



HAL
open science

Inexact Newton methods for model simulation

Stefania Bellavia, Silvia Magheri, Claudia Miani

► **To cite this version:**

Stefania Bellavia, Silvia Magheri, Claudia Miani. Inexact Newton methods for model simulation. International Journal of Computer Mathematics, 2011, pp.1. 10.1080/00207160.2011.563844 . hal-00710049

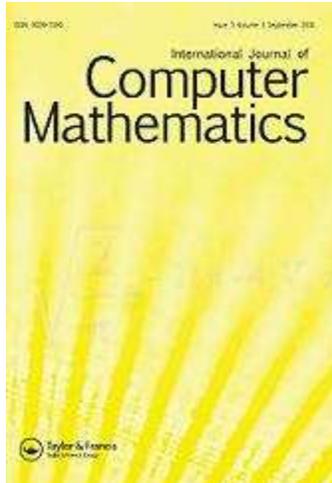
HAL Id: hal-00710049

<https://hal.science/hal-00710049>

Submitted on 20 Jun 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Inexact Newton methods for model simulation

Journal:	<i>International Journal of Computer Mathematics</i>
Manuscript ID:	GCOM-2010-0181-B.R1
Manuscript Type:	Original Article
Date Submitted by the Author:	14-Dec-2010
Complete List of Authors:	Bellavia, Stefania; University of Florence, Dipartimento di Energetica "S.Stecco" Magheri, Silvia; University of Florence, Dipartimento di Energetica "S.Stecco" Miani, Claudia; Bank of Italy, Economic Outlook and Monetary Policy Department
Keywords:	Inexact Newton methods, globalization strategy, preconditioners, forward looking models, matrix-free

SCHOLARONE™
Manuscripts

Inexact Newton methods for model simulation

Stefania Bellavia [§] Silvia Magheri Claudia Miani [¶]

14 dicembre 2010

Abstract

Robust and efficient solution techniques for solving macroeconomic models are increasingly becoming a key factor in developing models employed by policy-making institutions for policy simulations and forecasting. Traditionally, when solved in presence of forward-looking variables, these models are nonlinear, large-scale and sparse and give rise to large and highly structured nonlinear systems.

This paper proposes a Newton-GMRES method obtained tuning up the basic algorithm by properly choosing the forcing terms sequence and the preconditioning strategy. In addition, the Newton-GMRES method is wrapped into a globalization strategy based on a non monotone linesearch technique in order to enlarge its convergence basin and to enhance its robustness. The combination of these ingredients yields a reliable method with low memory requirements.

Numerical experiments using the MULTIMOD model and a basic Real Business Cycle model are presented. A Matlab code based on this approach is provided.

keywords Inexact Newton methods, globalization strategy, preconditioners, forward looking models, matrix-free

1 Introduction

Nonlinear forward-looking models solved for forward consistent expectations give rise to large scale and highly structured nonlinear systems. In order to cope with the high dimension of these systems, several algorithms were designed (see for example [7, 24, 23, 14, 21, 20, 17, 34]). In the Extended Fair-Taylor approach [14] the nonlinear system is split into small blocks and then an iterative procedure is used to ensure consistency across blocks until the full system converges. The horizon is extended to guarantee a stable path within the range

[§]Dipartimento di Energetica, Università di Firenze, Via Lombroso 6/17, 50134, Firenze, Italy, e-mail: stefania.bellavia@de.unifi.it, homepage: <http://ciro.de.unifi.it/~bellavia>

[¶]Bank of Italy, Economic Outlook and Monetary Policy Department. claudia.miani@bancaditalia.it

of simulation. Peculiarity of this approach is the small memory requirement, but in some situations it may fail to converge or it converges slowly. A widely used technique is the Newton-Raphson based Laffargue-Boucekkine-Juillard (L-B-J) algorithm [7, 23, 20]. In this approach the block-structure of the Jacobian matrix of the nonlinear system is exploited in the linear algebra phase of the Newton-method in order to save memory storage.

Interestingly, other techniques are possible and procedures based on Newton-Krylov methods have been employed by Gilli and Pauletto [17, 34] to solve forward-looking models. Krylov methods [25] are a broad class of iterative linear algebra methods that includes GMRES [39], which is of particular interest here. Newton-Krylov methods are combination of Newton-type procedures for nonlinear systems and Krylov methods for solving the arising Newton linear systems. Specifically, if we seek for a zero of a nonlinear function $F : \mathbb{R}^N \rightarrow \mathbb{R}^N$, at a generic iteration k of a Newton-Krylov method, a Krylov solver is applied to the Newton equation

$$F'(y^k)s = -F(y^k), \quad (1)$$

where y^k is the current iterate and F' is the Jacobian matrix of the nonlinear function F . Typically, the Krylov solver is applied to the Newton equation, until an iterate, say \bar{s} satisfies

$$\|F(y^k) + F'(y^k)\bar{s}\| \leq \bar{\eta}_k \|F(y^k)\|,$$

where $\eta^k \in (0, 1)$ is commonly called *forcing term*. The vector \bar{s} is an *Inexact Newton step* and is used to form the next iterate. Good performance of Newton-Krylov methods have been observed in the solution of a great variety of large scale nonlinear systems (see for example [37, 35, 1, 4] and the references therein) and in the solution of forward-looking models, as it is highlighted by the works of Gilli and Pauletto [17, 34].

Here we focus on the numerical solution of forward-looking models by Newton-GMRES methods, that are Newton-Krylov methods where the iterative solver GMRES [39] is used in the linear algebra phase. An interesting feature of Newton-GMRES methods is that they require only the action of the Jacobian of the linear system on vectors. This allows matrix-free implementations in which these products are approximated without forming and storing $F'(y^k)$. For this reason they are called matrix-free methods. This characteristic is shared with all Newton-Krylov methods employing a transpose-free iterative linear solver as GMRES, TFQMR [15], BICGSTAB [40]. The matrix-free feature of Newton-GMRES enables saving computational time everytime forming the Jacobian represents a significant fraction of the total execution time. We emphasize that when the analytic Jacobian is not available, approximating it by finite differences is generally the dominant cost of a Newton-type methods employing either iterative or direct solver for computing the Newton step.

Starting from Gilli and Pauletto approach, we move a step further investigating on some algorithmic choices that are crucial for efficiency and robustness of Newton-GMRES methods. First, we focus on the accuracy requirements for the computation of the Inexact Newton step. Well chosen forcing terms allow

1
2
3
4
5
6
7
8 to improve the efficiency of the method. Forward-looking models often show a
9 low level of nonlinearity and solving the Newton systems with high accuracy is
10 unproductive. In other words, small forcing terms in these applications, lead to
11 oversolve the Newton equation, by which we mean imposing an accuracy on the
12 solution of the Newton system that it is not necessary to produce a progress
13 towards the solution of the nonlinear system. Then, in the applications we are
14 dealing with, the choice of constant and relatively large forcing terms leads to
15 save linear iterations.

16 Second, we turn our attention to preconditioning issues. In fact, Krylov
17 iterative solvers sometime suffer from a slow convergence and therefore a pre-
18 conditioner for the Newton system must be used. The aim of preconditioning is
19 to reduce the number of Krylov solver iterations. Unfortunately, the need of a
20 preconditioner, in principle destroys the matrix-free feature of Newton-Krylov
21 solvers. Widely used algebraic preconditioner are Jacobian-based, requiring the
22 evaluation of the Jacobian blocks at each nonlinear iteration, which is computa-
23 tionally expensive. A possible alternative is given by the frozen preconditioner
24 that is reusing the same preconditioner on several Newton iterations. Very of-
25 ten, the performance of a frozen preconditioner deteriorates with the progress
26 of the nonlinear iterations. This is not true in this context, as given the weak
27 nonlinearities of economic models, the Jacobian of the arising nonlinear systems
28 does not change significantly from one nonlinear iteration to another and high
29 accuracy in the solution of the Newton equation is not required. Indeed, from
30 our numerical experience we noticed only a mild deterioration of the frozen pre-
31 conditioner's performance as the nonlinear process goes on. Therefore, in our
32 approach the preconditioner is computed only at the first nonlinear iteration
33 and then it is reused in all the subsequent iterations. Then, the combination of
34 frozen preconditioners and matrix-free methods allows to avoid the computation
35 of the Jacobian at each nonlinear iteration.

36 Third, we embed the Newton-GMRES method into a nonmonotone line-
37 search globalization strategy. In fact, the guarantee of convergence of Newton-
38 Krylov methods is local, as in any Newton-like method, and the numerical
39 method may fail in finding a solution even though a well defined solution ex-
40 ists. The globalization strategy allows to enlarge the convergence region of the
41 Newton-GMRES method and enhances its robustness. Line-search globalization
42 strategies date back to 70's and are widely used in the numerical optimization
43 community, but to our knowledge they are seldom used in computational econo-
44 metrics, despite their use allows to obtain convergence in economic experiments
45 where great magnitude of the shocks are involved.

46 Combining the previously outlined ingredients we obtain a procedure that
47 requires low computational cost and memory requirements, moreover the con-
48 vergence is guaranteed provided that the nonlinear function defining the model
49 is continuously differentiable in a region containing the generated sequence.

50 We show the behaviour of our procedure on the MULTIMOD (*MULTI-region*
51 *econometric MODEL*) model, developed by the International Monetary Fund
52 [30, 31, 28] and we compare our approach with the original Gilli and Pauletto
53 approach [17] and with the Newton method obtained computing the newton
54
55
56
57
58
59
60

step by direct methods for sparse and large linear systems. Moreover, in order to get more insight into the the behaviour and the robustness of our proposal we apply our procedure to nonlinear systems arising from a real business cycle models given in [26] and we compare its performance with those of the Newton method employing direct solvers, with and without globalization strategy.

A Matlab code, named NGMS (Newton GMRES for Model Simulation), based on this approach has been developed and will be freely accessible through the web site: <http://ngms.de.unifi.it>. With NGMS we intend to provide a theoretically well-founded solver that could be a valid tool for the numerical solution of forward looking models.

2 Forward looking models

The presence of forward (rational) expectations introduces a dependency of the solution in period t on anticipated changes at future dates. Solving for consistent expectation means that the rational expectations are given by the conditional expectation of the model. The forward dependency requires the specification of a 'terminal point' beyond the (length- T) horizon of the simulation. We stack the model period-by-period and solve simultaneously through T periods. Consider, at period t the following nonlinear forward-looking model:

$$\begin{cases} f_1(y_{t-\bar{r}}, \dots, y_t, \dots, y_{t+\bar{k}}, z_t, a) = 0 \\ \vdots \\ f_n(y_{t-\bar{r}}, \dots, y_t, \dots, y_{t+\bar{k}}, z_t, a) = 0 \end{cases} \quad (2)$$

where $y_t \in \mathbb{R}^n$ are the endogenous variables at period t , $z_t \in \mathbb{R}^n$ are the exogenous variables at period t , the parameters $a \in \mathbb{R}^p$ are given. Here, \bar{r} indicates the maximum lag and \bar{k} the maximum lead. Due to the presence of leads and lags, solving the model (2) in $[t_1, t_f]$, requires \bar{r} initial conditions and \bar{k} final conditions: i.e. the values of y_{t_1-i} , $i = 1, \dots, \bar{r}$ and y_{t_f+j} , $j = 1, \dots, \bar{k}$. Once these conditions are available the equations are stacked from period t_1 to period t_f and the following nonlinear system is obtained:

$$F(y) = 0 \quad F : \mathbb{R}^{nT} \rightarrow \mathbb{R}^{nT}, \quad y \in \mathbb{R}^{nT} \quad (3)$$

where $T = t_f - t_1 + 1$,

$$y = \begin{bmatrix} y_{t_1} \\ \vdots \\ y_{t_f} \end{bmatrix} \in \mathbb{R}^{nT}, \quad F(y) = \begin{bmatrix} F_1(y) \\ \vdots \\ F_T(y) \end{bmatrix} \quad (4)$$

$$F_j(y) = \begin{bmatrix} f_1(y_{t_j-\bar{r}}, \dots, y_{t_j}, \dots, y_{t_j+\bar{k}}, z, a) \\ \vdots \\ f_n(y_{t_j-\bar{r}}, \dots, y_{t_j}, \dots, y_{t_j+\bar{k}}, z, a) \end{bmatrix}, \quad j = 1, \dots, T. \quad (5)$$

Then, given the matrices:

$$D_j(y) = \frac{\partial F_j(y)}{\partial y_j} \in \mathbb{R}^{n \times n}, \quad j = 1, \dots, T$$

$$E_{i,j}(y) = \frac{\partial F_i(y)}{\partial y_{i-j}} \in \mathbb{R}^{n \times n}, \quad i = 2, \dots, T \quad j = 1, \dots, \min(i-1, \bar{r})$$

$$A_{i,j}(y) = \frac{\partial F_i(y)}{\partial y_{i+j}} \in \mathbb{R}^{n \times n}, \quad i = 1, \dots, T-1 \quad j = 1, \dots, \min(T-i, \bar{k})$$

the Jacobian F' of F exhibits the following block-banded structure:

$$F'(y) = \begin{bmatrix} D_1(y) & A_{1,1}(y) & \cdots & \cdots & A_{1,\bar{k}}(y) & \\ E_{2,1}(y) & D_2(y) & A_{2,1}(y) & \cdots & \cdots & A_{2,\bar{k}}(y) \\ E_{3,2}(y) & E_{3,1}(y) & D_3(y) & A_{3,1}(y) & \cdots & A_{3,\bar{k}-1}(y) \\ & \ddots & \ddots & \ddots & \ddots & \\ & & & E_{T,\bar{r}}(y) & \cdots & E_{T,1}(y) & D_T(y) \end{bmatrix}. \quad (6)$$

3 Newton-Krylov methods

In this section we briefly describe Newton-Krylov methods for the solution of nonlinear systems of equations. Given a system of nonlinear equations $F(y) = 0$, where $F : \mathbb{R}^N \mapsto \mathbb{R}^N$ is continuously differentiable, Newton-Krylov methods belong to the class of Inexact Newton methods [11], iterative processes that result in the following scheme, where the superscript k denotes the iteration's number:

Let y^0 be given.

