. The basic problem and notation

We consider nonlinear models with n endogenous variables, $y_{i,t}$, $i = 1, \dots, n$, m exogenous variables, $x_{j,t}$, $j = 1, \dots, m$, 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, for the vector of exogenous variables, and Θ for 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:

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

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 1–T. Obviously, this particular trajectory will correspond to specific initial conditions y_0^* and terminal conditions y_{T+1}^*.[2] Conditions for the existence of

[2] In 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 ensure that the solution is not sensitive to the assumed 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).

a single saddle-point stable trajectory are discussed in Blanchard and Kahn (1980) and Boucekkine (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:

$$f(z_t) = \begin{bmatrix} g_1(y_{t-1}, y_t, y_{t+1}, x_t, \theta) \\ g_n(y_{t-1}, y_t, y_{t+1}, x_t, \theta) \end{bmatrix}, \quad t = 1, \dots, T \tag{2}$$

where $z_t = [y'_{t-1}, y'_t, y'_{t+1}]'$. For the initial and terminal conditions, we write:

$$\begin{aligned} f_0 &= y_0 - y_0^* = 0, \\ f_{T+1} &= y_{T+1} - y_{T+1}^* = 0. \end{aligned} \tag{3}$$

This is a standard two-point boundary value problem which is best solved simultaneously for all the equations and all the periods at once.[3] 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

$$F(Y) = \begin{bmatrix} f_0(y_0) \\ f_1(z_1) \\ \vdots \\ f_T(z_T) \\ f_{T+1}(y_{T+1}) \end{bmatrix} = 0. \tag{4}$$

2.2. 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.

$$\Delta Y = -\left[\frac{\partial F}{\partial Y}\right]^{-1} F(Y). \tag{5}$$

The difficulty with this approach resides with the size of the Jacobian $\partial F(Y)/\partial Y$ which is a matrix of dimension $[n \times (T+2)] \times [n \times (T+2)]$, where

[3] Sections 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).

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 $51{,}000 \times 51{,}000$.[4] 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 $50{,}000 \times 112$ 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.[5]

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:

$$\begin{bmatrix} I & & & & & & & \\ L_1 & C_1 & F_1 & & & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & L_t & C_t & F_t & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & L_T & C_T & F_T & \\ & & & & & & I & \end{bmatrix} \Delta Y = - \begin{bmatrix} 0 \\ f_1(z_1) \\ \vdots \\ f_t(z_t) \\ \vdots \\ f_T(z_T) \\ 0 \end{bmatrix} \tag{6}$$

where L_t, C_t and F_t are the partial Jacobians:

$$L_t = \frac{\partial f_t(z_t)}{\partial y_{t-1}}, \quad C_t = \frac{\partial f_t(z_t)}{\partial y_t}, \quad F_t = \frac{\partial f_t(z_t)}{\partial y_{t+1}}. \tag{7}$$

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.

[4] As shown is in Section 4, 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.

[5] Gilli and Pauletto (this issue of the JEDC) solve exactly the same problem with a nonstationary iterative method. Their method also does not suffer from the convergence problems that can plague first-order methods. It would be interesting to examine how their method compares with the L–B–J method.

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:

$$L_1\,\Delta y_0 + C_1\,\Delta y_1 + F_1\,\Delta y_2 = -f_1(z_1). \tag{8}$$

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

$$\Delta y_1 + C_1^{-1}F_1\,\Delta y_2 = -C_1^{-1}f_1(z_1). \tag{9}$$

After this first step, the system looks like:

$$\begin{bmatrix} I & & & & & \\ & I & M_1 & & & \\ & L_2 & C_2 & F_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & L_T & C_T & L_T \\ & & & & I & \end{bmatrix} \Delta Y = \begin{bmatrix} 0 \\ d_1 \\ -f_2(z_2) \\ \vdots \\ -f_T(z_T) \\ 0 \end{bmatrix} \tag{10}$$

where

$$M_1 = C_1^{-1}F_1,$$

$$d_1 = -C_1^{-1}f_1(z_1). \tag{11}$$

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

$$L_t\,\Delta y_{t-1} + C_t\,\Delta y_t + F_t\,\Delta y_{t+1} = -f_t(z_t). \tag{12}$$

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

$$\Delta y_{t-1} + M_{t-1}\,\Delta y_t = d_{t-1}. \tag{13}$$

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

$$(C_t - L_tM_{t-1})\,\Delta y_t + F_t\,\Delta y_{t+1} = -f_t(z_t) - L_td_{t-1}. \tag{14}$$

Then, solving for Δy_t,

$$\Delta y_t + (C_t - L_tM_{t-1})^{-1}F_t\,\Delta y_{t+1} = -(C_t - L_tM_{t-1})^{-1}(f_t(z_t) + L_td_{t-1}). \tag{15}$$

After triangularization, the system looks like

$$\begin{bmatrix} I & & & & & \\ & I & M_1 & & & \\ & & \ddots & \ddots & & \\ & & & I & M_T & \\ & & & & I & \end{bmatrix} \Delta Y = \begin{bmatrix} 0 \\ d_1 \\ \vdots \\ d_T \\ 0 \end{bmatrix} \tag{16}$$

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 is then easily obtained through backward substitution:

$$\Delta y_t = d_t - M_t \, \Delta y_{t+1}. \tag{17}$$

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

2.3. 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. This information can then be exploited in the following way.

For one period, the argument of the function $f_t(\,)$ needs only the subvector of z_t containing lagged and leading variables actually used in the model. If the function $f_t(\,)$ 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_i, are those present with a lead in the model. Furthermore, in most large macroeconometric 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).

