Economic Modelling 133 (2024) 106670

Contents lists available at ScienceDirect

Economic Modelling

journal homepage: www.journals.elsevier.com/economic-modelling

Solving linear DSGE models with Newton methods

Alexander Meyer-Gohde, Johanna Saecker [*]

Goethe-Universität Frankfurt and Institute for Monetary and Financial Stability (IMFS), Theodor-W.-Adorno-Platz 3, 60629 Frankfurt am Main, Germany

A R T I C L E I N F O

Dataset link: Solving Linear DSGE Models with Newton Methods (Replication) (Original data)

JEL classification:
C61
C63
E17

Keywords:
Numerical accuracy
DSGE
Solution methods

A B S T R A C T

This paper presents and compares Newton-based methods from the applied mathematics literature for solving the matrix quadratic that underlies the recursive solution of linear DSGE models. The methods are compared using nearly 100 different models from the Macroeconomic Model Data Base (MMB) and different parameterizations of the monetary policy rule in the medium-scale New Keynesian model of Smets and Wouters (2007) iteratively. We find that Newton-based methods compare favorably in solving DSGE models, providing higher accuracy as measured by the forward error of the solution at a comparable computation burden. The methods, however, suffer from their inability to guarantee convergence to a particular, e.g. unique stable, solution, but their iterative procedures lend themselves to refining solutions either from different methods or parameterizations.

1. Introduction

The solution of linear DSGE models requires solving a matrix quadratic equation and standard existing methods predominantly rely on a generalized Schur or QZ decomposition (Moler and Stewart, 1973; Golub and van Loan, 2013) for solving this underlying matrix quadratic. While there are a few exceptions,[1] alternative methods from the applied mathematics literature have yet to be systematically studied in a DSGE context. This paper fills part of that gap, collecting Newton-based solution methods for matrix quadratic problems and applying them to the solution of linear DSGE models.[2] Newton methods require an initial guess and we find that for initial guesses close to the resulting solution, perhaps from a nearby parameterization, these methods perform favorably compared with QZ-based methods - a consequence of the asymptotic quadratic convergence of Newton methods. Precisely this iterative characteristic also enables the Newton methods we introduce to linear DSGE models to correct insufficiently accurate solutions of economic consequence as presented in Meyer-Gohde (2023).

One alternative to QZ-based methods are Newton-based algorithms, which although familiar to economists in root-finding settings have not yet been examined for solving linear DSGE models. The only exception we are aware of is Dynare's (Adjemian et al., 2011) (henceforth Dynare) undocumented file `quadratic_matrix_equation_solver.m` that implements Higham and Kim's (2001) Newton method with exact line searches, see Section 3. The applied mathematics literature has further developed and refined numerical methods for solving matrix quadratic methods past generalized Schur or QZ methods based on the classic contribution of Moler and Stewart (1973). Higham and Kim (2001) present a Newton algorithm incorporating exact line searches and show it improves global convergence by making it faster and more reliable. Furthermore, they derive a conditioning number and bound the backward error, as reviewed and applied to a DSGE context in Meyer-Gohde (2023). Long et al. (2008) introduce two new algorithms making Higham and Kim's (2001) line searches occasional, thus reducing the potential computational burden associated with Higham and Kim's (2001) method, with one producing better numerical results.

In this paper, we present six different Newton-based solution algorithms using a unified notation and for the application to solving linear DSGE models as an alternative to QZ-based methods. We engage in a number of experiments to compare the algorithms to QZ-based methods.[3] First we apply the different methods to the models in the Macroeconomic Model Data Base (see Wieland et al., 2012, 2016) (henceforth MMB), comparing the performance to the QZ-based

[*] Corresponding author.
E-mail addresses: meyer-gohde@econ.uni-frankfurt.de (A. Meyer-Gohde), saecker@hof.uni-frankfurt.de (J. Saecker).
[1] Such as the methods of Blanchard and Kahn (1980), Binder and Pesaran (1997) and Anderson (2010) and the cyclic reduction method of Dynare.
[2] For Bernoulli iterative methods, see Meyer-Gohde (2022), Rendahl (2017), and Binder and Pesaran (1997), and Huber et al. (2023) for doubling methods.
[3] We use Dynare's implementation of the QZ method, documented in Villemot (2011), to compare the Newton methods with.

https://doi.org/10.1016/j.econmod.2024.106670
Received 30 June 2023; Received in revised form 31 January 2024; Accepted 31 January 2024
Available online 2 February 2024

method of Dynare both unconditionally (i.e., replacing the QZ method) and then as a refinement (i.e., initializing the Newton methods with the solution generated from QZ). We find that conditional on convergence to the unique stable solution, the different Newton methods perform favorably compared with QZ, providing a solution at the same order of computational cost but at an order of magnitude higher accuracy.[4] That these methods are not guaranteed to converge to a specified solution is a known limitation from the applied mathematics literature (see Higham and Kim (2001)). Initializing the methods at the zero matrix, we find that our baseline method converges to the stable solution for roughly half of the models, while adding line searches increases this to about two-thirds.