For $k = 0$ until "convergence" do:

Find some $\bar{\eta}^k \in [0, 1)$ and \bar{s}^k that satisfy

$$\|F(y^k) + F'(y^k)\bar{s}^k\| \leq \bar{\eta}^k \|F(y^k)\|, \quad (7)$$

Set $y^{k+1} = y^k + \bar{s}^k$.

Clearly, these methods are variants of Newton's method in which at each iteration the Newton equation

$$F'(y^k)s = -F(y^k), \quad k \geq 0, \quad (8)$$

is solved only approximately.

In a Newton-Krylov framework, a Krylov linear solver is applied to (8) and it is iterate till an Inexact Newton step \bar{s}^k satisfying

$$\|F(y^k) + F'(y^k)\bar{s}^k\| \leq \bar{\eta}_k \|F(y^k)\|,$$

is computed. This approach is widely used whenever large scale nonlinear systems have to be solved.

Local convergence analysis for Inexact Newton methods [11] shows that, under standard conditions on the function F , if y^0 is sufficiently close to a solution y^* of the nonlinear system and the $\bar{\eta}^k$'s are uniformly bounded away from one, then the sequence $\{y^k\}$ converges to y^* . Moreover, the choice of the forcing term $\bar{\eta}^k$ has a strong relevance on the performance of Inexact Newton methods as it is shown in [11]. At this regard, we recall that, if $\{\bar{\eta}^k\} \rightarrow 0$ then $\{y^k\}$ exhibits local q-superlinear convergence to a solution y^* , while if $\{\bar{\eta}^k\} = O(F(y^k))$ the convergence rate is quadratic. In [13] two choices of $\bar{\eta}^k$'s yielding up to quadratic rate of convergence have been proposed.

Let us now focus on Newton-GMRES, namely the Newton-Krylov method where GMRES is used to compute the Inexact Newton step.

GMRES (Generalized Minimal RESidual) is an iterative Krylov method for solving indefinite systems of linear equations [39]. GMRES applied to (8) minimizes the residual norm $\|F(y^k) + F'(y^k)s\|$ over all corrections in the current Krylov subspace. Because iterates are based on this norm minimization property, the method does not break down in exact arithmetic, i.e., a new iterate can always be constructed if the current iterate is not the solution, provided the coefficient matrix is nonsingular.

An attractive feature of GMRES applied to the Newton equation (8) is that it requires only the action of the Jacobian F' on a vector v . For an appropriately chosen scalar ϵ , this action can be approximated by finite differences ([25])

$$F'(y)v \sim \frac{F(y + \epsilon v) - F(y)}{\epsilon}, \quad (9)$$

giving rise to a process that is referred to as “matrix-free”. In other words, a Newton-GMRES method can be implemented without requiring the computation and the storage of the Jacobian F' .

4 Newton-GMRES for model simulation

Here, we focus on the solution of (3) and we describe the approach taken in implementing a Newton-GMRES method specially designed for this class of problems. We assume that the Jacobian is not available and we approximate the blocks of the Jacobian that are needed to build the preconditioner, by finite differences.

More precisely, we describe our choice of the forcing terms sequence $\{\bar{\eta}^k\}$, of the preconditioner and of the employed non monotone globalization strategy. Finally, we give the algorithm summarizing the adopted approaches.

4.1 The forcing terms

Our choice of the forcing terms sequence is strictly related to the nature of macroeconomic models. Indeed, these models generally do not show an high level of nonlinearity and are equipped with a good approximation of the solution. In fact, even in presence of "moderate" shocks, the approximation given by the historical series are good enough and classical Newton-methods converge in a very low number of iterations. Then, it is not necessary to solve the Newton equation (8) to high accuracy, as a crude approximation of the Newton step is very often good enough to produce progress towards the solution. This way, pointless iterations of the iterative solver GMRES are avoided. In other words, in this type of applications, a less accurate solution of (8) may be both cheaper and effective enough in reducing the norm of F .

Then, despite a forcing sequence $\{\bar{\eta}^k\}$ converging to zero produces a fast convergence of the Inexact-Newton method to the solution, due to the characteristic of the class of models we are interested in, here we choose a constant forcing sequence with $\bar{\eta}^k = \bar{\eta}^0$ for any k .

4.2 The preconditioner

Effective preconditioning of the Newton systems is the most crucial ingredient for obtaining good performance of Newton-Krylov methods. The term preconditioning refers to transforming the Newton system into another system with more favourable properties for the iterative solver (GMRES in our implementation). A preconditioner for the k -th Newton system is a matrix P^k such that the preconditioned system takes the following form:

$$(P^k)^{-1}F'(y^k)s = -(P^k)^{-1}F(y^k).$$

Generally speaking, if P^k is a good preconditioner for $F'(y^k)$, we expect that the iterative solver converges faster on the preconditioned system than on the original one. When GMRES, or any iterative solver, is applied to the preconditioned system it is not necessary to form the preconditioned matrix $(P^k)^{-1}F'(y^k)$ explicitly; the action of the preconditioned matrix on a vector v is accomplished approximating the vector $w = F'(y^k)v$ by finite-differences and computing $z = (P^k)^{-1}w$ solving a linear system of the form $P^k z = w$ ($(P^k)^{-1}$ is not explicitly known). Algebraic preconditioner gives the preconditioner in a factorized and sparse form so that the solution of such linear systems is not an expensive task [6].

In our implementation of the Newton-GMRES method for forward-looking models, we provide two different preconditioners. Our choice of the preconditioners is strictly connected to the choice of the forcing terms: as we solve the linear systems only to a modest accuracy we can rely on unexpensive preconditioners, which are good enough to provide a step \bar{s}^k satisfying (7) within a reasonable number of GMRES iterations, despite are not able to really capture the ill-conditioning of the Jacobian matrix.

A quite common preconditioner for block-banded matrix as (6) evaluated at y^k , is the so called Jacoby preconditioner, i.e.

$$P^k = \begin{bmatrix} D_1(y^k) & & & \\ & D_2(y^k) & & \\ & & \ddots & \\ & & & D_T(y^k) \end{bmatrix}. \quad (10)$$

Anyway, as observed in [34], blocks $D_j(y^k)$ for $j = 1, \dots, T$ share the same structure. In fact, at the k -th nonlinear iteration, $D_j(y^k) = \frac{\partial F_j(y^k)}{\partial y_j}$ with F_j given in (4) and the only difference among the F_j is the period they are referred to. Therefore, in [17], at each Newton-GMRES iteration, the block diagonal preconditioner $P^k = \text{diag}(D_1(y^k), \dots, D_T(y^k))$ is used, with $D_1(y^k)$ given by the (1,1) block diagonal of $F'(y^k)$. Then, following this strategy, at iteration k of the Newton-GMRES method only the (1,1) block diagonal of $F'(y^k)$ needs to be evaluated. Further, its LU decomposition has to be computed, since at each GMRES iteration a linear system with coefficient matrix P^k has to be solved. Gilli and Pauleto provided interesting numerical results adopting this approach.

However, due to the low level of nonlinearity of these models it is not necessary to recompute the preconditioner at each nonlinear iteration. In our approach, we compute the preconditioner only at the first Newton-GMRES iteration and then we freeze it, that is we use the same preconditioner throughout the whole iterative process. This way, we need only to compute the (1,1) block diagonal of $F'(y^0)$ and its LU decomposition. Summarizing, at each Newton-GMRES iteration, we use the preconditioner

$$P^k \equiv P_D = \text{diag}(D_1(y^0), \dots, D_T(y^0)) \quad k \geq 0 \quad (11)$$

where $D_1(y^0)$ is given by the (1,1) block diagonal of $F'(y^0)$. In the sequel we will refer to this preconditioner with the name P_D (Block-Diagonal Preconditioner).

However, when T becomes large, this preconditioner may become less effective. Then, we rely on the following more expensive preconditioner and at each nonlinear iteration we employ

$$P^k \equiv P_B = \begin{bmatrix} D_1(y^0) & A_{1,1}(y^0) & \cdots & A_{1,\tilde{k}}(y^0) \\ E_{2,1}(y^0) & D_2(y^0) & A_{2,1}(y^0) & \cdots & A'_{2,\tilde{k}}(y^0) \\ E_{3,2}(y^0) & E_{3,1}(y^0) & D_3(y^0) & \cdots & A_{3,\tilde{k}-1}(y^0) \\ & \ddots & \ddots & \ddots & \\ & & E_{T,\tilde{r}}(y^0) & \cdots & D_T(y^0) \end{bmatrix} \quad (12)$$

with $\tilde{k} \leq \bar{k}$ and $\tilde{r} \leq \bar{r}$. In the sequel P_B (Block-Banded Preconditioner) will indicate preconditioner (12).

Clearly, when $\tilde{k} = \bar{k}$ and $\tilde{r} = \bar{r}$ we have $P_B = F'(y^0)$, while when $\tilde{k} = 0$ and $\tilde{r} = 0$, P_B reduces to:

$$P_B = \text{diag}(D_1(y^0), \dots, D_T(y^0)). \quad (13)$$

In any case, we remark that we always freeze the preconditioner, that is the factorization of the preconditioner is computed only at the first nonlinear iteration. The blocks of $F'(y^0)$ required by the preconditioner are approximated by finite differences.

4.3 The globalization strategy

In the approach detailed above the linear systems are solved only to a modest accuracy and as a consequence we rely on a very rough approximation of the Newton step. Then we may need to merge our Inexact-Newton method into a globalization strategy. This allows us to monitor the progress of the method toward a solution of the nonlinear system and to reduce the value of the merit function $\|F\|^2$ even if we do not have at disposal a good step to update the current iterate. Moreover, in some situation, for example when severe shocks are used or when we simulate the model for forecasting, the initial guess provided by the historical data is not good enough and Newton-like methods fail. In these cases, a globalization strategy is fundamental to enlarge the convergence basin of a Newton-like method and to enhance its robustness.

There are two major categories of globalizations: linesearch methods in which step lengths are adjusted (usually shortened) to obtain a satisfactory step and trust region methods in which a step is ideally chosen minimizing a local model of $\|F\|^2$ within a specified trust-region (see [25, 12, 32] for details). A further possibility is to employ homotopy methods. In this approach an easy problem with well known solution is gradually transformed into the original problem $F(y) = 0$, building an homotopy map or varying the parameter values in case of parameter dependent nonlinear systems. Homotopy methods are known to be more robust than Newton-like methods embedded into a linesearch or a trust-region strategy, but they are generally more expensive and become complicated near a bifurcation point, requiring sophisticated approaches [32]. Moreover, we are not aware of implementations of homotopy methods in a matrix-free setting as the Jacobian is generally required to devise efficient and theoretical supported implementations of these strategies. Regarding trust-region methods, dogleg strategy in conjunction with Inexact-Newton methods can be devised [36], but at least an approximation of the Jacobian is required. Then, in our opinion, in this context the linesearch strategy is the best option for the following reasons: easy to implement, matrix-free compatible, strong theoretically supported in conjunction with Inexact Newton methods.

Inexact Newton methods embedded into a linesearch strategy result in the following scheme. At iteration k , the Inexact Newton step \bar{s}^k is computed and the new iterate has the form $y^{k+1} = y^k + \lambda \bar{s}^k$ where $0 < \lambda \leq 1$ is such that

$$\|F(y^{k+1})\|^2 < (1 - \alpha\lambda)\|F(y^k)\|^2 \quad (14)$$

with $\alpha \in (0, 1)$. It is quite important to note, at this stage, that an Inexact Newton step satisfying (7) with $\eta^k < 1$ is ensured to be a descent direction for $\|F(y)\|^2$ at y^k , i.e.

$$(\nabla\|F(y^k)\|^2)^T \bar{s}^k < 0,$$

see [8]. Therefore, the existence of λ satisfying (14) is guaranteed.