3. 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 et al. (1995):

$$\begin{aligned} PDOT_t = {} & 0.414 PDOT_{t+1} + (1 - 0.414) PDOT_{t-1} \\ & + 0.196(g^2/(g - Y_t) - g) + 0.276(g^2/(g - Y_{t-1}) - g \\ RR_t = {} & RS_t - 0.414 PDOT_{t+1} - (1 - 0.414) PDOT_{t-1} \end{aligned}$$

$$RS_t = 3PDOT_t + Y_t$$

$$Y_t = 0.304Y_{T-1} - 0.098RR_t - 0.315RR_{t-1} + EY_t \tag{18}$$

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 arises in several empirical models of the monetary transmission mechanism. We examine now the consequences of a shock of 0.02 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

$$\begin{bmatrix} 1 & 0 & 0 & 0 & -0.48 & 0 & 0 & 0 & & & & \\ 0 & 1 & 0 & 0 & -0.72 & 0 & 0 & 0 & & & & \\ 0 & 0 & 1 & 0 & -1.3 & 0 & 0 & 0 & & & & \\ 0 & 0 & 0 & 1 & 0.19 & 0 & 0 & 0 & & & & \\ -0.59 & 0 & 0 & -0.28 & 1.0 & 0 & 0 & -0.20 & -0.41 & 0 & 0 & 0 \\ 0.59 & 0 & 0 & 0 & 0 & 1.0 & -1.0 & 0 & 0.41 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -3.0 & 0.0 & 1.0 & -1.0 & 0 & 0 & 0 & 0 \\ 0 & 0.32 & 0 & -0.30 & 0 & 0.1 & 0 & 1.0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\times \begin{bmatrix} \mathrm{PDOT}_3 \\ RR_3 \\ RS_3 \\ Y_3 \\ PDOT_4 \\ RR_4 \\ RS_4 \\ Y_4 \\ PDOT_5 \\ RR_5 \\ RS_5 \\ Y_5 \end{bmatrix} = \begin{bmatrix} 0.007 \\ 0.011 \\ 0.015 \\ -0.006 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{19}$$

After triangularization for period 4, the subsystem becomes

$$\begin{bmatrix} 1 & 0 & 0 & 0 & -0.48 & 0 & 0 & 0 & & & & \\ 0 & 1 & 0 & 0 & -0.72 & 0 & 0 & 0 & & & & \\ 0 & 0 & 1 & 0 & -1.3 & 0 & 0 & 0 & & & & \\ 0 & 0 & 0 & 1 & 0.19 & 0 & 0 & 0 & & & & \\ & & & & 1 & 0 & 0 & 0 & -0.48 & 0 & 0 & 0 \\ & & & & 0 & 1 & 0 & 0 & -0.70 & 0 & 0 & 0 \\ & & & & 0 & 0 & 1 & 0 & -1.2 & 0 & 0 & 0 \\ & & & & 0 & 0 & 0 & 1 & 0.21 & 0 & 0 & 0 \end{bmatrix}$$

$$\times \begin{bmatrix} \mathrm{PDOT}_3 \\ RR_3 \\ RS_3 \\ Y_3 \\ PDOT_4 \\ RR_4 \\ RS_4 \\ Y_4 \\ PDOT_5 \\ RR_5 \\ RS_5 \\ Y_5 \end{bmatrix} = \begin{bmatrix} 0.007 \\ 0.011 \\ 0.015 \\ -0.006 \\ 0.002 \\ -0.005 \\ 0 \\ -0.006 \end{bmatrix}. \tag{20}$$

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

Tables 1–4 show how the trajectory of the endogenous variables change from iteration to iteration. The values reported in these tables are for inflation, the short-term interest rate, the output gap and the long-term interest rate. In total, it takes four iterations to reach convergence.[6] 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