The iterative nature of the Newton algorithms is also an advantage, allowing us to explore their ability to refine the solutions provided by the QZ method. Initializing at the QZ solution, the methods provide additional orders of magnitude in accuracy at an addition computational cost that is a fraction of the original QZ cost, with convergence to the unique stable solvent for all of the models in the MMB. This iterative nature also lends itself to iterative parameter experiments or estimations and we compare the Newton algorithms with the QZ method in solving for different parameterizations of the monetary policy rule in the celebrated (Smets and Wouters, 2007) model of the US economy. We fill in a grid with different values of the reaction of the nominal interest rate rule to inflation and real activity; whereas the QZ method starts anew at each parameterization, the Newton methods can use the solution from the previous, nearby parameterization to initialize the algorithm. As the density of the grid increases, we find that all of the Newton methods surpass QZ by roughly an order of magnitude both in terms of computation cost and accuracy as measured by the forward error.

The remainder of the paper is organized as follows. Section 2 lays out the general DSGE model class. In Section 3, we present the set of different Newton-based methods we apply from the applied mathematics literature in a unified notation commensurate with our class of DSGE models. Section 4 examines practical and theoretical considerations such as the choice of initial value, solvability, accuracy and convergence. In Section 5, we compare the different Newton-based methods to the standard QZ method in two applications, one using the MMB of 99 different models and the second over a range of parameterizations within the (Smets and Wouters, 2007) model. Finally, Section 6 concludes.

2. Problem statement

Standard numerical solution packages available to economists and policy makers – e.g., Dynare, Gensys (Sims, 2001), (Perturbation) AIM (Anderson and Moore, 1985; Anderson et al., 2006), Uhlig's Toolkit (Uhlig, 1999) and Solab (Klein, 2000) – all analyze models that in some way or another can be expressed in the form of the nonlinear functional equation

$$0 = E_t[f(y_{t+1}, y_t, y_{t-1}, \varepsilon_t)] \tag{1}$$

The model equations (optimality conditions, resource constraints, market clearing conditions, etc.) are represented by the n_y-dimensional vector-valued function $f : \mathbb{R}^{n_y} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_e} \rightarrow \mathbb{R}^{n_y}$; $y_t \in \mathbb{R}^{n_y}$ is the vector of n_y endogenous variables; and $\varepsilon_t \in \mathbb{R}^{n_e}$ the vector of n_e exogenous shocks with a known distribution, where n_y and n_e are positive integers ($n_y, n_e \in \mathbb{N}$).

The solution to (1) is sought as the unknown function

$$y_t = y(y_{t-1}, \varepsilon_t), \quad y : \mathbb{R}^{n_y+n_e} \rightarrow \mathbb{R}^{n_y} \tag{2}$$

a function in the time domain that maps states, y_{t-1} and ε_t, into endogenous variables, y_t. An analytic form for (2) is rarely available and researchers and practitioners are compelled to find approximative solutions. However, a steady state, $\bar{y} \in \mathbb{R}^{n_y}$ a vector such $\bar{y} = y(\bar{y}, 0)$ and $0 = f(\bar{y}, \bar{y}, \bar{y}, 0)$ can frequently be recovered, either analytically or numerically, providing a point of expansion around which local solutions may be recovered.

A first-order, or linear, approximation of (1) at the steady state delivers,

$$0 = AE_t\left[y_{t+1}\right] + By_t + Cy_{t-1} + D\varepsilon_t \tag{3}$$

where A, B, C, and D are the derivatives of f in (1) with respect to its arguments and, recycling notation, the y's in (3) refer to (log) deviations of the endogenous variables from their steady states, $\bar{y}$.

In analogy to (2), the standard approach to finding a solution to the linearized model (3) is to find a linear solution in the form

$$y_t = P\, y_{t-1} + Q\, \varepsilon_t \tag{4}$$

a recursive solution in the time domain–solutions that posit y_t as a function of its own past, y_{t-1}, and exogenous innovations, ε_t.

Inserting (4) into (3) and taking expectations ($E_t\left[\varepsilon_{t+1}\right] = 0$), yields the restrictions

$$0 = AP^2 + BP + C, \quad 0 = (AP + B)\,Q + D \tag{5}$$

Generally, a unique P with eigenvalues inside the closed unit circle is sought. Lan and Meyer-Gohde (2014) prove the latter can be uniquely solved for Q if such a P can be found. Hence, the hurdle is the former, matrix quadratic equation.

Most linear DSGE methods use a generalized Schur or QZ decomposition (Moler and Stewart, 1973; Golub and van Loan, 2013) of the companion linearization of (3)[5] in some form or another. We will take a different route and instead solve for P in (5) using Newton-based methods to which we turn now.

3. Newton methods for linear DSGE models

This section contains the methods from the applied mathematics literature that we will analyze in the context of solving linear DSGE models as introduced above. We will begin by introducing Newton's method in a univariate context to fix ideas and then proceed to the different methods from the literature suggested for the solution of matrix quadratic equations.

3.1. Newton's method

We will begin by analyzing a univariate equation, see, e.g., Judd (1992, pp. 152–153) or Corless and Fillion (2013, pp. 113–116), to fix ideas and illustrate some of the obstacles faced when using Newton methods to solve quadratic equations.

Consider the root-finding problem $f(x) : \mathbb{C}^1 \rightarrow \mathbb{C}^1$

$$0 = f(x), \quad f(x) : \mathbb{C}^1 \rightarrow \mathbb{C}^1 \tag{6}$$

and form a Taylor expansion of $\tilde{x} \equiv x + \Delta x$ at x

$$0 \approx f(x) + f'(x)(\tilde{x} - x) \tag{7}$$

Using the definition of $\tilde{x}$ and solving for Δx yields

$$\Delta x = -\left(f'(x)\right)^{-1} f(x) \tag{8}$$

[4] Our measure of accuracy is the forward error of Meyer-Gohde (2023).