Moreover, our numerical experience enlightened that a nonmonotone strategy is enough to enhance robustness of Newton-GMRES methods when applied to forward-looking models. In other words, it is not necessary to impose a monotone decrease of the merit function $\|F\|^2$, but it is enough to obtain a reduction within $q \geq 1$ iterations. Then, the acceptance criterion (14) is relaxed and the nonlinear residual at y^{k+1} is required to satisfy the following condition:

$$\|F(y^{k+1})\|^2 < (1 - \alpha\lambda)M^k \quad (15)$$

where

$$M^k = \max\{\|F(y^k)\|^2, \|F(y^{k-1})\|^2, \dots, \|F(y^{k-q})\|^2\}, \quad (16)$$

$q \geq 0$ is a given integer and the superscript k indicates the iteration number. The choices $q > 0$ and $q = 0$ correspond to a nonmonotone and to a monotone strategy, respectively.

4.4 The algorithm and the Matlab code

Now, we sketch the algorithm that is the core of our code.

NGMS Method

Let $y^0, \eta \in (0, 1), \alpha \in (0, 1), 0 < \theta_m < \theta_{max} < 1, q \geq 1, P, \tilde{r}, \tilde{k}, G, M_nt, tol_nt$ be given.

1. If $P=1$

Compute by finite differences the (1,1) diagonal block of $F'(y^0)$ and perform its LU factorization.

else

Compute by finite differences the preconditioner (12) and perform its LU factorization.

2. For $iter = 1, \dots, M_nt$

2.1 Using GMRES (with preconditioner P_D given in (11) if $P=1$, with preconditioner P_B given in (12) otherwise), compute \bar{s}^k such that

$$\|F(y^k) + F'(y^k)\bar{s}^k\| \leq \eta\|F(y^k)\|.$$

2.2 Set $\lambda = 1$.

2.3 If $G=1$ (Perform the linesearch strategy)

2.3.1 Compute M^k by (16).

2.3.2 While $\|F(y^k + \lambda\bar{s}^k)\|^2 \geq (1 - \alpha\lambda)M^k$ do:

Choose $\theta \in [\theta_m, \theta_{max}]$.

Update $\lambda = \theta\lambda$.

- 2.4 Set $y^{k+1} = y^k + \lambda \bar{s}^k$.
 2.5 If $\|F(y^{k+1})\| < tol_nt$ then EXIT

The previous algorithm is the core of the Matlab code NGMS. We have striven to make NGMS as simple to use as possible. The use of NGMS requires only a minimal description of the problem to be solved: the size of the problem, the initial approximate solution, the stopping tolerances tol_nt , the number of allowed nonlinear iteration M_nt and a user-supplied function for evaluating the non linear function (2). At this regard, it should be underlined that in many macroeconometrics models the variables can be reordered so that the nonlinear systems exhibits a large block of interdependent equations, which is preceded and followed by recursive equations. This way, only the block of interdependent equations is needed to be solved by the NGMS method. Our Matlab code is structured in order to exploit this decomposition whenever it is available.

Finally, the user is required to specify the value of the forcing term (that is taken constant throughout all the iterative process), the choice of the preconditioner and if the globalization strategy is desired.

The parameters P , \tilde{r} and \tilde{k} , in input govern the choice of preconditioner. If P is set to one the preconditioner P_D given in (11) is used. Its factors L and U are computed at the beginning of the iterative process and are then used at each iteration of GMRES to solve the linear systems with coefficient matrix given by the preconditioner P_D . With the choice $P=2$, $\tilde{k} = 0$ and $\tilde{r} = 0$, the preconditioner P_B given by (13) is used. In this case, as well as in the more expensive case, $\tilde{k} > 0$ and/or $\tilde{r} > 0$ the LU factorization of the whole preconditioner is computed. The LU factorizations needed to factorize the preconditioner are computed with the built-in Matlab function `lu` with syntax that allows to apply LU factorization of sparse matrices developed in UMFPAK package [10]. In this case preordering and symbolic analysis that limit fill-in in the LU factorization are carried out.

In Step, 2.1, we use GMRES(m) (GMRES with restart) with $m = 150$, a maximum of 10 restarts and with null initial guess. If GMRES is not able to provide a step \bar{s}_k satisfying (7) within the maximum number of allowed iterations, if

$$\frac{\|F(y^k) + F'(y^k)\bar{s}^k\|}{\|F(y^k)\|} < 1 \quad (17)$$

we go on with the last step computed by GMRES, otherwise the NGMS Matlab code stops with a failure. This choice is motivated by the fact that if (17) is satisfied, then \bar{s}^k is a descent direction for $\|F\|^2$ at y^k . Therefore, if the globalization strategy is activated, \bar{s}^k may be good enough to produce progress towards the solution, even if it does not solve the Newton equation to the prescribed accuracy.

Products of $F'(y^k)$ with vectors that are required by GMRES are approximated using the finite-difference formula given in (9) with

$$\epsilon = \sqrt{\epsilon_m} \|y^k\| / \|v\|,$$

where ϵ_m is the machine precision.

In Step 2.3.1-2.3.2 the nonmonotone linesearch strategy is performed, if it is required by the user. That is, if the parameter G that governs this choice is set to 1. In the linesearch strategy, moving along the direction of the initial Inexact step \bar{s}^k , successively shorter steps of the form $s^k = \lambda \bar{s}^k$ are selected until the sufficient decrease condition (15) is met; this process is commonly referred to as *backtracking*. In our Matlab implementation, the parameter α is set to 10^{-4} , q is set to 6 and the choice of the scalar θ is performed using the three-point parabolic model as described in [25]. If the sufficient decrease condition (15) is not satisfied within 10 backtracks, we go on with the last computed value of λ .

In Step 2.5 the tolerance tol_nt is used to control the level of accuracy and the algorithm is stopped as soon as the stopping criterion

$$\|F(y^{k+1})\| < tol_nt \quad (18)$$

is met. If the above stopping criterion is not satisfied within M_nt iterations the algorithm stops with a failure. In the NGMS Matlab code the process is terminate with an error condition in this case.

As a final remark, we would like to stress that, if the globalization strategy is adopted ($G = 1$), the Inexact-Newton method that is the basement of our approach is globally convergent provided that the nonlinear function F is continuously differentiable in an open set containing the generated sequence. Then, if this condition is satisfied, the convergence does not depend in a critical way on the closeness of the initial guess to the solution. Moreover, eventually the Inexact Newton step s^k is taken. In other words, the decrease condition (15) is satisfied with $\lambda = 1$.

5 Numerical results

In this section we report on our numerical experience with the code NGMS.

The tests were conducted on a Intel Xeon (TM) 3.4 Ghz, 1GB RAM using MATLAB 7.0. The machine precision is $\epsilon_m \simeq 2.10^{-16}$.

The initial guess y^0 was those provided by the hystorical data, M_nt was set to 100 and $tol_nt = 10^{-6}$.

Results are reported in tables where the following data are shown: the preconditioner used in the column P, the number it_new of performed nonlinear iterations, the average number A_gmit of performed GMRES iterations, the norm of the function F at the final iterate in column $\|F\|$, the execution time in seconds in column **time**, the time (in seconds) spent to build the preconditioner and its LU factorization in column **time_prec**, the number N_bt of performed backtracks in the linesearch strategy. Failures of the method due to a breakdown in the linear algebra phase (GMRES is not able to provide an Inexact Newton step that satisfies (17)) are indicated in the tables by the symbol "-".

T	P	<i>it_new</i>	<i>A_gmit</i>	$\ F\ $	time
5	P_D	21	3	7.d-7	3.3
10	P_D	27	7	4.d-7	6.1
20	P_D	25	15	5.d-7	11.6
30	P_D	24	22	9.d-7	19.6
50	P_D	24	47	8.d-7	57.9
50	P_B	24	25	5.d-7	83.2
60	P_D	24	70	9.d-7	107.1
60	P_B	24	30	5.d-7	107.3
80	P_D	-	-	-	-
80	P_B	23	52	9.d-7	192.7
100	P_B	-	-	-	-

Table 1: Temporary shock: CA_G is increased by 1 per cent of CA_GDP in $t_1 = 2001$, $\eta = 0.5$.

5.1 Multimod model

In the first sets of experiments we apply NGMS to the MULTIMOD (*MULTI-region econometric MODeL*) model, version Mark III, developed by the International Monetary Fund [30, 31, 28]. This is a forward-looking model with 601 equations, a maximum lag of 3 periods, and a maximum lead of 10 periods. To our knowledge the analytic jacobian of the model is not available.

In order to analyze the computational performance of NGMS, we conducted experiments with different choices of the forcing term η . We used constant forcing terms and we compare the behaviour of our code solving the linear systems to a modest accuracy ($\eta = 0.5$ and $\eta = 0.1$) and to a tight accuracy ($\eta = 10^{-4}$). Moreover, our runs were performed varying the value of T . More precisely, we considered nonlinear systems arising by stacking T periods, with $T = 5, 10, 20, 30, 50, 60, 80, 100$. As a result, the dimension of the problems solved varies from 3005 to 60100. We always activate the globalization strategy if necessary, and employ preconditioner P_D . Moreover, in order to compare the behaviour of the preconditioners, problems corresponding to $T \geq 50$ have been solved using both preconditioners P_D and P_B given by (13), that is P_B with $\tilde{k} = \tilde{r} = 0$.

In tables 1-3 we report the results obtained with different values of η and different values of T . We started the simulation at $t_1 = 2001$ and shocked the variable of Canada's government expenditures by the 1% of Canadian GDP for the first year of simulation. In these simulations the linesearch strategy is never activated.

As we can see from these tables and we can expect from the theory, the number of outer iterations required by the Newton-GMRES method grows as the accuracy in the solution of the nonlinear systems decreases.

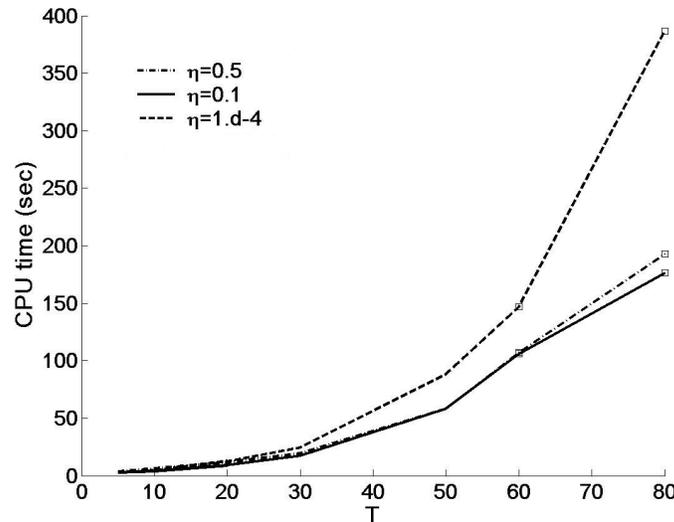
On the other hand, the average number of GMRES iterations obviously

T	P	<i>it_new</i>	<i>A_gmit</i>	$\ F\ $	time	time_prec
5	P_D	8	8	3.d-7	3	0.9
10	P_D	8	20	5.d-7	4	0.9
20	P_D	8	41	2.d-7	9	0.9
30	P_D	8	64	1.d-7	17.3	0.9
50	P_D	8	130	1.d-7	53.9	0.9
50	P_B	8	70	1.d-7	76.4	47.3
60	P_D	8	203	1.d-7	95.4	0.9
60	P_B	8	82	1.d-7	96.1	55.2
80	P_D	-	-	-	-	-
80	P_B	8	134	4.d-7	171.8	77.3
100	P_B	-	-	-	-	-

Table 2: Temporary shock: CA_G is increased by 1 per cent of CA_GDP in $t_1 = 2001$, $\eta = 0.1$.

T	P	<i>it_new</i>	<i>A_gmit</i>	$\ F\ $	time
5	P_D	3	31	1.d-7	2.5
10	P_D	3	73	1.d-7	4.5
20	P_D	3	162	1.d-7	12.3
30	P_D	3	234	2.d-7	22.6
50	P_D	3	521	7.d-7	78.5
50	P_B	3	276	6.d-7	93.7
60	P_D	4	719	3.d-9	132.3
60	P_B	4	324	3.d-9	121.3
80	P_D	-	-	-	-
80	P_B	4	753	3.d-9	369.5
100	P_B	-	-	-	-

Table 3: Temporary shock: CA_G is increased by 1 per cent of CA_GDP in $t_1 = 2001$, $\eta = 10^{-4}$.