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

Table 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

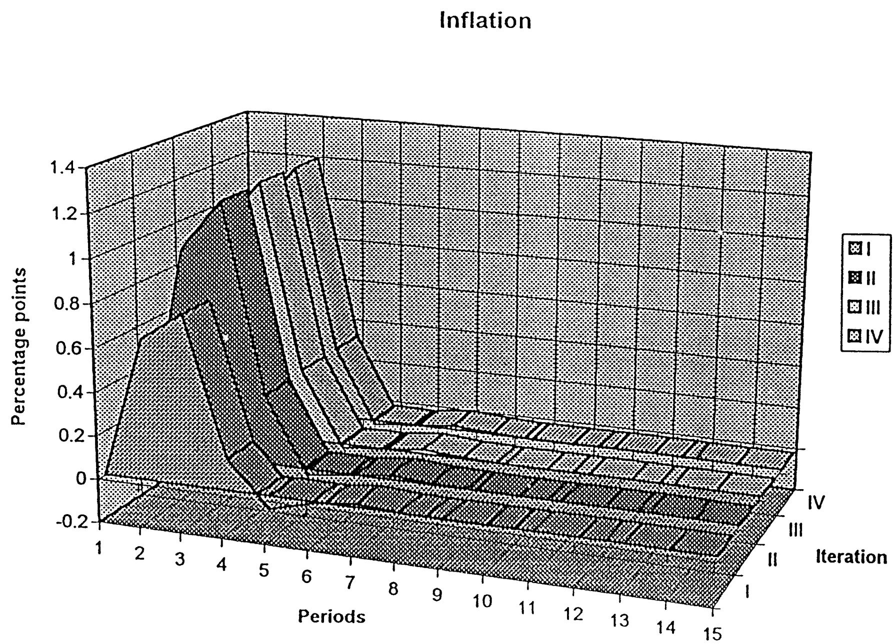
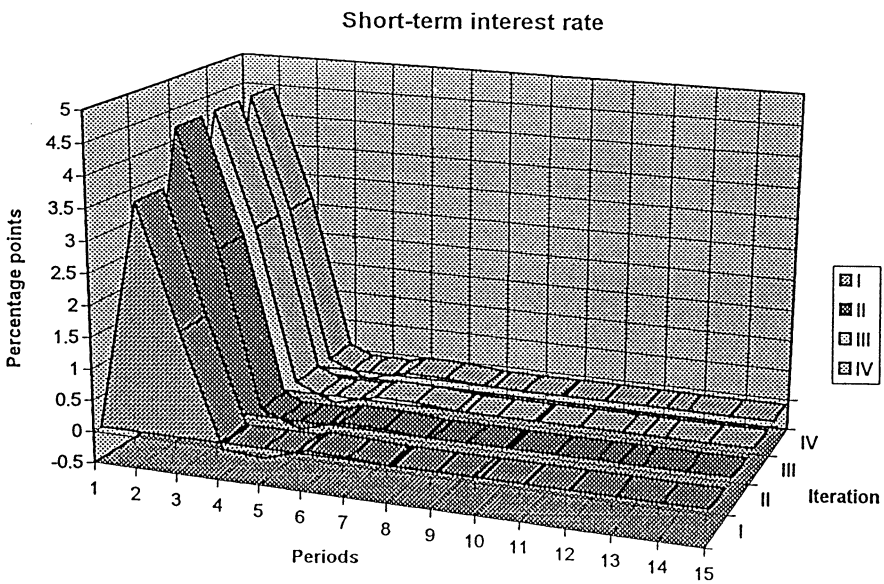


Table 2
Small model results for short-term interest rate

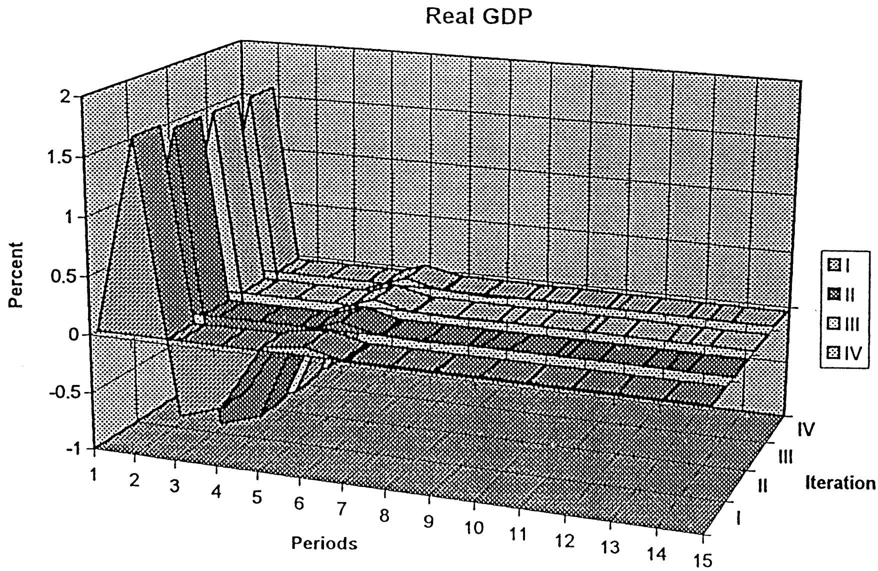
Period	Base	I	II	III	IV
1	0	0	0	0	0
2	0	3.625	4.5	4.513	4.513
3	0	1.674	2.631	2.658	2.658
4	0	– 0.1005	0.1208	0.1525	0.1525
5	0	– 0.1524	– 0.122	– 0.1063	– 0.1063
6	0	0.05571	0.06419	0.06358	0.0636
7	0	0.04206	0.05792	0.05702	0.05702
8	0	– 0.00408	0.00121	0.001947	0.001946
9	0	– 0.00625	– 0.00682	– 0.00645	– 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



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

Table 3
Small model results for real GDP

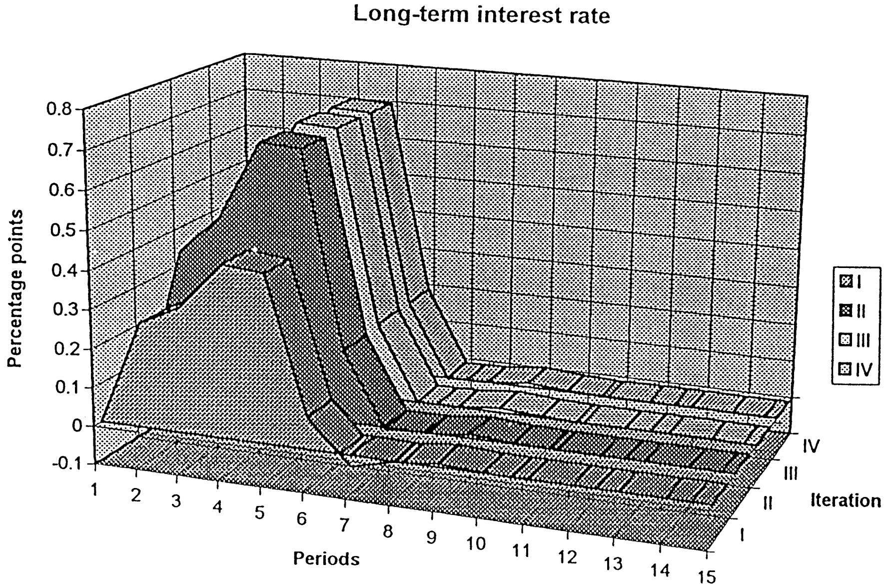
Period	Base	I	II	III	IV
1	0	0	0	0	0
2	0	1.676	1.608	1.607	1.607
3	0	− 0.652	− 0.962	− 0.9668	− 0.9668
4	0	− 0.5349	− 0.8453	− 0.8548	− 0.8548
5	0	0.02739	− 0.04691	− 0.05739	− 0.0574
6	0	0.07405	0.0762	0.07135	0.07134
7	0	− 0.00824	− 0.0046	− 0.00416	− 0.00417
8	0	− 0.01513	− 0.01852	− 0.01793	− 0.01793
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