[5] For a presentation of the QZ decomposition for solving linear DSGE models with the method of undetermined coefficients and a multivariate pivoted (Blanchard, 1979) approach, see Meyer-Gohde (2023).

Starting with some x_0 and iterating through the foregoing produces a solution for $f(x)$ that converges quadratically asymptotically. Convergence may initially be slow and may even fail, for example, if $f'(x) = 0$ for some x.

The problem generated by (5) is a (matrix) quadratic problem. Again to fix ideas, consider its univariate equivalent

$$0 = f(x) = ax^2 + bx + c \tag{9}$$

where we consider (in accordance with our DSGE model), a, b, and $c \in \mathbb{R}^1$. From the above, we need to form $f'(x)$ and solve for Δx. Accordingly,

$$f'(x) = 2ax + b \tag{10}$$

and hence

$$\Delta x = -\frac{ax^2 + bx + c}{2ax + b} \tag{11}$$

Inspection highlights a difficulty with Newton-based methods, namely that $2ax + b \approx 0$ will be likely ill-conditioned and produces arbitrarily large Δx. Furthermore, given convergence of the algorithm, it is not obvious a priori to which of the two roots

$$x_{1,2} = \frac{-b \pm \left(b^2 - 4ac\right)^{1/2}}{2a} \tag{12}$$

the recursion will converge for a given initialization, x_0. While this so-called basin of attraction has been established for scalar quadratic equations, see Corless and Fillion (2013, p. 115) or Schröder (1870), cubic or higher order equations (as an n'th order matrix quadratic would generate in its determinant for example) lead to complicated (chaotic) basins, see Corless and Fillion (2013, p. 115–116) or Cayley (1879). Higham (2002) highlights this hurdle in the solution of matrix quadratic equations, specifically if a particular solution or a solution with particular properties (such as the saddle point stability in DSGE models) is sought.

Turning now to our matrix problem, we will formalize the matrix quadratic equation in (5). For A, B, and $C \in \mathbb{R}^{n_y \times n_y}$, a matrix quadratic $M(P) : \mathbb{C}^{n_y \times n_y} \to \mathbb{C}^{n_y \times n_y}$ is defined as

$$M(P) \equiv A\,P^2 + B\,P + C \tag{13}$$

with its solutions, called solvents, given by $P \in \mathbb{C}^{n_y \times n_y}$ if and only if $M(P) = 0$. The eigenvalues of the solvent, called latent roots of the associated lambda matrix[6] $M(\lambda) : \mathbb{C} \to \mathbb{C}^{n \times n}$ (here of degree two), are given via

$$M(\lambda) \equiv A\,\lambda^2 + B\,\lambda + C \tag{14}$$

The latent roots are (i) values of $\lambda \in \mathbb{C}$ such that $\det M(\lambda) = 0$ and (ii) $n_y - \text{rank}(A)$ infinite roots. The intimately related quadratic eigenvalue problem given via

$$\lambda \in \mathbb{C} : \left(A\lambda^2 + B\lambda + C\right) x = 0 \text{ for some } x \neq 0 \tag{15}$$

has been reviewed extensively by Tisseur and Meerbergen (2001) and Hammarling et al. (2013) provide a comprehensive method to improve the accuracy of its solutions. A result of particular interest for QZ solutions is that of Tisseur (2000), who demonstrates the potential for the QZ method to fail when solving polynomial eigenvalue problems. This is due to the companion linearization (or stacking of polynomial problems to reduce them to generalized eigenvalue problems) which breaks the backward stability of the QZ algorithm — see Meyer-Gohde (2023) for details in the matrix quadratic context applicable to DSGE models.

[6] See, e.g., Dennis et al. (1976, p. 835) or Gantmacher (1959, vol. I, p. 228).

The matrix quadratic (13) can be expanded following Higham and Kim (2001) as

$$\begin{aligned} M(P + \Delta P) &= A\,(P + \Delta P)^2 + B\,(P + \Delta P) + C \quad &(16)\\ &= A\,P^2 + B\,P + C + A\,\Delta P^2 + A\,(P\Delta P + \Delta P\,P) + B\,\Delta P \quad &(17)\\ &= M(P) + (A\,\Delta P\,P + (A\,P + B\,)\,\Delta P) + A\,\Delta P^2 \quad &(18)\\ &= M(P) + \mathscr{D}_P\,(\Delta P) + A\,\Delta P^2 \quad &(19) \end{aligned}$$

where $\mathscr{D}_P\,(\Delta P)$ is the Fréchet derivative of M at P in the direction ΔP.

3.2. Baseline Newton-based methods

Newton's method ignores the second order term in (19) and calculates ΔP to solve

$$M(P + \Delta P) \approx M(P) + \mathscr{D}_P\,(\Delta P) \overset{!}{=} 0 \tag{20}$$

and proceeds iteratively, updating P with $P + \Delta P$ until convergence has been achieved. Hence each step requires the solution of

$$A\,\Delta P\,P + (A\,P + B\,)\,\Delta P = -M(P) \tag{21}$$

for ΔP given a P.[7] This gives the baseline Newton procedure as

Baseline Newton Method

```
• Given $A$, $B$, $C$, an initial $P_0$, and a convergence criterion $\epsilon$
• While criterion$(P_j) > \epsilon$
    (1) Solve for $\Delta P_j$ in
            $A\,\Delta P_j\,P_j + \left(A\,P_j + B\,\right)\Delta P_j = -M(P_j)$      (22)
    (2) Set $P_{j+1} = P_j + \Delta P_j$
    (3) Advance $j = j + 1$
• Return $P_j$
```

This baseline Newton's method requires solving (21) at each step, a generalized Sylvester equation, delivering quadratic convergence at a computational cost of at least $52n^2$ flops (Higham and Kim, 2001).