Figure 1: temporary shock with different η

increases with decreasing η and becomes large whenever $\eta = 10^{-4}$ is used. On this set of problems preconditioner P_B is more effective than preconditioner P_D : the average number of GMRES iterations reduces greatly when P_B is used. However, the cost of computing and factorizing the preconditioner must be paid of. In fact, the cost of computing and factorizing P_B is T times those of computing and factorizing P_D and this cost dominates the cost of the whole procedure for $T \leq 50$. This is shown in Table 2, where the time spent to build the preconditioner and its LU factorization is reported. We underline that this time is independent on T , whenever the preconditioner P_D is used. Moreover, this time does not depend on the forcing term η adopted. As a result, in the solution of tests with $T \leq 50$, even if less GMRES iterations are performed, the execution time with P_B is higher than with P_D , independent on the chosen forcing term.

On the other hand, we can observe that the Newton systems become more difficult to be solved as T increases. In fact, the number of GMRES iterations increases with T and for $T > 50$, P_B becomes competitive. Finally, for $T > 60$, P_D does not accelerate the convergence of GMRES and this yields a failure of NGMS. Then, for $T > 60$, P_B has to be used. We would like to underline that failures of the method due to failures of GMRES in providing an Inexact Newton step satisfying (17) are not to be ascribed to the choice of the maximum number of allowed GMRES iterations or to the freezing of the preconditioner. Indeed, these failures occur in the first nonlinear iteration and are not recovered even allowing a greater number of GMRES iterations before restarting (namely GMRES(250)). The point is that the preconditioner is not effective and GMRES

T	P	$\eta = 0.1$			$\eta = 10^{-4}$		
		<i>it_new</i>	<i>A_gmit</i>	time	<i>it_new</i>	<i>A_gmit</i>	time
5	P_D	8	8	10.7	3	31	4.6
10	P_D	8	20	11	3	73	6.6
20	P_D	8	41	15.6	8	163	14.3
30	P_D	8	63	23	3	233	24.8
50	P_D	8	127	59.8	3	520	81.2
50	P_B	8	70	411.5	3	276	187.5
60	P_D	8	195	115.4	4	720	136.5
60	P_B	8	81	500.3	4	323	226.4
80	P_D	-	-	-	-	-	-
80	P_B	8	134	718.9	4	720	585.3
100	P_B	-	-	-	-	-	-

Table 4: Temporary shock: CA_G is increased by 1 per cent of CA_GDP in $t1 = 2001$, recomputed preconditioner

gets stuck.

Looking to the overall computational time, in Figure 1 we plot the execution time versus T , for different values of η . For $T > 60$, we report the execution times with preconditioner P_B .

From this picture it is clear that it is not necessary to solve the linear systems with a tight accuracy and the best choice seems to be $\eta = 0.1$.

Moreover, we underline that the behaviour of the code on these tests does not depend on the freezing of the Jacobian. In Table 4 we report, both for $\eta = 0.1$ and $\eta = 10^{-4}$, the number of performed GMRES iterations and the execution time, when the Jacobian is recomputed at each nonlinear iterations. The comparison between this table and Tables 2 and Table 3 makes clear that recomputation of the preconditioner does not yield reduction in the number of GMRES iterations and obviously increases the execution time of the method. The increase is more evident when $\eta = 0.1$ is used, as the method performs a higher number of nonlinear iteration and this calls for a higher number of preconditioner recomputations.

All things considered, in the remaining of the subsection $\eta = 0.1$ is used, P_D (frozen) is adopted whenever $T \leq 50$, while P_B (frozen) is employed for the larger tests.

In passing we note that Gilli and Pauletto in [17] adopted the choice $\eta = 10^{-4}$ combined with the recomputation of the Jacobian at each nonlinear iteration. Therefore, our numerical experience (see Tables 2 and Table 4) makes clear that our choice of using greater η (i.e. $\eta = 0.1$) and freezing the preconditioner yields to an improvement of Gilli and Pauletto's proposal. This is still true if the analytic jacobian is available, since there is not any advantage in recomputing the preconditioner in terms of GMRES iterations and the benefit of using a

greater η does not depend on the time spent at the first nonlinear iteration to approximate the Jacobian's blocks needed to build the preconditioner.

In order to get more insight into the behaviour of our implementation of Newton-GMRES methods, we compare our approach with the "exact" Newton method obtained employing direct methods for sparse and large linear systems. More precisely, the Newton method has been implemented solving the Newton equation by the built-in Matlab function `lu` with syntax that allows to apply LU factorization of sparse matrices developed in UMFPAK package [10]. We have run the Newton method with $T = 5, 10, 20, 30, 50$. In all cases it takes four nonlinear iterations to get the required approximate solution. Then, as expected, the convergence of the nonlinear procedure is slower when only approximate solutions of the linear systems are computed. However, the execution time is dramatically higher (from 134 seconds with $T = 5$ till 2894 seconds with $T = 50$), when the exact approach is adopted. We underline the 95% percent of the execution time is spent in the approximation of the nonzero blocks of the Jacobian, while the time spent to solve the Newton equation is negligible because the built-in Matlab function `lu` is a C- precompiled function. This clearly shows that the use of a matrix-free approach combined with a frozen preconditioner allows to obtain a very efficient procedure as the approximation of the Jacobian at each nonlinear iteration is avoided. Moreover, also a great saving in terms of memory storage is obtained as only the diagonal blocks of the Jacobian need to be stored. On the opposite, a large fill-in occurs in the LU factorization of the Jacobian, even if the built-in Matlab function `lu` carries out reordering and symbolic analysis to limit fill-in. This can be seen in figure 2 where we plot the sparsity pattern of the original Jacobian, along with the sparsity factors of its factors L and U . These plots are obtained with $T = 30$ and the Jacobian has been evaluated at the initial guess used to perform the runs.

The second set of experiments is carried out shocking with a permanent 10 per cent and 20 per cent increase in CA.MT starting in $t_1 = 2001$. This tests are more difficult to be solved as we can see from Table 5. In fact, we needed to activate the linesearch strategy in order to get the convergence of the method in the majority of the runs.

Our final set of experiments with the Multimod model is carried out in order to investigate if the use of the globalization strategy allows us to solve models with "severe" shocks. This can be seen from Table 6, where, for $T = 50$, we report the obtained results using eight shocks; namely, 3 increasing shocks to real Government expenditure (G) of France, 20% increase in the target money supply (MT) of Canada, US, and Germany and an increase of 10% and 20% to the target money supply of all industrial countries. We can see from this table that, thanks to the globalization strategy, we managed to solve all the tests. At this regard it should be underlined that for some tests, a great number of backtracks has been performed and the globalization strategy was crucial to obtain the convergence. In fact the classical Newton method without globalization strategy fails on these tests.

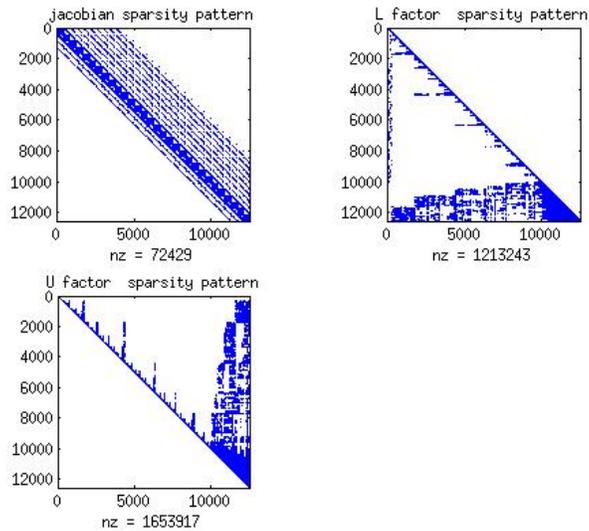


Figure 2: sparsity pattern and number of nonzeros entries of the Jacobian and its factors.

T	P	shock 10%			shock 20%		
		<i>it_new</i>	<i>A_gmit</i>	<i>N_bt</i>	<i>it_new</i>	<i>A_gmit</i>	<i>N_bt</i>
5	P_D	10	10	1	11	10	1
10	P_D	10	23	1	11	22	1
20	P_D	9	44	0	10	47	1
30	P_D	9	67	0	10	70	1
50	P_D	10	157	1	10	142	1
60	P_B	10	100	1	12	122	5
80	P_B	18	220	22	22	240	34
100	P_B	53	457	173	86	510	341

Table 5: Permanent shock: permanent 10 per cent and 20 per cent increase in CA.MT starting in $t_1 = 2001$, $\eta = 0.1$.

variable	shock	<i>it_new</i>	<i>A_gmit</i>	<i>N_bt</i>	$\ F\ $	time
FR_G	5%	10	113	0	6.d-7	62.1
FR_G	10%	11	112	0	2.d-7	68.5
FR_G	20%	16	101	3	3.d-7	86.6
CA_MT	20%	10	142	1	7.d-7	78.2
US_MT	20%	21	310	35	2.d-7	356.8
GR_MT	20%	30	348	71	2.d-7	575.2
MT all countries	10%	18	268	23	3.d-7	261.9
MT all countries	20%	22	300	39	7.d-7	358.4

Table 6: Different kind of permanent shock with $T = 50$, $\eta = 0.1$ e $P = P_D$.

5.2 Real Business Cycle model

In this subsection numerical results obtained with NGMS applied to a real business cycle model are shown. Simulating the real business cycle models [27, 29] is a popular topic in modern dynamic macroeconomics. The model specification follows from the solution of dynamic optimization problems under uncertainty by optimizing agents populating the economy. The set of conditions characterizing the equilibrium gives rise to a nonlinear dynamic system of equations. Here, we use a basic RBC model with monopolistic competition, calibrated for the US economy, as shown in [26]. When the framework is deterministic, the RBC model is used with the assumption of perfect foresight. This means that, when the effect of a temporary technology shock is analysed, at the beginning of the period the shock is revealed to everyone. Armed with this information agents then make their optimal consumption and investment decisions. Most often the model is supposed to be in a state of equilibrium in the first period, when the shock hits the economy. The purpose of the simulation is to describe the reaction to the shock, until the system returns to the original state of equilibrium (if the model is stable). Besides that, we study the response to a permanent shock to technology, simulating the model trajectory in a long enough horizon to come permanently to a new state of equilibrium. The basic RBC model is a forward-looking model with 8 equations, lag and lead of 1 period. We solved the model stacking $T = 2000$ periods with $t_1 = 1$ and $t_f = 2001$, so that we obtain a nonlinear system of dimension 16000×16000 . Based on the computational experience gained on the MULTIMOD model we solved the nonlinear systems arising from this model using $\eta = 0.1$. We performed runs using preconditioner P_D , preconditioner P_B with $k = \tilde{r} = 0$ (see (13)) and the banded preconditioner P_B with one lower and upper block diagonals, that is $\tilde{k} = \tilde{r} = 1$. Note that in this latter case, $P_B \equiv F'(y_0)$. Observe that the matrix $F'(y^0)$ is very sparse and we have verified that only moderate fill-in occurs during its LU factorization. Then, factors L and U remain really very sparse.

We start with a temporary (nine periods) shock to the level of technology set to the value of 0.1 (the level of technology at steady state is zero). Then,

shock	P	it_new	A_gmit	N_bt	$\ F\ $	time
0.1	P_D	6	335	0	4.d-7	140.8
0.1	P_B	4	1.2	0	3.d-7	7.7
0.3	P_D	7	353	0	4.d-7	169.7
0.3	P_B	5	1.8	0	3.d-7	8.8
0.5	P_D	7	352	0	3.d-7	164.4
0.5	P_B	5	1.8	0	2.d-7	8.3
0.8	P_D	-	-	-	-	-
0.8	P_B	7	3.6	2	3.d-7	11.2
1.0	P_B	9	6.3	5	8.d-7	13.2

Table 7: Temporary (nine periods) shock to the level of technology; $\eta = 0.1$, preconditioners P_D and P_B with $\tilde{k} = \tilde{r} = 1$.

we increase the value of the shock up to 1. In theory the model goes back asymptotically to initial equilibrium no matter of the size of the shock. When we simulate numerically the model equations, it happens that big shocks (beyond 0.8) cause the model does not converge at all when a Newton-type method not equipped with a globalization strategy is employed. The numerical procedure presented here, thanks to the globalization strategy, permits to simulate the response to shocks of magnitude 1, as shown in Table 7, where results obtained with both preconditioners P_D and P_B with $\tilde{k} = \tilde{r} = 1$ are given.