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 Table 3, the negative effects on real *GDP* are also

Table 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



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.

4. 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 first-order iterative scheme to obtain convergence. This process involves specifying guesses for the leads of the model – in the example discussed in Section 3 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 first-order algorithm.[7] 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.[8] For example, Armstrong et al. (1995) provide some

[7] Much of the discussion in the literature of the relative merits of Newton–Raphson based methods versus first order has ignored techniques to deal with the sparse and repetitive structure of the Jacobian – see, for example, Hughes Hallett and Fisher (1990). 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 Reid (1977) and Press et al. (1992).

[8] Hughes Hallett et al. (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 not encountered many of these cases yet in our work with MULTIMOD. 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.

examples of models that are saddle-point stable where the Fair–Taylor algorithm is simply incapable of finding a solution.[9]

5. An algorithm competition on the Mark II Version of MULTIMOD

5.1. MULTIMOD Mark II

There are a few reasons why the Mark II version of MULTIMOD represents an interesting test case for a new algorithm.[10] First, because this version of the model was developed with the Fair–Taylor algorithm, the comparison is unlikely to be biased against it and in favor of the new L–B–J method. Second, the structure of the model mimics in many respects the structure of many other modern forward-looking macromodels 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.[11] Third, this version of 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).

The development of MULTIMOD has taken place principally with the TROLL modeling software. The current version of the F–T algorithm in portable TROLL is implemented as a macro.[12] TROLL users have the option of solving Type I iterations with either the Newton–Raphson or first-order algorithms. However, experience inside the Fund has suggested that it is difficult to obtain reliable solutions for MULTIMOD when Gauss–Seidel is used to solve for Type I iterations. Consequently, MULTIMOD development over the years has relied principally on a version of F–T that uses Newton–Raphson code to solve for Type I iterations.[13] Consequently, this version of the F–T algorithm is really a hybrid algorithm because a Newton-based technique is used to solve for Type I iterations and a first-order technique is used to solve for Type II iterations. However, users of MULTIMOD at, for example, the University of

[9] By saddle-point stable, we mean that they satisfy that Blanchard–Kahn (1980) stability conditions.

[10] Documentation for the Mark II version of MULTIMOD can be found in Masson et al. (1990).

[11] For an exhaustive review of the properties of these types of models see Bryant et al. (1993).

[12] The F–T macro was developed by Jon Faust, Ralph Tryon and Joe Gagnon at the Federal Reserve Board of Governors.

[13] The implementation of the Newton–Raphson algorithm in TROLL relies on the sparse matrix code developed in the 1970s at Harwell Labs – see Reid (1977).

Strathclyde and at various other mainly UK institutions have reported some success in solving MULTIMOD where first-order techniques are used to solve for both Type I and Type II iterations.[14] This section focusses mainly on a comparison of L–B–J and F–T methods in TROLL although we do report some of the convergence problems that we encountered when we relied upon the first-order methods available in SLIM.

5.2. L–B–J versus F–T in TROLL

The base-case set of results are reported in Table 5. 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 yr and the tolerance level for Type II convergence is set at 0.002.[15] The table includes results for eight shocks to the US economy. These include shocks to: real government expenditures (G); the target level of nominal government debt (BT); potential output ($YCAP$); the target money supply (MT); population; and the risk premium term in the interest parity equation (ER). Because several countries are assumed to fix their exchange rate with Germany in this version of the model the propagation mechanism of the model can vary considerably depending on where the shock originates from. For this reason, we also include two shocks to the target money supply in Germany and two fiscal shocks in France.[16] The final set of shocks that we consider are more global shocks: an increase in the target level of government debt and the money supply in all industrial countries as well as an increase in the world oil price. In four cases we double the size of the shock to provide some indication of the degree of nonlinearity in the model. While our list of shocks is certainly not exhaustive it probably provides a reasonable characterization of the performance of alternative algorithms.

There are several interesting results reported in Table 5. First, if we examine the results for the L–B–J method we can see that the model solves in about 3 to 6 L–B–J iterations. Indeed, in Tables 6 and 7 which reports the response of the Deutsche Mark (DM/US$) for the first two shocks reported in the table, we can

[14] The software that they used is SLIM. It was supplied by Fisher (1990) and others and its methodology may be roughly defined as an accelerated Fair–Taylor algorithm using incomplete inner iterations (explained in Fisher (1992).

[15] These simulations were run over night on an RS/6000 workstation that has been configured specifically for this application. In particular, the L–B–J method will run much more efficiently if the matrices can be stored in real memory. Hollinger (1996) reports some comparisons of these two algorithms using a Pentium with 32 megabytes of RAM.