3.3. Modified Newton's method

Long et al. (2008) (Algorithm 2.2) note that a convergent algorithm can be designed with only partial updating of (21). For each iteration, the algorithm solves

$$A\,\Delta P_j\,P_0 + \left(A\,P_0 + B\,\right)\Delta P_j = -M(P_j) \tag{23}$$

[7] Note that ΔP, and hence $P + \Delta P$, will be real valued if P is real valued as ΔP solves the linear equation (21) with real valued coefficients. This is in contrast to existing implementations of QZ which operates in the complex domain. For real valued coefficient problems like ours, complex eigenvalues occur in conjugate pairs and splitting the eigenspace on an annulus ensures that both will either be included or excluded. Uhlig (1999) argues that the inclusion or exclusion of both eigenvalues of any complex pair should result in a real valued solution and Klein (2000) discusses, but his Matlab program `solab.m` does not implement, a real generalized Schur approach that by construction ensures the solution remains real valued.

for ΔP_j given P_j from the previous iteration and the initial P_0. This gives the following modified Newton's algorithm

Modified Newton Method

```
• Given $A$, $B$, $C$, an initial $P_0$, and a convergence criterion $\epsilon$
• While criterion($P_j$) > $\epsilon$
    (1) Solve for $\Delta P_j$ in
            $A\,\Delta P_j P_0 + \left(A\,P_0 + B\right)\Delta P_j = -M(P_j)$    (24)
    (2) Set $P_{j+1} = P_j + \Delta P_j$
    (3) Advance $j = j + 1$
• Return $P_j$
```

This modified Newton's method again requires solving a generalized Sylvester equation, now (23) at each step, but now with constant coefficients on the left-hand side at each iteration as only P_j in $M(P_j)$ on the right-hand side of (23) is updated. This simplifies the solvability considerations, presents an opportunity to economize on computational costs, but comes at the cost of quadratic convergence (Long et al., 2008).

3.4. Newton's method with Šamanskii technique

Combining the two previous techniques, the Šamanskii algorithm, Algorithm 2.3 of Long et al. (2008), runs a fixed number of interim modified Newton updates in between each baseline Newton step, striking a balance between the potential computational savings of the modified algorithm and the quadratic rate of convergence of the baseline algorithm. This gives us the following algorithm

Šamanskii Technique

```
• Given $A$, $B$, $C$, an initial $P_0$, an integer $m$, and a convergence criterion $\epsilon$
• While criterion($P_j$) > $\epsilon$
• Set $i = 0$ and $P_{j,0} = P_j$
    (1) While $i < m$
        (a) Solve for $\Delta P_{j,i}$ in
                $A\,\Delta P_{j,i} P_j + \left(A\,P_j + B\right)\Delta P_{j,i} = -M(P_{j,i})$    (25)
        (b) Set $P_{j,i+1} = P_{j,i} + \Delta P_{j,i}$
        (c) Advance $i = i + 1$
    (2) Set $P_{j+1} = P_{j,m}$
    (3) Advance $j = j + 1$
• Return $P_j$
```

When $m = 1$, the baseline Newton method is recovered. Long et al. (2008) show that an $m = 2$ – that is, one intermittent modified step – delivers a cubic convergence rate in j at an economical increase in computation cost over the baseline method. It is important to stress, however, that j is the outer index and our comparisons in Section 5 will not support the implication that this method has stronger convergence properties than the other methods we examine.

3.5. Newton-based method with exact line searches

Higham and Kim (2001) lay out a Newton method with exact line searches, which is motivated by the inaccuracies of the linear approximation in (20) that ignores the second order term in (19). If P_j is far from a solvent ($P : M(P) = 0$), the update $P_{j+1} = P_j + \Delta P_j$ might be farther from a solvent than P_j. They propose a line search, a multiple of the Newton step, $P_{j+1} = P_j + t\Delta P_j$ where t is an appropriate scalar. Obviously, if $t = 1$, the baseline Newton algorithm is recovered. They select the multiple of the Newton step by finding a t that minimizes the merit function

$$t = \underset{x\in[0,2]}{\arg\min}\, \|M(P + x\Delta P)\|_F^2 \tag{26}$$

Higham and Kim (2001) show that this particular choice of merit function (including the Frobenius norm) is convenient as

$$g(x) \equiv \|M(P + x\Delta P)\|_F^2 = \gamma x^4 - \beta x^3 + (\alpha + \beta)\,x^2 - 2\alpha x + \alpha \tag{27}$$

$$g'(x) = 2\alpha\,(x - 1) + \beta\left(2x - 3x^2\right) + 4\gamma x^3 \tag{28}$$

where $\alpha = \|M(P)\|_F^2$, $\beta = trace\left(M(P)^* A\,(\Delta P)^2 + \left(A\,(\Delta P)^2\right)^* M(P)\right)$ and $\gamma = \left\|A\,(\Delta P)^2\right\|_F$. As $g(x)$ is a quartic polynomial it has at most two minima and, as $g'(0) < 0$ and $g'(2) \geq 0$, has a zero in the interval $(0, 2]$ corresponding to either a minimum or an inflection point. Implementing t from (26) is straightforward as either there is a single real zero of $g'(x)$ which lies in the $(0, 2]$ interval and is the global minimum of $g(x)$ or $g'(x)$ has three real zeros, of which at most two correspond to minima of $g(x)$. Hence, finding the zeros of $g'(x)$ and comparing the associated values of $g(x)$ with the value of $g(2)$ enables t from (26) to be readily found.