Some comments on Table 7 are in order. We observe that the construction and LU factorization of P_D takes about 10^{-2} seconds where building and factorizing P_B requires about 5.6 second. However, Table 7 shows that our approach with P_B is really more efficient than with P_D , despite the higher P_B construction cost. This is due to the fact that preconditioner P_D is not effective even in the first nonlinear iteration, where it is built on information of the current Jacobian. As a result, a great number of linear iterations is performed. On the contrary, preconditioner P_B is the true Jacobian at the first nonlinear iteration and its performance does not deteriorate at all as the nonlinear process goes on. As a result, Inexact Newton steps are computed with an extremely low number of GMRES iterations. So, the great savings in the number of linear iterations compensates the extra work for computing P_B . We performed runs also with the block diagonal preconditioner P_B given in (13) ($\tilde{k} = \tilde{r} = 0$). However, despite in this case all diagonal blocks of $F'(y^0)$ are retained, the preconditioner shows the same behaviour as P_D does. In other words, retaining the diagonal blocks instead of approximating them with $D_1(y^0)$ does not enhance preconditioner's performance. We would like to stress that this behaviour of block diagonal preconditioners P_D and P_B does not depend on the fact that they are taken frozen. This is evident from Figure 3, where, the performance of the frozen preconditioners and of their recomputed counterparts, in the solution of the nonlinear system corresponding to the shock's value 0.3, are

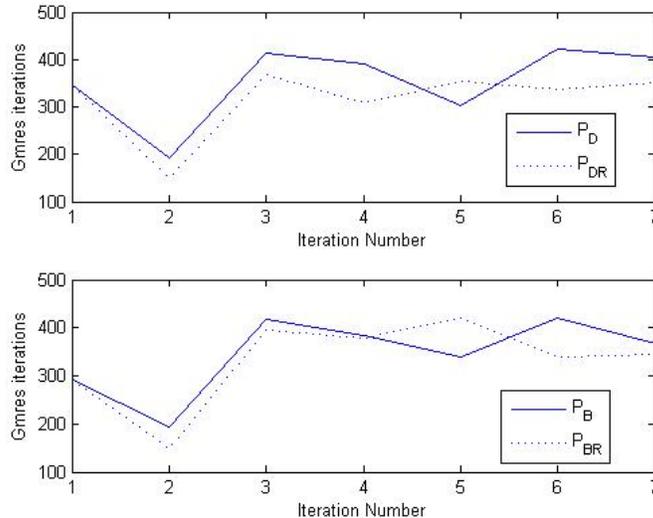


Figure 3: Comparison between frozen and recomputed preconditioner: GMRES(m) iterations versus the nonlinear iterations.

compared. Let P_{DR} and P_{BR} be the block diagonal preconditioners recomputed at each nonlinear iteration, i.e. $P_{DR} = \text{diag}(D_1(y^k), \dots, D_1(y^k))$, for $k \geq 0$ and $P_{BR} = \text{diag}(D_1(y^k), \dots, D_T(y^k))$, for $k \geq 0$. In the upper part of Figure 3, we plot the number of GMRES(m) iterations performed at each nonlinear iteration using the frozen preconditioner P_D and the recomputed preconditioner P_{DR} . The same chart with P_B and P_{BR} is produced in the bottom part. We can see from this picture that the frozen preconditioner slightly deteriorates with respect to the recomputed one and the recomputation of the preconditioner at each nonlinear iteration does not improve substantially its quality.

Note also that the globalization strategy is activate with larger shocks. As the shock increases, the nonlinear systems become more difficult to be solved and, when it is greater than 0.5, GMRES with preconditioner P_D fails in producing the Inexact Newton step at the first nonlinear iteration, so this failure is not to be ascribed to the fact that the preconditioner is frozen.

We would like to stress that using preconditioner P_B with $\tilde{k} = \tilde{r} = 1$ does not produce memory saving with respect to a Newton approach employing direct methods, as $P_B = F'(y^0)$, but the computation of the Jacobian at each nonlinear iteration is avoided and a more efficient procedure is obtained. We underline also that only the preconditioner is frozen, while the current Jacobian is “felt” through the approximation of its action on a vector by finite-differences. So, the method employed here is different from a traditional modified Newton method where the Jacobian is held frozen. In table 8 we report the results obtained with the “exact” Newton method obtained solving the Newton equation by the built-

shock	EN_G				EN		
	<i>it_new</i>	<i>N_bt</i>	$\ F\ $	time	<i>it_new</i>	$\ F\ $	time
0.1	3	0	1.d-10	18.9	3	1.d-10	18.9
0.3	4	0	6.d-12	25.0	4	6.d-12	25.0
0.5	4	1	1.d-9	25.1	4	3.d-7	24.9
0.8	5	2	2.d-10	32.3	8	3.d-7	49.3
1.0	5	4	1.d-9	32.3	-	-	-

Table 8: Temporary (nine periods) shock to the level of technology; $\eta = 0.1$.

in Matlab function `lu` with syntax that allows to apply LU factorization of sparse matrices developed in UMFPACK package [10]. We report results obtained using the exact Newton method with and without globalization strategy. In the Table, these two procedures are indicated as EN_G (Exact Newton with Globalization) and EN (Exact Newton), respectively.

The comparison between Table 7 and Table 8 shows that the inexact approach with preconditioner P_B clearly outperforms EN_G in terms of computational time. Moreover, Table 8 enlightens that the globalization strategy is crucial to obtain convergence for the largest shock even when the Newton direction is computed by a direct solver. Indeed, Newton method without globalization strategy fails in simulating response to the biggest shock. Further, the globalization strategy speeds-up the convergence of the Newton method when a shock of magnitude 0.8 is applied, reducing the overall computational cost.

Finally, we would like to underline that both NGMS and EN_G fail in solving the model with shocks of magnitude greater than 1 despite the globalization strategy; these failures are to be ascribed to the fact that the nonlinear function F describing the model is not defined everywhere and both procedures generate an iterate y^k that does not belong to the domain of the function F .

The experiments with a permanent shock to the level of technology are even more revealing the robustness and efficiency of our technique. We can deal with permanent shocks of magnitude up to 0.3, as it is shown in Table 9. We underline that the globalization strategy was crucial to obtain convergence for shocks greater than 0.1. Indeed, both the inexact method and its exact counterpart fail on these tests if the globalization strategy is not employed. Finally, again failures for shocks values greater than 0.3 of both the inexact and the exact approach, are to be ascribed to the fact that both methods do not handle the situation where the nonlinear function is not defined everywhere in \mathbb{R}^n .

6 Conclusion and Perspective

We have presented our numerical experience with Newton-GMRES methods for forward-looking models. In our approach the Jacobian of the model is not re-

shock	P	it_new	A_gmit	N_bt	$\ F\ $	time
0.1	P_B	6	2.5	0	1.d-7	9.1
0.2	P_B	7	3.7	1	2.d-7	10.2
0.3	P_B	7	4.5	4	1.d-7	10.9

Table 9: Permanent shock to the level of technology; $\eta = 0.1$, preconditioner P_B .

quired and the method requires limited memory storage. The Newton-GMRES method is wrapped into a nonmonotone linesearch strategy and this allows to solve also models with severe shocks. The reported numerical results suggest that the related Matlab code NGMS could be a valid tool for the numerical solution of forward-looking models. However, it should be taken into account that the variables involved in these models are not meaningful everywhere; for example, some variables need to be positive in order to have an economical meaning. Moreover the domain of the nonlinear function F does not coincide with \mathbb{R}^n . Our numerical experimentation makes clear that when a Newton-like method is employed to solve these models, it may happen that an iterate y^k does not belong to the domain of the function F and the procedure breaks down. In the NGMS code, this event is reported with a warning message. It seems then desirable, to develop a variant of this code, able to overcome this failures and to cope with the constraints that need to be imposed to the variables of the model. Approaches able to handle the bounds on the variable have been proposed in [2, 5, 3], but they cannot be realized in a matrix-free manner.

References

- [1] S. Bellavia and S. Berrone: *Globalization strategies for Newton-Krylov methods for stabilized FEM discretization of Navier-Stokes equations*, Journal of Computational Physics, Vol. 226, pp 2317-2340, 2007.
- [2] S. Bellavia, M. Macconi, B. Morini, *STRSCNE: A scaled trust-region solver for constrained nonlinear equations*, Comp. Opt. and Appl., Vol. 28, pp. 31-50, 2004.
- [3] S. Bellavia, M. Macconi, S. Pieraccini, *Constrained Dogleg Methods for nonlinear systems with simple bounds*, Tech. Report 5/2009, Dipartimento di Energetica, Università di Firenze, 2009.
- [4] S. Bellavia and B. Morini, *A globally convergent Newton-GMRES subspace method for systems of nonlinear equations*, SIAM Journal Scientific Computing, Vol. 23 n. 3, pp. 940-960, 2001.
- [5] S. Bellavia and B. Morini, *Subspace Trust-Region methods for large bound constrained nonlinear equations*, SIAM Journal Numer. Anal., Vol. 44, pp. 1535-1555, 2006
- [6] M. Benzi, *Preconditioning Techniques for Large Linear Systems: A Survey*, Journal of Computational Physics, Vol. 182, pp. 418-477, 2002.
- [7] R. Boucekkine: *An Alternative Methodology for Solving Nonlinear Forward-Looking Models*, Journal of Economic Dynamics and Control, Vol. 19, No. 4, pp. 711734,1995
- [8] P. N. Brown and Y. Saad: *Convergence theory of nonlinear Newton-Krylov algorithms*, SIAM J. Optim., Vol. 4, pp. 297-330, 1994.
- [9] A.R. Conn, N.I.M. Gould, and P.L. Toint: *Trust-region methods*, SIAM, Philadelphia, 2000.
- [10] T. A. Davis: *Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method*, ACM Transactions on Mathematical Software, vol 30, pp. 196-199, 2004.
- [11] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., Vol. 19, pp. 400-408, 1982.
- [12] J.E. Dennis and Jr.R.B. Schnabel: *Numerical methods for unconstrained optimization and nonlinear equations*, Englewood Cliffs, N.J. Prentice-Hall, 1983.
- [13] S.C. Eisenstat and H.F. Walker: *Choosing the forcing terms in an inexact Newton method*, SIAM, J. Sci. Comput., Vol. 17, pp. 16-22, 1996.

- 1
2
3
4
5
6
7
8 [14] R.C. Fair and J.B. Taylor: *Solution and Maximum Likelihood Estimation of*
9 *Dynamic Nonlinear Rational Expectations Models* Econometrica, Vol. 51,
10 No. 4., pp. 1169-1185, Luglio 1983.
- 11 [15] R. W. Freund, *A transpose-free quasi-minimum residual algorithm for non-*
12 *Hermitian linear systems*, SIAM J. Sci. Comput., Vol. 14 , pp. 470-482,
13 1993.
- 14 [16] M.G. Gasparo: *A nonmonotone hybrid method for nonlinear systems*, Op-
15 *timization Methods and Software*, Vol. 13, pp. 79-94, 2000.
- 16 [17] M. Gilli and G. Pauletto: *Krylov Methods for Solving Models with Forward-*
17 *Looking Variables*, Journal of Economic Dynamics and Control, Vol. 22,
18 1275-1289, 1998.
- 19 [18] G.H. Golub and C.F. Ortega: *Matrix Computation*, Johns Hopkins, Balti-
20 *more*, 1989.
- 21 [19] R. Haas and P. Masson: *MINIMOD: Specification and Simulation Results*,
22 *Staff Papers*, International Monetary Fund, Vol. 33, pp 722-67, 1986.
- 23 [20] P. Hollinger *The Stacked-Time Simulator in TROLL: A Robust Algorithm*
24 *for Solving Forward-Looking Models*, paper presented at the Second In-
25 *ternational Conference on Computing in Economics and Finance*, Geneva
26 (Switzerland), 1996.
- 27 [21] P. Hollinger: *Beyond Newton: Robust Method for Solving Large Nonlinear*
28 *Models in TROLL*, Intex Solutions, Inc., Geneva (Switzerland), 2000.
- 29 [22] International Monetary Fund: [http://www.imf.org/external/np/res/mmod/](http://www.imf.org/external/np/res/mmod/index.htm)
30 [index.htm](http://www.imf.org/external/np/res/mmod/index.htm), Washigton DC.
- 31 [23] M. Juillard: *DYNARE: A program for the resolution and simulation*
32 *of dynamic models with forward variables through the use of a relax-*
33 *ation algorithm*, CEPREMAP Working Papers (Couverture Orange) 9602,
34 CEPREMAP, revised, 1996.
- 35 [24] M. Juillard, D. Laxton, P. McAdam and H. Pioro: *An algorithm compe-*
36 *tition: First-order iterations versus Newton-based techniques*, Journal of
37 *Economic Dynamics & Control*, Vol. 22 pp. 291-1318, 1998.
- 38 [25] C.T. Kelley: *Iterative Methods for Linear and Nonlinear Equations*, SIAM,
39 Vol. 16, 1995.
- 40 [26] R. G. King, S. T. Rebelo, *Resuscitating Real Business Cycles*. Handbook
41 *of Macroeconomics*, edited by John B. Taylor and Michael Woodford, Vol.
42 1B. Elsevier, 1999
- 43 [27] F. E. Kydland, E. C. Prescott, *Time to Build and Aggregate Fluctuations*,
44 *Econometrica*, Vol. 50, pp. 1345-1370, 1982.
- 45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