[16] In the past, MULTIMOD users employing first-order methods indicated having more convergence problems with monetary and fiscal shocks to Germany.

Table 5
L–B–J method versus Fair–Taylor in TROLL

	L–B–J		Fair–Taylor		Maximum Difference		
	Time (min)	Type 1 Iterations	Time (min)	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-*G* shock (10%)	1.85	5	11.42	104	0.014	0.017	0.535
US-*BT* shock (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	0.347
US-*ER* shock (0.5%)	1.50	4	1.68	15	0.110	0.146	7.258
US population shock (1%)	1.17	3	2.22	20	0.013	0.029	0.128
US population shock (2%)	1.17	3	2.88	26	0.032	0.064	0.421
FR-*G* shock (5%)	1.52	4	2.47	22	0.012	0.015	0.603
FR-*BT* shock (5%)	1.17	3	0.28	2	0.001	0.004	0.148
GR-*MT* shock (10%)	1.52	4	18.28	167	0.000	0.006	0.088
GR-*MT* shock (20%)	1.52	4	19.17	175	0.000	0.011	0.176
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
Quadruple world oil price	2.22	6	2.47	22	0.008	0.009	0.389
Double world oil price	1.52	4	1.70	15	0.001	0.004	0.029
Total time	23.53		97.55				

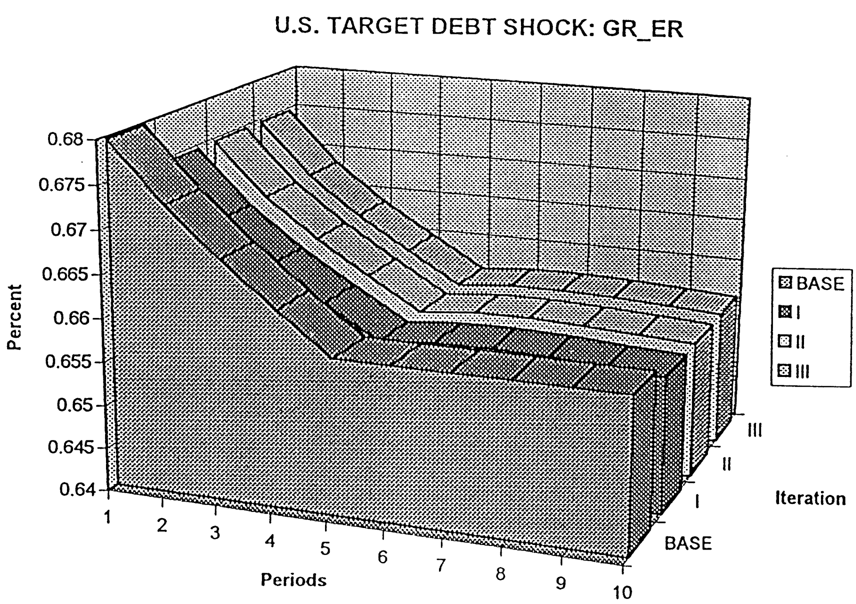
Multimod version: MULTAQ, simulation period: 50, N–R convergence criteria: 0.001, F–T type 2 convergence criteria: 0.002.

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 relative prices that reflect the effects of all shocks.[17]

[17] The exchange rate is determined by an interest parity equation. For countries with flexible exchange rates, it is the key jump variable that will respond to all shocks and move the economy gradually back to a full stock-flow equilibrium where aggregate demand is equal to aggregate supply and net foreign assets (or liabilities) are a constant share of nominal GDP.

Table 6
Multimod results for the value of the Deutsche Mark

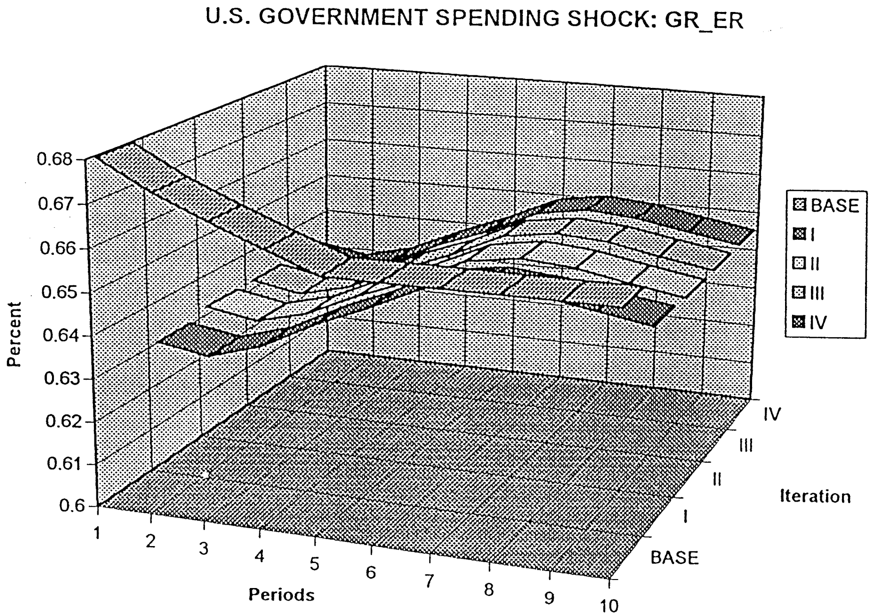
Period	Base	I	II	III
1	0.679684	0.674546	0.674598	0.674598
2	0.672895	0.668248	0.668297	0.668297
3	0.667468	0.663719	0.663758	0.663758
4	0.662476	0.659825	0.659853	0.659853
5	0.657789	0.65609	0.656109	0.656109
6	0.657789	0.656627	0.656645	0.656645
7	0.657789	0.656734	0.656751	0.656751
8	0.657789	0.656523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