This gives the Newton procedure with exact line searches as

Exact Line Searches

```
• Given $A$, $B$, $C$, an initial $P_0$, and a convergence criterion $\epsilon$
• While criterion($P_j$) > $\epsilon$
    (1) Solve for $\Delta P_j$ in
            $A\,\Delta P_j P_j + \left(A\,P_j + B\right)\Delta P_j = -M(P_j)$    (29)
    (2) Solve for $t_j$ in
            $t_j = \underset{x\in[0,2]}{\mathrm{argmin}} \left\|M(P_j + x\Delta P_j)\right\|_F^2$    (30)
    (3) Set $P_{j+1} = P_j + t_j \Delta P_j$
    (4) Advance $j = j + 1$
• Return $P_j$
```

This method requires solving (21) as in the baseline Newton method and additionally calculating the line-search step. The additional costs are "negligible" at $5n^3$ flops and Higham and Kim (2001) show that the line-search step does not interfere with the quadratic convergence of the baseline Newton method. Hence, at a small additional cost, non-local missteps can be avoided while maintaining the local, fast convergence of the baseline Newton method.

3.6. Newton-based method with occasional exact line searches

Algorithm 3.1 of Long et al. (2008) notes that the line searches in Higham and Kim's (2001) method above are needed only when the linear approximation in (20) that ignores the second order term in (19) is problematic, i.e. when the current P is far from a solvent. Hence, they suggest implementing line searches only when the current iteration is far from a solvent so as to avoid the additional computational burden of these searches when the quadratic convergence rate of the Newton algorithm sets in. This gives the Newton procedure with occasional exact line searches as

Occasional Exact Line Searches

```
• Given $A$, $B$, $C$, an initial $P_0$, and two convergence criteria $\epsilon$ and $\epsilon_0$
• While criterion$(P_j) > \epsilon$
    (1) Solve for $\Delta P_j$ in
          $A\,\Delta P_j P_j + \left(A\,P_j + B\right)\Delta P_j = -M(P_j)$      (31)
    (2) if criterion$(P_j + \Delta P_j) > \epsilon_0$
          (a) Solve for $t_j$ in
                $t_j = \operatorname{argmin}_{x\in[0,2]} \left\| M(P_j + x\Delta P_j) \right\|_F^2$      (32)
          (b) Set $P_{j+1} = P_j + t_j \Delta P_j$
    (3) else
          (a) Set $P_{j+1} = P_j + \Delta P_j$
    (4) Advance $j = j + 1$
• Return $P_j$
```

This method is identical to the line-search method above, except that the line searches are implemented only on a need-be basis. This further reduces the small additional cost of line searches, maintaining the avoidance of non-local missteps of the line-search method and the local, fast convergence of the baseline Newton method.

3.7. Newton-based method with occasional exact line searches and Šamanskii technique

Long et al. (2008, Algorithm 3.2) combines the cubic convergence of the Šamanskii technique above with the line-search approach of Higham and Kim (2001) to avoid non-local missteps of using the Newton algorithm when the current iteration on P is far from a solvent. This Newton procedure with occasional exact line searches and the Šamanskii technique is

Occasional Exact Line Searches and Šamanskii Technique

```
• Given $A$, $B$, $C$, an initial $P_0$, $m$ and two convergence criteria $\epsilon$ and $\epsilon_0$
• While criterion$(P_j) > \epsilon$
    (1) Solve for $\Delta P_j$ in
          $A\,\Delta P_j P_j + \left(A\,P_j + B\right)\Delta P_j = -M(P_j)$      (33)
    (2) if criterion$(P_j + \Delta P_j) > \epsilon_0$
          (a) Solve for $t_j$ in
                $t_j = \operatorname{argmin}_{x\in[0,2]} \left\| M(P_j + x\Delta P_j) \right\|_F^2$      (34)
          (b) Set $P_{j+1} = P_j + t_j \Delta P_j$
    (3) else
          (a) Set $i = 1$ and $P_{j,1} = P_j + \Delta P_j$
                (i) While $i < m$
                      (A) Solve for $\Delta P_{j,i}$ in
                            $A\,\Delta P_{j,i} P_j + \left(A\,P_j + B\right)\Delta P_{j,i}$
                            $= -M(P_{j,i})$      (35)
                      (B) Set $P_{j,i+1} = P_{j,i} + \Delta P_{j,i}$
                      (C) Advance $i = i + 1$
                (ii) Set $P_{j+1} = P_{j,m}$
    (4) Advance $j = j + 1$
• Return $P_j$
```

When $m = 1$, the Newton method with occasional line searches is recovered. Long et al. (2008) show that an $m = 2$ – that is, one intermittent modified step – delivers a cubic convergence rate in j at an economical increase in computation cost over the baseline method.