- 1
2
3
4
5
6
7
8 [28] D. Laxton, P. Isard, H. Faruqee, E. Prasad and B. Turtelboom: *MULTI-*
9 *MOD Mark III: The Core Dynamic and Steady-State Models*, Occasional
10 Paper 164, International Monetary Fund, 1998.
- 11 [29] J. Long, C. Plosser: *Real business cycles*, Journal of Political Economy, Vol.
12 91, pp. 36-69, 1983.
- 13 [30] P.R. Masson, S. Symansky, R. Haas and M. Dooley: *MULTIMOD: A Multi-*
14 *Region Econometric Model*, Staff Studies for the World Economic Outlook,
15 pp 50-104, International Monetary Fund, Washington DC, 1988.
- 16 [31] P.R. Masson, S. Symansky and G. Meredith: *MULTIMOD Mark II: A*
17 *Revised and Extended Model*, Occasional Paper 71, International Monetary
18 Fund, Washington DC, 1990.
- 19 [32] J. Nocedal, S.J. Wright: *Numerical Optimization*, Springer Series in Oper-
20 ation Research, 1999.
- 21 [33] C.F. Ortega and W.C. Rheinboldt: *Iterative Solution of Nonlinear Equa-*
22 *tions in Several Variables*, Academic Press, New York, 1970.
- 23 [34] G. Pauletto: *Computational solution of large-scale macroeconomic mod-*
24 *els*, Thesis, Department of Econometrics, University of Geneva, Switzer-
25 land, 1995.
- 26 [35] R. P. Pawlowski, J. N. Shadid, and J. P. Simonis H. F. Walker: *Globaliza-*
27 *tion techniques for Newton-Krylov methods and applications to the fully-*
28 *coupled solution of the Navier-Stokes equations*, SIAM Review, Vol. 48, pp.
29 700-721, 2006
- 30 [36] R. P. Pawlowski, J. P. Simonis, H. F. Walker, J. N. Shadid, *Inexact Newton*
31 *dogleg methods*, SIAM Journal Numer. Anal., Vol. 46, pp. 2112–2132, 2008
- 32 [37] M. Pernice and H. F. Walker: *NITSOL: a new iterative solver for nonlinear*
33 *systems*, SIAM Journal Sci. Comput., Vol. 19, pp. 302–318, 1998.
- 34 [38] Y. Saad: *Iterative Methods for Sparse Linear Systems, 2nd edition*, SIAM,
35 Philadelphia, PA, 2003.
- 36 [39] Y. Saad and M. Schultz: *GMRES: A Generalized Minimal Residual Al-*
37 *gorithm for Solving Nonsymmetric Linear Systems*, SIAM J. Sci. Statist.
38 Comput. Vol. 7, 856-869, 1986.
- 39 [40] H.A., Van der Vorst, , *BI-CGSTAB: A fast and smoothly converging variant*
40 *of BI-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci.
41 Stat. Comput., Vol. 13, 631-644, 1992.
- 42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

Inexact Newton methods for model simulation

Stefania Bellavia [§] Silvia Magheri Claudia Miani [¶]

14 febbraio 2011

Abstract

Robust and efficient solution techniques for solving macroeconomic models are increasingly becoming a key factor in developing models employed by policy-making institutions for policy simulations and forecasting. Traditionally, when solved in presence of forward-looking variables, these models are nonlinear, large-scale and sparse and give rise to large and highly structured nonlinear systems.

This paper proposes a Newton-GMRES method obtained tuning up the basic algorithm by properly choosing the forcing terms sequence and the preconditioning strategy. In addition, the Newton-GMRES method is wrapped into a globalization strategy based on a non monotone linesearch technique in order to enlarge its convergence basin and to enhance its robustness. The combination of these ingredients yields a reliable method with low memory requirements.

Numerical experiments using the MULTIMOD model and a basic Real Business Cycle model are presented. A Matlab code based on this approach is provided.

keywords Inexact Newton methods, globalization strategy, preconditioners, forward looking models, matrix-free

1 Introduction

Nonlinear forward-looking models solved for forward consistent expectations give rise to large scale and highly structured nonlinear systems. In order to cope with the high dimension of these systems, several algorithms were designed (see for example [7, 24, 23, 14, 21, 20, 17, 34]). In the Extended Fair-Taylor approach [14] the nonlinear system is split into small blocks and then an iterative procedure is used to ensure consistency across blocks until the full system converges. The horizon is extended to guarantee a stable path within the range

[§]Dipartimento di Energetica, Università di Firenze, Via Lombroso 6/17, 50134, Firenze, Italy, e-mail: stefania.bellavia@de.unifi.it, homepage: <http://ciro.de.unifi.it/~bellavia>

[¶]Bank of Italy, Economic Outlook and Monetary Policy Department. claudia.miani@bancaditalia.it

of simulation. Peculiarity of this approach is the small memory requirement, but in some situations it may fail to converge or it converges slowly. A widely used technique is the Newton-Raphson based Laffargue-Boucekkine-Juillard (L-B-J) algorithm [7, 23, 20]. In this approach the block-structure of the Jacobian matrix of the nonlinear system is exploited in the linear algebra phase of the Newton-method in order to save memory storage.

Interestingly, other techniques are possible and procedures based on Newton-Krylov methods have been employed by Gilli and Pauletto [17, 34] to solve forward-looking models. Krylov methods [25] are a broad class of iterative linear algebra methods that includes GMRES [39], which is of particular interest here. Newton-Krylov methods are combination of Newton-type procedures for nonlinear systems and Krylov methods for solving the arising Newton linear systems. Specifically, if we seek for a zero of a nonlinear function $F : \mathbb{R}^N \rightarrow \mathbb{R}^N$, at a generic iteration k of a Newton-Krylov method, a Krylov solver is applied to the Newton equation

$$F'(y^k)s = -F(y^k), \quad (1)$$

where y^k is the current iterate and F' is the Jacobian matrix of the nonlinear function F . Typically, the Krylov solver is applied to the Newton equation, until an iterate, say \bar{s} satisfies

$$\|F(y^k) + F'(y^k)\bar{s}\| \leq \bar{\eta}_k \|F(y^k)\|,$$

where $\eta^k \in (0, 1)$ is commonly called *forcing term*. The vector \bar{s} is an *Inexact Newton step* and is used to form the next iterate. Good performance of Newton-Krylov methods have been observed in the solution of a great variety of large scale nonlinear systems (see for example [37, 35, 1, 4] and the references therein) and in the solution of forward-looking models, as it is highlighted by the works of Gilli and Pauletto [17, 34].

Here we focus on the numerical solution of forward-looking models by Newton-GMRES methods, that are Newton-Krylov methods where the iterative solver GMRES [39] is used in the linear algebra phase. An interesting feature of Newton-GMRES methods is that they require only the action of the Jacobian of the linear system on vectors. This allows matrix-free implementations in which these products are approximated without forming and storing $F'(y^k)$. For this reason they are called matrix-free methods. This characteristic is shared with all Newton-Krylov methods employing a transpose-free iterative linear solver as GMRES, TFQMR [15], BICGSTAB [40]. The matrix-free feature of Newton-GMRES enables saving computational time everytime forming the Jacobian represents a significant fraction of the total execution time. We emphasize that when the analytic Jacobian is not available, approximating it by finite differences is generally the dominant cost of a Newton-type methods employing either iterative or direct solver for computing the Newton step.

Starting from Gilli and Pauletto approach, we move a step further investigating on some algorithmic choices that are crucial for efficiency and robustness of Newton-GMRES methods. First, we focus on the accuracy requirements for the computation of the Inexact Newton step. Well chosen forcing terms allow

1
2
3
4
5
6
7
8
9
10
11
12
13
14
to improve the efficiency of the method. Forward-looking models often show a low level of nonlinearity and solving the Newton systems with high accuracy is unproductive. In other words, small forcing terms in these applications, lead to oversolve the Newton equation, by which we mean imposing an accuracy on the solution of the Newton system that it is not necessary to produce a progress towards the solution of the nonlinear system. Then, in the applications we are dealing with, the choice of constant and relatively large forcing terms leads to save linear iterations.

15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
Second, we turn our attention to preconditioning issues. In fact, Krylov iterative solvers sometime suffer from a slow convergence and therefore a preconditioner for the Newton system must be used. The aim of preconditioning is to reduce the number of Krylov solver iterations. Unfortunately, the need of a preconditioner, in principle destroys the matrix-free feature of Newton-Krylov solvers. Widely used algebraic preconditioner are Jacobian-based, requiring the evaluation of the Jacobian blocks at each nonlinear iteration, which is computationally expensive. A possible alternative is given by the frozen preconditioner that is reusing the same preconditioner on several Newton iterations. Very often, the performance of a frozen preconditioner deteriorates with the progress of the nonlinear iterations. This is not true in this context, as given the weak nonlinearities of economic models, the Jacobian of the arising nonlinear systems does not change significantly from one nonlinear iteration to another and high accuracy in the solution of the Newton equation is not required. Indeed, from our numerical experience we noticed only a mild deterioration of the frozen preconditioner's performance as the nonlinear process goes on. Therefore, in our approach the preconditioner is computed only at the first nonlinear iteration and then it is reused in all the subsequent iterations. Then, the combination of frozen preconditioners and matrix-free methods allows to avoid the computation of the Jacobian at each nonlinear iteration.

35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
Third, we embed the Newton-GMRES method into a nonmonotone line-search globalization strategy. In fact, the guarantee of convergence of Newton-Krylov methods is local, as in any Newton-like method, and the numerical method may fail in finding a solution even though a well defined solution exists. The globalization strategy allows to enlarge the convergence region of the Newton-GMRES method and enhances its robustness. Line-search globalization strategies date back to 70's and are widely used in the numerical optimization community, but to our knowledge they are seldom used in computational econometrics, despite their use allows to obtain convergence in economic experiments where great magnitude of the shocks are involved.

Combining the previously outlined ingredients we obtain a procedure that requires low computational cost and memory requirements, moreover the convergence is guaranteed provided that the nonlinear function defining the model is continuously differentiable in a region containing the generated sequence.

We show the behaviour of our procedure on the MULTIMOD (*MULTI-region econometric MODEL*) model, developed by the International Monetary Fund [30, 31, 28] and we compare our approach with the original Gilli and Pauletto approach [17] and with the Newton method obtained computing the newton

step by direct methods for sparse and large linear systems. Moreover, in order to get more insight into the the behaviour and the robustness of our proposal we apply our procedure to nonlinear systems arising from a real business cycle models given in [26] and we compare its performance with those of the Newton method employing direct solvers, with and without globalization strategy.

A Matlab code, named NGMS (Newton GMRES for Model Simulation), based on this approach has been developed and will be freely accessible through the web site: <http://ngms.de.unifi.it>. With NGMS we intend to provide a theoretically well-founded solver that could be a valid tool for the numerical solution of forward looking models.