In terms of absolute time, the L–B–J method produces some very impressive results with the most difficult simulation – the quadrupling of oil prices – completed in 2.22 min. Turning to the Fair–Taylor results, we can see that in a few cases that F–T solves the model incredibly quickly. For example, in the case of shocks to government debt in France or the United States, the solution times are extremely fast. However, as can be seen in the table the simulation times increase

Table 7
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	0.643719	0.643719
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



dramatically in some cases when a large number of Type II iterations are required for achieving convergence in expectations. Table 5 also reports the maximum differences between the L–B–J and the F–T methods of the first-period solutions for interest rates and the exchange rate. With a 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 US interest parity relationship there can be errors in the exchange rate that can be as large as 7%.

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 considered. These results are reported in Table 8. These results show that it is possible to replicate the L–B–J results but it takes

Table 8
L–B–J method versus Fair–Taylor in TROLL

	L–B–J		Fair–Taylor		Maximum Difference		
	Time (min)	Type 1 Iterations	Time (min)	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-*G* shock (10%)	1.87	5	22.45	205	0.000	0.000	0.003
US-*BT* shock (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
US population shock (1%)	1.17	3	15.07	139	0.000	0.000	0.006
US population shock (2%)	1.17	3	17.12	157	0.000	0.000	0.006
FR-*G* shock (5%)	1.52	4	16.78	154	0.000	0.000	0.004
FR-*BT* shock (5%)	1.15	3	10.57	97	0.000	0.000	0.000
GR–MT shock (10%)	1.52	4	123.27	1130	0.000	0.000	0.000
GR–MT shock (20%)	1.53	4	127.13	1168	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
Quadruple world oil price	2.22	6	13.57	117	0.000	0.000	0.001
Double world oil price	1.53	4	11.40	104	0.000	0.000	0.006
Total time	23.60		467.03				

Multimod version: MULTAQ, simulation period: 50, N–R convergence criteria: 0.001, F–T type 2 convergence criteria: 0.00001.

fairly tight convergence criteria and many more Type II iterations. Indeed, the fact that F–T methods can only replicate L–B–J results with very tight convergence criteria suggests that the latter technique is exceptionally accurate in finding the 'true' solution.

For some shocks in MULTIMOD it can take more than 50 yr to achieve a new steady-state equilibrium especially if the shock changes the desired net-foreign-asset position of the economy. Tables 9 and 10 redo the experiments

Table 9
L–B–J method versus Fair–Taylor in TROLL

	L–B–J		Fair–Taylor		Maximum Difference		
	Time (min)	Type 1 Iterations	Time (min)	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-*G* shock (10%)	5.43	5	96.73	303	0.009	0.010	0.380
US-*BT* shock (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
US population shock (1%)	4.38	4	7.10	22	0.030	0.057	0.213
US population shock (2%)	4.38	4	11.52	36	0.055	0.089	1.246
FR-*G* shock (5%)	4.38	4	8.38	26	0.025	0.033	1.073
FR-*BT* shock (5%)	3.30	3	0.72	2	0.006	0.001	0.406
GR–MT shock (10%)	4.38	4	53.15	166	0.006	0.002	0.449
GR–MT shock (20%)	4.38	4	55.45	173	0.012	0.004	0.845
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
Quadruple world oil price	6.48	6	7.08	22	0.003	0.004	0.319
Double world oil price	4.38	4	4.85	15	0.002	0.005	0.037
Total time	73.38		564.88				

Multimod version: MULTAQ, simulation period: 150, N–R convergence criteria: 0.001, F–T type 2 convergence criteria: 0.002.

Table 10
L–B–J method versus Fair–Taylor in TROLL

	L–B–J		Fair–Taylor		Maximum Difference		
	Time (min)	Type 1 Iterations	Time (min)	Type 2 Iterations	*RS* (% pts)	*RL* (% pts)	*ER* (%)
US-*G* shock (5%)	4.42	4	163.72	499	0.000	0.000	0.003
US-*G* shock (10%)	5.42	5	173.35	543	0.000	0.000	0.003
US-*BT* shock (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
US population shock (1%)	4.38	4	110.78	347	0.000	0.000	0.002
US population shock (2%)	4.37	4	141.58	444	0.000	0.000	0.004
FR–G shock (5%)	4.38	4	153.73	447	0.000	0.000	0.003
FR–BT shock (5%)	3.30	3	92.18	290	0.000	0.000	0.002
FR–BT shock (5%)	3.30	3	92.18	290	0.000	0.000	0.002
US-*G* shock (10%)	5.42	5	173.35	543	0.000	0.000	0.003
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
Quadruple world oil price	6.55	6	97.05	288	0.000	0.000	0.002
Double world oil price	4.85	4	88.32	259	0.000	0.000	0.003
Total time	72.98		3,342.56				

Multimod version: MULTAQ, simulation period: 150, N–R convergence criteria: 0.001, F–T type 2 convergence criteria: 0.00001.