4. Theoretical and practical considerations

4.1. Initial value

All Newton methods need an initial value, P_0. In contrast to the scalar quadratic equation whose "basins of attraction" for initial values are known – see Section 3.1 – there is minimal guidance for the choice of an initial value for the matrix quadratic equation. Indeed this constitutes one of the remaining open problems noted by Higham and Kim (2001).

As our goal is to obtain the minimal solvent P – the solvent with the smallest in absolute value eigenvalues – we choose the initial value $P_0 = 0$. In the absence of any other guidance, this choice satisfies the requirement of having all eigenvalues inside the unit circle. In our experiments, we check whether the solvent produced by the methods of the previous section with the initial value is the minimal solvent.

In an iterative analysis, say using an MCMC Bayesian estimation procedure (An and Schorfheide, 2007) or a parameter robustness exercise, the solvent P from the previous parameterization might be used to initialize the new Newton procedure to solve for the solvent with the current parameter draw. This can be formalized as follows. Given a solvent from a previous parameter draw, P such that $M(P) = 0$, update with information about the change in the matrix quadratic at the current parameter draw using an analogous expansion to the matrix quadratic $\tilde{M} = M + \Delta M$ as in (13)

$$\tilde{M}(\tilde{P}) \equiv (M + \Delta M)(P + \Delta P) \tag{36}$$
$$= \tilde{A}\,\tilde{P}^2 + \tilde{B}\,\tilde{P} + \tilde{C} \tag{37}$$
$$= (A + \Delta A)(P + \Delta P)^2 + (B + \Delta B)(P + \Delta P) + C + \Delta C \tag{38}$$

where P is a solvent of M, $M(P) = AP^2 + BP + C = 0$, and ΔA, ΔB, and ΔC are perturbations in the parameters of the matrix quadratic (i.e., the changes in the coefficient matrices resulting from the change in the parameter vector in an MCMC procedure). Developing this further

$$\tilde{M}(\tilde{P}) = (A + \Delta A)(P^2 + \Delta P P + P\Delta P + \Delta P^2)$$
$$+ (B + \Delta B)(P + \Delta P) + C + \Delta C \tag{39}$$
$$= M(P) + \Delta A\, P^2 + \Delta B\, P + \Delta C + (A + \Delta A)(\Delta P P + P\Delta P + \Delta P^2)$$
$$+ (B + \Delta B)\,\Delta P \tag{40}$$
$$= \Delta M(P) + (A + \Delta A)(\Delta P P + P\Delta P + \Delta P^2) + (B + \Delta B)\,\Delta P \tag{41}$$
$$= \Delta M(P) + \tilde{A}\,\Delta P P + (\tilde{A}P + \tilde{B})\,\Delta P + \tilde{A}\,\Delta P^2 \tag{42}$$
$$= \Delta M(P) + \tilde{\mathscr{D}}_P(\Delta P) + \tilde{A}\,\Delta P^2 \tag{43}$$

where the third line follows as $M(P) = 0$ was assumed. $\tilde{\mathscr{D}}_P(\Delta P)$ is the Fréchet derivative of $\tilde{M}$ at P in the direction ΔP.

Analogously to Newton's method in the previous section, we ignore the second order term ΔP^2 in (43) and calculate ΔP to solve

$$\tilde{M}(\tilde{P}) = \Delta M(P) + \tilde{\mathscr{D}}_P(\Delta P) = 0 \tag{44}$$

or

$$\tilde{A}\,\Delta P P + (\tilde{A}P + \tilde{B})\,\Delta P = -\Delta M(P) \tag{45}$$

for ΔP given a P such that $M(P) = 0$. This would be identical with (21), apart from the notation to indicate a change in the coefficient matrices of the matrix quadratic $\tilde{A}$ instead of A, etc., the left-hand side would read $-\tilde{M}(P)$ instead of $-\Delta M(P)$. But as $M(P) = 0$ and $\tilde{M}(P) = M(P) + \Delta M(P)$, the two are identical. Thus, if a solvent P from a nearby problem $M(P)$ is available, $M(P) = 0$, then the chosen Newton procedure from the previous section can be initialized with $\tilde{P}_0 = P$.

4.2. Solvability

All of the methods in the previous section involve solving a generalized Sylvester equation of the form

$$A\,X\,P_j + \left(A\,P_j + B\right)X + M(P_j) = 0 \tag{46}$$

The necessary and sufficient conditions for the solvability of such Sylvester equations are given by Theorem 1 of Chu (1987) which requires the two matrix pencils formed by the leading and trailing matrix coefficients of a generalized Sylvester equation to be regular and have disjoint spectra. Adapted here in the following

Proposition 1. *There exists a unique solution, $X \in \mathbb{R}^{m\times n}$, for the Sylvester equation*

$$AXB + CXD + E = 0$$

where $A, C \in \mathbb{R}^{m\times m}$ and $D, B \in \mathbb{R}^{n\times n}$, if and only if

(1) $P_{AC}(z) \equiv Az + C$ and $P_{DB}(z) \equiv Dz - B$ are regular matrix pencils, and
(2) $\rho(P_{AC}) \cap \rho(P_{DB}) = \emptyset$

where $P_{AC}(z) = Az + C$ (equivalently for $P_{DB}(z)$) is called regular if there exists a $z \in \mathbb{C}$ such that $\det(Az + C) \neq 0$ and the spectrum of the regular pencil $P_{AC}(z)$ is the finite set defined via $\rho(P_{AC}) = \{z \in \mathbb{C} : \det P_{AC}(z) = 0\}$, extended to include infinite eigenvalues, the multiplicity of which is given by m less the rank of A (equivalently n less the rank of D).