2 Forward looking models

The presence of forward (rational) expectations introduces a dependency of the solution in period t on anticipated changes at future dates. Solving for consistent expectation means that the rational expectations are given by the conditional expectation of the model. The forward dependency requires the specification of a 'terminal point' beyond the (length- T) horizon of the simulation. We stack the model period-by-period and solve simultaneously through T periods. Consider, at period t the following nonlinear forward-looking model:

$$\begin{cases} f_1(y_{t-\bar{r}}, \dots, y_t, \dots, y_{t+\bar{k}}, z_t, a) = 0 \\ \vdots \\ f_n(y_{t-\bar{r}}, \dots, y_t, \dots, y_{t+\bar{k}}, z_t, a) = 0 \end{cases} \quad (2)$$

where $y_t \in \mathbb{R}^n$ are the endogenous variables at period t , $z_t \in \mathbb{R}^n$ are the exogenous variables at period t , the parameters $a \in \mathbb{R}^p$ are given. Here, \bar{r} indicates the maximum lag and \bar{k} the maximum lead. Due to the presence of leads and lags, solving the model (2) in $[t_1, t_f]$, requires \bar{r} initial conditions and \bar{k} final conditions: i.e. the values of y_{t_1-i} , $i = 1, \dots, \bar{r}$ and y_{t_f+j} , $j = 1, \dots, \bar{k}$. Once these conditions are available the equations are stacked from period t_1 to period t_f and the following nonlinear system is obtained:

$$F(y) = 0 \quad F: \mathbb{R}^{nT} \rightarrow \mathbb{R}^{nT}, \quad y \in \mathbb{R}^{nT} \quad (3)$$

where $T = t_f - t_1 + 1$,

$$y = \begin{bmatrix} y_{t_1} \\ \vdots \\ y_{t_f} \end{bmatrix} \in \mathbb{R}^{nT}, \quad F(y) = \begin{bmatrix} F_1(y) \\ \vdots \\ F_T(y) \end{bmatrix} \quad (4)$$

$$F_j(y) = \begin{bmatrix} f_1(y_{t_j-\bar{r}}, \dots, y_{t_j}, \dots, y_{t_j+\bar{k}}, z, a) \\ \vdots \\ f_n(y_{t_j-\bar{r}}, \dots, y_{t_j}, \dots, y_{t_j+\bar{k}}, z, a) \end{bmatrix}, \quad j = 1, \dots, T. \quad (5)$$

Then, given the matrices:

$$D_j(y) = \frac{\partial F_j(y)}{\partial y_j} \in \mathbb{R}^{n \times n}, \quad j = 1, \dots, T$$

$$E_{i,j}(y) = \frac{\partial F_i(y)}{\partial y_{i-j}} \in \mathbb{R}^{n \times n}, \quad i = 2, \dots, T \quad j = 1, \dots, \min(i-1, \bar{r})$$

$$A_{i,j}(y) = \frac{\partial F_i(y)}{\partial y_{i+j}} \in \mathbb{R}^{n \times n}, \quad i = 1, \dots, T-1 \quad j = 1, \dots, \min(T-i, \bar{k})$$

the Jacobian F' of F exhibits the following block-banded structure:

$$F'(y) = \begin{bmatrix} D_1(y) & A_{1,1}(y) & \cdots & \cdots & A_{1,\bar{k}}(y) \\ E_{2,1}(y) & D_2(y) & A_{2,1}(y) & \cdots & \cdots & A_{2,\bar{k}}(y) \\ E_{3,2}(y) & E_{3,1}(y) & D_3(y) & A_{3,1}(y) & \cdots & A_{3,\bar{k}-1}(y) \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ E_{T,\bar{r}}(y) & \cdots & E_{T,1}(y) & D_T(y) & \cdots & \cdots \end{bmatrix}. \quad (6)$$

3 Newton-Krylov methods

In this section we briefly describe Newton-Krylov methods for the solution of nonlinear systems of equations. Given a system of nonlinear equations $F(y) = 0$, where $F : \mathbb{R}^N \mapsto \mathbb{R}^N$ is continuously differentiable, Newton-Krylov methods belong to the class of Inexact Newton methods [11], iterative processes that result in the following scheme, where the superscript k denotes the iteration's number:

Let y^0 be given.

For $k = 0$ until "convergence" do:

Find some $\bar{\eta}^k \in [0, 1)$ and \bar{s}^k that satisfy

$$\|F(y^k) + F'(y^k)\bar{s}^k\| \leq \bar{\eta}^k \|F(y^k)\|, \quad (7)$$

Set $y^{k+1} = y^k + \bar{s}^k$.

Clearly, these methods are variants of Newton's method in which at each iteration the Newton equation

$$F'(y^k)s = -F(y^k), \quad k \geq 0, \quad (8)$$

is solved only approximately.

In a Newton-Krylov framework, a Krylov linear solver is applied to (8) and it is iterate till an Inexact Newton step \bar{s}^k satisfying

$$\|F(y^k) + F'(y^k)\bar{s}^k\| \leq \bar{\eta}_k \|F(y^k)\|,$$

is computed. This approach is widely used whenever large scale nonlinear systems have to be solved.

Local convergence analysis for Inexact Newton methods [11] shows that, under standard conditions on the function F , if y^0 is sufficiently close to a solution y^* of the nonlinear system and the $\bar{\eta}^k$'s are uniformly bounded away from one, then the sequence $\{y^k\}$ converges to y^* . Moreover, the choice of the forcing term $\bar{\eta}^k$ has a strong relevance on the performance of Inexact Newton methods as it is shown in [11]. At this regard, we recall that, if $\{\bar{\eta}^k\} \rightarrow 0$ then $\{y^k\}$ exhibits local q-superlinear convergence to a solution y^* , while if $\{\bar{\eta}^k\} = O(F(y^k))$ the convergence rate is quadratic. In [13] two choices of $\bar{\eta}^k$'s yielding up to quadratic rate of convergence have been proposed.

Let us now focus on Newton-GMRES, namely the Newton-Krylov method where GMRES is used to compute the Inexact Newton step.

GMRES (Generalized Minimal RESidual) is an iterative Krylov method for solving indefinite systems of linear equations [39]. GMRES applied to (8) minimizes the residual norm $\|F(y^k) + F'(y^k)s\|$ over all corrections in the current Krylov subspace. Because iterates are based on this norm minimization property, the method does not break down in exact arithmetic, i.e., a new iterate can always be constructed if the current iterate is not the solution, provided the coefficient matrix is nonsingular.

An attractive feature of GMRES applied to the Newton equation (8) is that it requires only the action of the Jacobian F' on a vector v . For an appropriately chosen scalar ϵ , this action can be approximated by finite differences ([25])

$$F'(y)v \sim \frac{F(y + \epsilon v) - F(y)}{\epsilon}, \quad (9)$$

giving rise to a process that is referred to as “matrix-free”. In other words, a Newton-GMRES method can be implemented without requiring the computation and the storage of the Jacobian F' .

4 Newton-GMRES for model simulation

Here, we focus on the solution of (3) and we describe the approach taken in implementing a Newton-GMRES method specially designed for this class of problems. We assume that the Jacobian is not available and we approximate the blocks of the Jacobian that are needed to build the preconditioner, by finite differences.

More precisely, we describe our choice of the forcing terms sequence $\{\bar{\eta}^k\}$, of the preconditioner and of the employed non monotone globalization strategy. Finally, we give the algorithm summarizing the adopted approaches.

4.1 The forcing terms

Our choice of the forcing terms sequence is strictly related to the nature of macroeconomic models. Indeed, these models generally do not show an high level of nonlinearity and are equipped with a good approximation of the solution. In fact, even in presence of "moderate" shocks, the approximation given by the hystorical series are good enough and classical Newton-methods converge in a very low number of iterations. Then, it is not necessary to solve the Newton equation (8) to high accuracy, as a crude approximation of the Newton step is very often good enough to produce progress towards the solution. This way, pointless iterations of the iterative solver GMRES are avoided. In other words, in this type of applications, a less accurate solution of (8) may be both cheaper and effective enough in reducing the norm of F .

Then, despite a forcing sequence $\{\bar{\eta}^k\}$ converging to zero produces a fast convergence of the Inexact-Newton method to the solution, due to the characteristic of the class of models we are interested in, here we choose a constant forcing sequence with $\bar{\eta}^k = \bar{\eta}^0$ for any k .

4.2 The preconditioner

Effective preconditioning of the Newton systems is the most crucial ingredient for obtaining good performance of Newton-Krylov methods. The term preconditioning refers to transforming the Newton system into another system with more favourable properties for the iterative solver (GMRES in our implementation). A preconditioner for the k -th Newton system is a matrix P^k such that the preconditioned system takes the following form:

$$(P^k)^{-1}F'(y^k)s = -(P^k)^{-1}F(y^k).$$

Generally speaking, if P^k is a good preconditioner for $F'(y^k)$, we expect that the iterative solver converges faster on the preconditioned system than on the original one. When GMRES, or any iterative solver, is applied to the preconditioned system it is not necessary to form the preconditioned matrix $(P^k)^{-1}F'(y^k)$ explicitly; the action of the preconditioned matrix on a vector v is accomplished approximating the vector $w = F'(y^k)v$ by finite-differences and computing $z = (P^k)^{-1}w$ solving a linear system of the form $P^kz = w$ ($(P^k)^{-1}$ is not explicitly known). Algebraic preconditioner gives the preconditioner in a factorized and sparse form so that the solution of such linear systems is not an expensive task [6].

In our implementation of the Newton-GMRES method for forward-looking models, we provide two different preconditioners. Our choice of the preconditioners is strictly connected to the choice of the forcing terms: as we solve the linear systems only to a modest accuracy we can rely on unexpensive preconditioners, which are good enough to provide a step \bar{s}^k satisfying (7) within a reasonable number of GMRES iterations, despite are not able to really capture the ill-conditioning of the Jacobian matrix.

A quite common preconditioner for block-banded matrix as (6) evaluated at y^k , is the so called Jacoby preconditioner, i.e.

$$P^k = \begin{bmatrix} D_1(y^k) & & & \\ & D_2(y^k) & & \\ & & \ddots & \\ & & & D_T(y^k) \end{bmatrix}. \quad (10)$$

Anyway, as observed in [34], blocks $D_j(y^k)$ for $j = 1, \dots, T$ share the same structure. In fact, at the k -th nonlinear iteration, $D_j(y^k) = \frac{\partial F_j(y^k)}{\partial y_j}$ with F_j given in (4) and the only difference among the F_j is the period they are referred to. Therefore, in [17], at each Newton-GMRES iteration, the block diagonal preconditioner $P^k = \text{diag}(D_1(y^k), \dots, D_1(y^k))$ is used, with $D_1(y^k)$ given by the (1,1) block diagonal of $F'(y^k)$. Then, following this strategy, at iteration k of the Newton-GMRES method only the (1,1) block diagonal of $F'(y^k)$ needs to be evaluated. Further, its LU decomposition has to be computed, since at each GMRES iteration a linear system with coefficient matrix P^k has to be solved. Gilli and Pauletto provided interesting numerical results adopting this approach.

However, due to the low level of nonlinearity of these models it is not necessary to recompute the preconditioner at each nonlinear iteration. In our approach, we compute the preconditioner only at the first Newton-GMRES iteration and then we freeze it, that is we use the same preconditioner throughout the whole iterative process. This way, we need only to compute the (1,1) block diagonal of $F'(y^0)$ and its LU decomposition. Summarizing, at each Newton-GMRES iteration, we use the preconditioner

$$P^k \equiv P_D = \text{diag}(D_1(y^0), \dots, D_1(y^0)) \quad k \geq 0 \quad (11)$$

where $D_1(y^0)$ is given by the (1,1) block diagonal of $F'(y^0)$. In the sequel we will refer to this preconditioner with the name P_D (Block-Diagonal Preconditioner).

However, when T becomes large, this preconditioner may become less effective. Then, we rely on the following more expensive preconditioner and at each nonlinear iteration we employ

$$P^k \equiv P_B = \begin{bmatrix} D_1(y^0) & A_{1,1}(y^0) & \cdots & A_{1,\tilde{k}}(y^0) \\ E_{2,1}(y^0) & D_2(y^0) & A_{2,1}(y^0) & \cdots & A'_{2,\tilde{k}}(y^0) \\ E_{3,2}(y^0) & E_{3,1}(y^0) & D_3(y^0) & \cdots & A_{3,\tilde{k}-1}(y^0) \\ & \ddots & \ddots & \ddots & \\ & & E_{T,\tilde{r}}(y^0) & \cdots & D_T(y^0) \end{bmatrix} \quad (12)$$

with $\tilde{k} \leq \bar{k}$ and $\tilde{r} \leq \bar{r}$. In the sequel P_B (Block-Banded Preconditioner) will indicate preconditioner (12).

Clearly, when $\tilde{k} = \bar{k}$ and $\tilde{r} = \bar{r}$ we have $P_B = F'(y^0)$, while when $\tilde{k} = 0$ and $\tilde{r} = 0$, P_B reduces to:

$$P_B = \text{diag}(D_1(y^0), \dots, D_T(y^0)). \quad (13)$$

In any case, we remark that we always freeze the preconditioner, that is the factorization of the preconditioner is computed only at the first nonlinear iteration. The blocks of $F'(y^0)$ required by the preconditioner are approximated by finite differences.

4.3 The globalization strategy

In the approach detailed above the linear systems are solved only to a modest accuracy and as a consequence we rely on a very rough approximation of the Newton step. Then we may need to merge our Inexact-Newton method into a globalization strategy. This allows us to monitor the progress of the method toward a solution of the nonlinear system and to reduce the value of the merit function $\|F\|^2$ even if we do not have at disposal a good step to update the current iterate. Moreover, in some situation, for example when severe shocks are used or when we simulate the model for forecasting, the initial guess provided by the historical data is not good enough and Newton-like methods fail. In these cases, a globalization strategy is fundamental to enlarge the convergence basin of a Newton-like method and to enhance its robustness.