considered in the first two tables with a simulation horizon increased from 50 to 150 yr. One of the important advantages of the L–B–J method is that solution times are approximately a linear function of the simulation horizon. Indeed, if we compare the L–B–J results in Table 9 with the L–B–J results in Table 5 – or Table 10 with Table 8 – 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 3. This is not the case for algorithms such as F–T. In some cases, solution time increases by less than a factor of 3 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 depends 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. The tables also report summary measures of the total time required to run all of the shocks with each method. As can be seen in the tables, it takes approximately 24 min to execute all of the shocks with the L–B–J method for a horizon 50 periods and 73 min when the horizon is extended to 150 yr. Of course, as mentioned above the results for F–T will depend on desired accuracy. When tight convergence criteria are used to guarantee precision in the estimates for all these shocks (see Tables 8 and 10) it would take about 467 min to solve the model for 50 yr and about 3343 min to solve the model for 150 yr. With a shorter horizon of 50 yr it takes F–T about 20 times longer to find the solutions and with a horizon of 150 yr it takes about 46 times longer. These estimates provide some indication of the relative efficiency gains of the L–B–J method. Obviously, if experienced users knew what convergence criteria were required to achieve accurate results with F–T they could always reset Type II convergence criteria in a way that depends on the shock. However, this sort of fine tuning would be difficult to manage reliably in an environment where the model is being constantly changed in order to examine new issues.

5.3. Experiences with first-order techniques in SLIM

As was mentioned, SLIM users at the University of Strathclyde and at various other mainly UK institutions have reported problems in solving MULTIMOD with first-order techniques. This section briefly describes some of the experiences that we encountered when we attempted to replicate some of the same experiments that were reported in the earlier tables. Table 11 reports an indication of the convergence problems of first-order methods. As can be seen in the table the model failed to converge in a number of cases either because the shock was too large or we were unsuccessful in finding the appropriate damping factors, convergence criteria or feasible orderings of the model.

Obviously, we did not consider all possible orderings of MULTIMOD. The number of possible orderings of a simultaneous equation model becomes an extremely large number as the model expands beyond a few equations. The number of possible orderings of an N equation model stacked for M periods is NM factorial! This has created nontrivial problems for users of first-order methods because simulation performance can depend intricately on the ordering of the equations and it can be a fairly laborious process to find an ordering that works. Hughes Hallet and Piscitelli (this issue of the JEDC) have developed

Table 11
Some results from Fair–Taylor in SLIM[a]

Shock	Size	Results
US-*G*	5%	Converges
US-*G*	10%	Converges
US-*BT*	5%	Converges
US-*BT*	10%	Converges
US-*YCAP*	5%	Fails
US-*YCAP*	10%	Fails
US-*MT*	5%	Converges
US-*MT*	10%	Fails
US-*ER*	0.5	Fails
BT in all countries	5%	Converges
MT in all countries	10%	Fails
MT in all countries	5%	Fails

[a]Abridged from Pioro et al. (1996).

a re-ordering algorithm that starts from an initial ordering and attempts to find better orderings.[18] Based on some models of dimension 6 or less they argue that their re-ordering algorithm can result in significant efficiency gains. It is unclear how their re-ordering algorithm performs on multicountry models such as MULTIMOD or individual country models with much larger dimension.[19]

These types of convergence problems are consistent with the experiences of other MULTIMOD users outside the Fund that have attempted to implement what we would consider to be very moderate changes – from the point of view of

[18] Their re-ordering algorithm was unable to find a better ordering for the Bank of England's model (45 equations) or the COMPACT model (22 equations). There is some confusion in the literature about the absolute speed of Newton-based methods on problems of these dimensions. For example, Fisher et al. (1986) and Hughes Hallett and Fisher (1990) argue that second-order methods are inferior to first-order methods because it is computationally expensive to evaluate and invert matrices of partial derivatives. With the computers that are on the desktops of economists, problems of the dimensions considered by Hughes Hallett and Piscitelli (1997) can be solved in under a second even with sparse-matrix inversion code that dates back to the early 1970s.

[19] Users are not only interested in how quickly one finds a solution but they are also concerned about how accurate the solutions are. In general, numerical errors in solutions obtained from first-order solution techniques will be a function of the convergence criterion as well as the specific ordering that is chosen for the model. Hughes Hallett and Piscitelli (1997) do not report the numerical errors in their solutions, so it is unclear if their re-ordering algorithm produces accurate solutions in less time.

simulation robustness – to the structure of the model. This paper suggests that in many of these cases where there are convergence problems with first-order techniques users are strongly recommended to try alternative algorithms *before* concluding that it is a problem with the basic structure of the model.

6. Experiences with more recent versions of MULTIMOD

Simulation speed is only one criteria for comparing alternative algorithms. A much 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 mentioned earlier, there is no guarantee that first-order algorithms 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 – see, for an example, Bartolini et al. (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 et al. (1996).

7. Conclusions

The development and use of forward-looking macromodels in policy making 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. In many cases, these 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. Indeed, the numerical complexity of solving a forward-looking macromodel is considerably more onerous than solving a traditional backward-looking reduced-form model of the same size. The enormous advance of inexpensive computer technology over the last few years has finally made it possible to design and implement more robust and efficient solution algorithms.

This paper compared the performance of a new algorithm based on a Newton–Raphson iterative method to more traditional first-order techniques such as the Fair–Taylor algorithm. Relative to traditional algorithms in use today for solving MULTIMOD, we find that this algorithm is considerably faster and much less prone to simulation failure. Because the algorithm can also 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 involved in the development of national models.