Proof. See Chu (1987). Notice the rearrangement and redefinition of terms. □

Hence, the existence of a unique solution X for $A\,X\,P_j + \left(A\,P_j + B\right)X + M(P_j) = 0$ requires

(1) the existence of a $z \in \mathbb{C}$ such that $\det\left(Az + \left(A\,P_j + B\right)\right) \neq 0$
(2) the existence of a $z \in \mathbb{C}$ such that $\det\left(Iz - P_j\right) \neq 0$
(3) $\left\{z \in \mathbb{C} : \det\left(Az + \left(A\,P_j + B\right)\right) = 0\right\} \cap \left\{z \in \mathbb{C} : \det\left(Iz - P_j\right) = 0\right\} = \emptyset$

From Lemma 4.3 and Proposition 4.4 of Lan and Meyer-Gohde (2014), these conditions are fulfilled at $P_j = P$ if P is the unique, stable solvent of $M(P)$, which is equivalent to the nonsingularity of the Fréchet derivative of M at P, $\tilde{\mathscr{D}}_P$, in Lemma 3.1 of Higham and Kim (2001) at a minimal solvent. For our initial value $P_0 = 0$, a unique solution for X of $BX + M(P_0) = 0$ requires B to be of full rank.

Thus like Klein (2000) noted for the QZ approach, solving a Sylvester equation is "always good enough when there is a unique solution to the system", which is the usual case in the literature — although here it is important to note that this only holds with certainty *at* the unique, stable solvent *should* it exist. If there are multiple solutions with respect to some annulus (the unit circle being the standard assumption, see Sims (2001), Lan and Meyer-Gohde (2014), and Al-Sadoon (2018) for methods that construct solutions with eigenvalues outside the unit circle), then the fulfillment of the necessary assumptions for the solvability of the Newton increment P_j at the particular solvent cannot be guaranteed as the spectra of the (say minimum) solvent P and the remainder of the latent roots may coincide. In such a case, a Newton algorithm is unlikely to be able to converge to the solvent even in its vicinity — that is, the Fréchet derivative of M at P in the direction ΔP, $\mathscr{D}_P(\Delta P)$, is singular. This discontinuity is consistent with the critiques of Al-Sadoon (2023), who provides a detailed analysis of the solution space beyond the unique, stable (with respect to some annulus) assumption we tacitly made in positing the solution forms (2) and (4). Hence, the solution methods here are subject to the same critiques in this regard as more familiar QZ based methods and add no specific insights in the case of indeterminacy.

4.3. Convergence and accuracy

Higham and Kim (2001) note that for a P_j sufficiently close to P, standard convergence results for Newton's method apply and, if P is the unique, stable solvent of $M(P)$, the iteration converges and does so at a quadratic rate. Convergence of a sequence of P_j is determined by a stopping criterion. Long et al. (2008) use the residual $\left\|M(P_j)\right\|_F < \epsilon$, where $\|\;\|_F$ indicates the Frobenius norm and ϵ is a small number, say, machine precision (using Matlab 2022a and double precision, $\epsilon = 2^{-52} = 2.2204e{-}16$). Higham and Kim (2001) use the relative residual $\left\|M(P_j)\right\|_F / \left(\|A\|_F \left\|P_j^2\right\|_F + \|B\|_F \left\|P_j\right\|_F + \|C\|_F\right) < n_y\epsilon$.

To assess the accuracy of a computed solution $\hat{P}$ numerically, we apply the practical forward error bounds of Meyer-Gohde (2023),

$$\underbrace{\frac{\left\|P - \hat{P}\right\|_F}{\|P\|_F}}_{\text{Forward Error}} \leq \underbrace{\frac{\left\|H_{\hat{P}}^{-1}\mathrm{vec}\left(R_{\hat{P}}\right)\right\|_2}{\left\|\hat{P}\right\|_F}}_{\text{Forward Error Bound 1}} \leq \underbrace{\left\|H_{\hat{P}}^{-1}\right\|_2 \frac{\left\|R_{\hat{P}}\right\|_F}{\left\|\hat{P}\right\|_F}}_{\text{Forward Error Bound 2}} \tag{47}$$

where $R_{\hat{P}} = A\hat{P}^2 + B\hat{P} + C$ is the residual of the matrix quadratic and $H_{\hat{P}} = I_{n_y} \otimes \left(A\hat{P} + B\right) + \hat{P}' \otimes A$. Stewart's (1971) separation function, see also Kågström (1994), Kågström and Poromaa (1996), and Chen and Lv (2018), is

$$\mathrm{sep}\left[\left(A, A\hat{P} + B\right), \left(I, -\hat{P}\right)\right] = \min_{\|X\|_F = 1} \left\|AX\hat{P} + \left(A\hat{P} + B\right)X\right\|_F \tag{48}$$

$$= \min_{\|\text{vec}(X)\|_2=1} \left\| H_{\hat{P}} \text{vec}\left(X\right) \right\|_2 \tag{49}$$

$$= \sigma_{\min}\left(H_{\hat{P}}\right) \leq \min\left| \lambda\left(A, A\hat{P}+B\right) - \lambda\left(\hat{P}\right)\right| \tag{50}$$