There are two major categories of globalizations: linesearch methods in which step lengths are adjusted (usually shortened) to obtain a satisfactory step and trust region methods in which a step is ideally chosen minimizing a local model of $\|F\|^2$ within a specified trust-region (see [25, 12, 32] for details). A further possibility is to employ homotopy methods. In this approach an easy problem with well known solution is gradually transformed into the original problem $F(y) = 0$, building an homotopy map or varying the parameter values in case of parameter dependent nonlinear systems. Homotopy methods are known to be more robust than Newton-like methods embedded into a linesearch or a trust-region strategy, but they are generally more expensive and become complicated near a bifurcation point, requiring sophisticated approaches [32]. Moreover, we are not aware of implementations of homotopy methods in a matrix-free setting as the Jacobian is generally required to devise efficient and theoretical supported implementations of these strategies. Regarding trust-region methods, dogleg strategy in conjunction with Inexact-Newton methods can be devised [36], but at least an approximation of the Jacobian is required. Then, in our opinion, in this context the linesearch strategy is the best option for the following reasons: easy to implement, matrix-free compatible, strong theoretically supported in conjunction with Inexact Newton methods.

Inexact Newton methods embedded into a linesearch strategy result in the following scheme. At iteration k , the Inexact Newton step \bar{s}^k is computed and the new iterate has the form $y^{k+1} = y^k + \lambda \bar{s}^k$ where $0 < \lambda \leq 1$ is such that

$$\|F(y^{k+1})\|^2 < (1 - \alpha\lambda)\|F(y^k)\|^2 \quad (14)$$

with $\alpha \in (0, 1)$. It is quite important to note, at this stage, that an Inexact Newton step satisfying (7) with $\eta^k < 1$ is ensured to be a descent direction for $\|F(y)\|^2$ at y^k , i.e.

$$(\nabla\|F(y^k)\|^2)^T \bar{s}^k < 0,$$

see [8]. Therefore, the existence of λ satisfying (14) is guaranteed.

Moreover, our numerical experience enlightened that a nonmonotone strategy is enough to enhance robustness of Newton-GMRES methods when applied to forward-looking models. In other words, it is not necessary to impose a monotone decrease of the merit function $\|F\|^2$, but it is enough to obtain a reduction within $q \geq 1$ iterations. Then, the acceptance criterion (14) is relaxed and the nonlinear residual at y^{k+1} is required to satisfy the following condition:

$$\|F(y^{k+1})\|^2 < (1 - \alpha\lambda)M^k \quad (15)$$

where

$$M^k = \max\{\|F(y^k)\|^2, \|F(y^{k-1})\|^2, \dots, \|F(y^{k-q})\|^2\}, \quad (16)$$

$q \geq 0$ is a given integer and the superscript k indicates the iteration number. The choices $q > 0$ and $q = 0$ correspond to a nonmonotone and to a monotone strategy, respectively.

4.4 The algorithm and the Matlab code

Now, we sketch the algorithm that is the core of our code.

NGMS Method

Let y^0 , $\eta \in (0, 1)$, $\alpha \in (0, 1)$, $0 < \theta_m < \theta_{max} < 1$, $q \geq 1$, P , \tilde{r} , \tilde{k} , G , M_{nt} , tol_{nt} be given.

1. If $P=1$

 Compute by finite differences the (1,1) diagonal block of $F'(y^0)$ and perform its *LU* factorization.

 else

 Compute by finite differences the preconditioner (12) and perform its *LU* factorization.

2. For $iter = 1, \dots, M_{nt}$

 2.1 Using GMRES (with preconditioner P_D given in (11) if $P=1$, with preconditioner P_B given in (12) otherwise), compute \bar{s}^k such that

$$\|F(y^k) + F'(y^k)\bar{s}^k\| \leq \eta\|F(y^k)\|.$$

 2.2 Set $\lambda = 1$.

 2.3 If $G=1$ (Perform the linesearch strategy)

 2.3.1 Compute M^k by (16).

 2.3.2 While $\|F(y^k + \lambda\bar{s}^k)\|^2 \geq (1 - \alpha\lambda)M^k$ do:

 Choose $\theta \in [\theta_m, \theta_{max}]$.

 Update $\lambda = \theta\lambda$.

- 2.4 Set $y^{k+1} = y^k + \lambda \bar{s}^k$.
 2.5 If $\|F(y^{k+1})\| < tol_nt$ then EXIT

The previous algorithm is the core of the Matlab code NGMS. We have striven to make NGMS as simple to use as possible. The use of NGMS requires only a minimal description of the problem to be solved: the size of the problem, the initial approximate solution, the stopping tolerances tol_nt , the number of allowed nonlinear iteration M_nt and a user-supplied function for evaluating the non linear function (2). At this regard, it should be underlined that in many macroeconometrics models the variables can be reordered so that the nonlinear systems exhibits a large block of interdependent equations, which is preceded and followed by recursive equations. This way, only the block of interdependent equations is needed to be solved by the NGMS method. Our Matlab code is structured in order to exploit this decomposition whenever it is available.

Finally, the user is required to specify the value of the forcing term (that is taken constant throughout all the iterative process), the choice of the preconditioner and if the globalization strategy is desired.

The parameters P , \tilde{r} and \tilde{k} , in input govern the choice of preconditioner. If P is set to one the preconditioner P_D given in (11) is used. Its factors L and U are computed at the beginning of the iterative process and are then used at each iteration of GMRES to solve the linear systems with coefficient matrix given by the preconditioner P_D . With the choice $P=2$, $\tilde{k} = 0$ and $\tilde{r} = 0$, the preconditioner P_B given by (13) is used. In this case, as well as in the more expensive case, $\tilde{k} > 0$ and/or $\tilde{r} > 0$ the LU factorization of the whole preconditioner is computed. The LU factorizations needed to factorize the preconditioner are computed with the built-in Matlab function `lu` with syntax that allows to apply LU factorization of sparse matrices developed in UMFPACK package [10]. In this case reordering and symbolic analysis that limit fill-in in the LU factorization are carried out.

In Step, 2.1, we use GMRES(m) (GMRES with restart) with $m = 150$, a maximum of 10 restarts and with null initial guess. If GMRES is not able to provide a step \bar{s}_k satisfying (7) within the maximum number of allowed iterations, if

$$\frac{\|F(y^k) + F'(y^k)\bar{s}^k\|}{\|F(y^k)\|} < 1 \quad (17)$$

we go on with the last step computed by GMRES, otherwise the NGMS Matlab code stops with a failure. This choice is motivated by the fact that if (17) is satisfied, then \bar{s}^k is a descent direction for $\|F\|^2$ at y^k . Therefore, if the globalization strategy is activated, \bar{s}^k may be good enough to produce progress towards the solution, even if it does not solve the Newton equation to the prescribed accuracy.

Products of $F'(y^k)$ with vectors that are required by GMRES are approximated using the finite-difference formula given in (9) with

$$\epsilon = \sqrt{\epsilon_m} \|y^k\| / \|v\|,$$

where ϵ_m is the machine precision.

In Step 2.3.1-2.3.2 the nonmonotone linesearch strategy is performed, if it is required by the user. That is, if the parameter G that governs this choice is set to 1. In the linesearch strategy, moving along the direction of the initial Inexact step \bar{s}^k , successively shorter steps of the form $s^k = \lambda \bar{s}^k$ are selected until the sufficient decrease condition (15) is met; this process is commonly referred to as *backtracking*. In our Matlab implementation, the parameter α is set to 10^{-4} , q is set to 6 and the choice of the scalar θ is performed using the three-point parabolic model as described in [25]. If the sufficient decrease condition (15) is not satisfied within 10 backtracks, we go on with the last computed value of λ .

In Step 2.5 the tolerance tol_nt is used to control the level of accuracy and the algorithm is stopped as soon as the stopping criterion

$$\|F(y^{k+1})\| < tol_nt \quad (18)$$

is met. If the above stopping criterion is not satisfied within M_nt iterations the algorithm stops with a failure. In the NGMS Matlab code the process is terminate with an error condition in this case.

As a final remark, we would like to stress that, if the globalization strategy is adopted ($G = 1$), the Inexact-Newton method that is the basement of our approach is globally convergent provided that the nonlinear function F is continuously differentiable in an open set containing the generated sequence. Then, if this condition is satisfied, the convergence does not depend in a critical way on the closeness of the initial guess to the solution. Moreover, eventually the Inexact Newton step \bar{s}^k is taken. In other words, the decrease condition (15) is satisfied with $\lambda = 1$.

5 Numerical results

In this section we report on our numerical experience with the code NGMS.

The tests were conducted on a Intel Xeon (TM) 3.4 Ghz, 1GB RAM using MATLAB 7.0. The machine precision is $\epsilon_m \simeq 2.10^{-16}$.

The initial guess y^0 was those provided by the hystorical data, M_nt was set to 100 and $tol_nt = 10^{-6}$.

Results are reported in tables where the following data are shown: the preconditioner used in the column `P`, the number `it_new` of performed nonlinear iterations, the average number `A_gmit` of performed GMRES iterations, the norm of the function F at the final iterate in column `||F||`, the execution time in seconds in column `time`, the time (in seconds) spent to build the preconditioner and its *LU* factorization in column `time_prec`, the number `N_bt` of performed backtracks in the linesearch strategy. Failures of the method due to a breakdown in the linear algebra phase (GMRES is not able to provide an Inexact Newton step that satisfies (17)) are indicated in the tables by the symbol "-".

T	P	it_{new}	A_{gmit}	$\ F\ $	time
5	P_D	21	3	7.d-7	3.3
10	P_D	27	7	4.d-7	6.1
20	P_D	25	15	5.d-7	11.6
30	P_D	24	22	9.d-7	19.6
50	P_D	24	47	8.d-7	57.9
50	P_B	24	25	5.d-7	83.2
60	P_D	24	70	9.d-7	107.1
60	P_B	24	30	5.d-7	107.3
80	P_D	-	-	-	-
80	P_B	23	52	9.d-7	192.7
100	P_B	-	-	-	-

Table 1: Temporary shock: CA_G is increased by 1 per cent of CA_GDP in $t_1 = 2001$, $\eta = 0.5$.

5.1 Multimod model

In the first sets of experiments we apply NGMS to the MULTIMOD (*MULTI-region econometric MODeL*) model, version Mark III, developed by the International Monetary Fund [30, 31, 28]. This is a forward-looking model with 601 equations, a maximum lag of 3 periods, and a maximum lead of 10 periods. To our knowledge the analytic jacobian of the model is not available.

In order to analyze the computational performance of NGMS, we conducted experiments with different choices of the forcing term η . We used constant forcing terms and we compare the behaviour of our code solving the linear systems to a modest accuracy ($\eta = 0.5$ and $\eta = 0.1$) and to a tight accuracy ($\eta = 10^{-4}$). Moreover, our runs were performed varying the value of T . More precisely, we considered nonlinear systems arising by stacking T periods, with $T = 5, 10, 20, 30, 50, 60, 80, 100$. As a result, the dimension of the problems solved varies from 3005 to 60100. We always activate the globalization strategy if necessary, and employ preconditioner P_D . Moreover, in order to compare the behaviour of the preconditioners, problems corresponding to $T \geq 50$ have been solved using both preconditioners P_D and P_B given by (13), that is P_B with $\tilde{k} = \tilde{r} = 0$.

In tables 1-3 we report the results obtained with different values of η and different values of T . We started the simulation at $t_1 = 2001$ and shocked the variable of Canada's government expenditures by the 1% of Canadian GDP for the first year of simulation. In these simulations the linesearch strategy is never activated.

As we can see from these tables and we can expect from the theory, the number of outer iterations required by the Newton-GMRES method grows as the accuracy in the solution of the nonlinear systems decreases.

On the other hand, the average number of GMRES iterations obviously

T	P	<i>it_new</i>	<i>A_gmit</i>	$\ F\ $	time	time_prec
5	P_D	8	8	3.d-7	3	0.9
10	P_D	8	20	5.d-7	4	0.9
20	P_D	8	41	2.d-7	9	0.9
30	P_D	8	64	1.d-7	17.3	0.9
50	P_D	8	130	1.d-7	53.9	0.9
50	P_B	8	70	1.d-7	76.4	47.3
60	P_D	8	203	1.d-7	95.4	0.9
60	P_B	8	82	1.d-7	96.1	55.2
80	P_D	-	-	-	-	-
80	P_B	8	134	4.d-7	171.8	77.3
100	P_B	-	-	-	-	-

Table 2: Temporary shock: CA_G is increased by 1 per cent of CA_GDP in $t_1 = 2001$, $\eta = 0.1$.

T	P	<i>it_new</i>	<i>A_gmit</i>	$\ F\ $	time
5	P_D	3	31	1.d-7	2.5
10	P_D	3	73	1.d-7	4.5
20	P_D	3	162	1.d-7	12.3
30	P_D	3	234	2.d-7	22.6
50	P_D	3	521	7.d-7	78.5
50	P_B	3	276	6.d-7	93.7
60	P_D	4	719	3.d-9	132.3
60	P_B	4	324	3.d-9	121.3
80	P_D	-	-	-	-
80	P_B	4	753	3.d-9	369.5
100	P_B	-	-	-	-

Table 3: Temporary shock: CA_G is increased by 1 per cent of CA_GDP in $t_1 = 2001$, $\eta = 10^{-4}$.