Acknowledgements

We thank Raouf Boucekkine, Ken Judd, Jean-Pierre Laffargue, Pierre Malgrange and Steve Symansky, for encouraging us to do work in this area. We also thank Peter Hollinger for developing the new stacked-time algorithm in Portable TROLL and an anonymous referee who provided comments on a preliminary version of this paper. Finally, we would be remiss if we did not thank Helen Hwang, Victoria Ejikeme and Susanna Mursula for excellent technical assistance. Naturally, they are not responsible for any errors or omissions.

References

Armstrong, J., Black, R., Laxton, D., Rose, D., 1995. A robust method for simulating forward-looking models. The Bank of Canada's New Quarterly Projection Model, Part 2, Technical Report No. 73, Ottawa, Canada.

Bartolini, L., Razin, A., Symansky, S., 1995. G-7 fiscal restructuring in the 1990s: macroeconomic effects. Economic Policy 20, 111–146.

Bayoumi, Tamim, Coe, D.T., Helpman, E., 1996. R&D Spillovers and Global Growth. IMF Working Paper No. WP/96/47, International Monetary Fund, Washington.

Black, R., Laxton, D., Rose, D., Tetlow, R., 1994. The steady state model: SSQPM. The Bank of Canada's New Quarterly Projection Model, Part 1, Technical Report No. 72, Ottawa, Canada.

Blanchard, O.J., Kahn, C.M., 1980. The solution of linear difference models under rational expectations.. Econometrica 48, 1305–1311.

Boucekkine, R., 1995. An Alternative Methodology for Solving Nonlinear Forward-Looking Models. Journal of Economic Dynamics and Control 19 (4), 711–734.

Bryant, R.C., Hooper, P., Mann, C.L., 1993. Evaluating policy regimes: New research in emperical macroeconomics, The Brookings Institution, Washington.

Coletti, D., Hunt, B., Rose, D., Tetlow, R., 1996. The dynamic model: QPM. The Bank of Canada's New Quarterly Projection Model, Part 3, Technical Report No. 75, Ottawa, Canada.

Edison, H.J., Marquez, J.R., Tryon, R.W., 1987. The structures and properties of the federal reserve board multicountry model. Economic Modeling 4, 115–315.

Fair, R.C., Taylor, J.B., 1983. Solution and maximum likelihood estimation of dynamic nonlinear rational expectations models. Econometrica 51 (4), 1169–1185.

Fisher, P.G., Holly, S., Hughes Hallett, A.J., 1986. Efficient solution techniques for dynamic non-linear rational expectations models. Journal of Economic Dynamics and Control 10, 139–145.

Fisher, P.G., 1990. SLIM: User's Manual For Ver. 81, University of Warwick.

Fisher, P.G., 1992. Rational Expectations in Macroeconomic Models. Kluwer Academic Publishers, Dordrecht.

Gagnon, J.E., 1991. A Forward-Looking Multicountry Model for Policy Analysis: MX3. Economic and Financial Computing 1, 311–361.

Hall, S.G., 1985. On the solution of large economic models with consistent expectations. Bulletin of Economic Research 37, 157–161.

Helliwell, J.F., Meredith, G., Bagnoli, P., Durand, Y., 1990. INTERMOD 1.1: A G-7 Version of the IMF's Multimod. Economic Modelling 7, 3–62.

Hollinger, P., 1996. The Stacked-Time Simulator in TROLL: A robust algorithm for solving forward-looking models. Paper presented at the 2nd Int. Conf. on Computing in Economics and Finance, Geneva, Switzerland.

Hughes Hallett, A.J., Fisher, P.G., 1990. On economic structures and model solution methods: Or should econometricians use newton methods for model solution? Oxford Bulletin of Economics and Statistics 52 (3) 317–330.

Hughes Hallett, A.J., Piscitelli, L., 1997. Simple reordering techniques for expanding the convergence radins of first order iterative techniques. Journal of Economic Dynamics and Control (this issue).

Hughes Hallett, A.J., Ma, Y., Yin, Y.P., 1996. Hybrid algorithms with automatic switching for solving nonlinear equations systems in economics. Journal of Economic Dynamics and Control 20, 1051–1071.

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.

Laffargue, J.P., 1990. Résolution d'un modèle macroéconomique avec anticipations rationnelles. Annales d'Economie et Statistique 17, 97–119.

Laxton, D., Meredith, G., Rose, D., 1995. Asymmetric effects of economic activity on inflation. IMF Staff papers Vol. 42, No. 2 International Monetary Fund, Washington, 344–374.

McKibbin, W.J., Sachs, J.D., 1991 Global linkages: Macroeconomic interdependance and cooperation in the world economy, The Brookings Institution, Washington.

Masson, P.R., Symansky, S., Meredith, G., 1990. MULTIMOD MARK II: A revised and extended model. IMF Occasional Paper No. 71, International Monetary Fund, Washington.

Pioro, H., McAdam, P., Laxton, D., 1996. Solving MULTIMOD with first-Order and Newton-Based Techniques. University of Strathclyde Discussion Paper Working.

Pauletto, G., 1995. Solution and simulation of macroeconometric models, Thesis, Department of Econometrics, University of Geneva, Switzerland.

Press, W., Teukolsky, S., Vetterling, W., Flannery, B., 1992. Numerical Recipes In C: The Art of Scientific Computing. 2nd ed., Cambridge University Press, New York.

Reid, J.K., 1977. Sparse Matrices. In: Jacobs, D. (Ed.), The State of the Art in Numerical Analysis. London Academic Press, , pp. 85–146.

Taylor, J.B., 1988. The Treatment of Expectations in Large Multicountry Econometric Models. Bryant, Henderson, and others, 161–182.