where $\lambda\left(A, A\hat{P}+B\right)$ is the spectrum or set of (generalized) eigenvalues of the pencil $\left(A, A\hat{P}+B\right)$ (and, accordingly, $\lambda\left(\hat{P}\right)$ the set of eigenvalues of $\hat{P}$) and the last line holds with equality for $A = I$ and $\hat{P}$ and $\hat{P}+B$ regular – hence, the separation between the two pencils – the smallest singular value of $H_{\hat{P}}$ - is generically smaller than the minimal separation between their spectra. Analogously to the generalized Sylvester and algebraic Riccati equations, the separation function provides the natural extension of the conditioning number from standard linear equations to these structured problems, and the a posteriori condition number for the matrix quadratic is given by $\text{sep}^{-1}\left[\left(A, A\hat{P}+B\right), \left(I, -\hat{P}\right)\right] = \left\| H_{\hat{P}}^{-1} \right\|_2 = \sigma_{\min}\left(H_{\hat{P}}\right)^{-1}$, which – from above – can be arbitrarily larger than the inverse of the minimal distance between the spectra of the pencils $\left(A, A\hat{P}+B\right), \left(I, -\hat{P}\right)$. Thus a measure based on this separation provides an analogy to other familiar methods as the inverse of the separation relates an upper bound to the forward error directly to the residual, like the condition number for a standard linear system. A tighter bound takes into account the structure more carefully and considers the linear operator $H_{\hat{P}}$ and the residual $R_{\hat{P}}$ jointly — as the linear system $H_{\hat{P}}^{-1}\text{vec}\left(R_{\hat{P}}\right)$ must be solved this bound is computationally more intensive, especially for larger models, as $H_{\hat{P}}$ increases in the square of the dimension of the underlying problem ($n_y^2 \times n_y^2$). This gives us two measures of the relative error of a numerically computed solution to an exact solution, the upper bound is easier to compute and directly analogous to familiar measures for linear equations and the lower bound is tighter, and hence more informative, but prohibitively expensive to calculate for larger models. We favor this measure over the den Haan and Marcet (1994) statistic as the latter is known to frequently not be sensitive enough to discern among nonlinear methods, see Aruoba et al. (2006) and also Juillard and Villemot (2011).[8]

5. Applications

We conduct a number of experiments to assess the performance of the algorithms presented above. When choosing the models to run the experiments we take two considerations into account. On the one hand we would like to assess the methods in a model that is policy-relevant and on the other hand we would like to protect against the danger that our conclusions are model specific. To this end we first compare the algorithms on the Smets and Wouters (2007) model, an estimated, medium-scale New Keynesian model that has found considerable application in policy analysis. We then turn to the Macroeconomic Model Data Base (MMB), a model comparison initiative at the Institute for Monetary and Financial Stability (IMFS),[9] that allows us to examine our algorithms across a wide selection of different models. The benchmark for our algorithms is Dynare's QZ-based method.[10] and we compare the convergence of our different methods to the stable solvent,[11] the accuracy of the solvents, and the associated computation times relative to Dynare.

5.1. Model of Smets and Wouters (2007)

We begin with the model of Smets and Wouters (2007) upon which we place specific attention, particularly on the monetary policy rule in the second set of experiments. In their pivotal work, Smets and Wouters (2007) analyze and estimate a DSGE model based on macroeconomic data from the US economy, providing a compact medium scale model that is the benchmark for structural policy analyses. They build a New Keynesian model featuring sticky prices and wages, inflation indexation, consumption habit formation as well as production frictions concerning investment, capital and fixed costs. The model includes the following log-linearized monetary policy rule,

$$r_t = \rho r_{t-1} + (1-\rho)(r_\pi \pi_t + r_Y(y_t - y_t^p)) + r_{\Delta y}((y_t - y_t^p) - (y_{t-1} - y_{t-1}^p)) + \varepsilon_t^r, \tag{51}$$

which prescribes that the policy authority sets the interest rate r_t reacting to inflation π_t, the current output gap $(y_t - y_t^p)$ and the change in the output gap, and where the parameters r_π, r_Y and $r_{\Delta y}$ describe the strength of each of these reactions. Additionally, ρ controls the degree of interest rate smoothing and ε_t^r is the monetary policy shock following an AR(1)-process with iid normally distributed error. In the paper, the authors employ seven macroeconomic time series from the US economy to estimate the model parameters using Bayesian estimation. They show that the model matches the US macroeconomic data very closely and that out-of-sample forecasting performance is as good as the one of VAR and BVAR models.

We assess the Newton algorithms first as solution refinement methods by initializing them with the output from Dynare's QZ algorithm and then with an uninformed initialization of the stable solvent (the zero matrix). We begin by examining the potential computational savings of the different Newton-based methods relative to standard QZ-based methods when exploring the parameter space of a model, that of the monetary policy rule in Smets and Wouters (2007), where the solvent from the previous parameterization is used as the initial value for the solvent in the new, and likely nearby, parameterization. In particular, we successively narrow the spacing of the parameterization to make the notion of "nearby" concrete. We then turn to the performance at the posterior mode of Smets and Wouters (2007) and finally as refinements to QZ at a numerically problematic parameterization inside the prior from Meyer-Gohde (2023).

We begin by using the Newton-based algorithms to solve the model of Smets and Wouters (2007) iteratively for different parameterizations of the Taylor rule. The goal here is to explore whether solutions from previous, nearby parameterizations can be used to efficiently initialize the Newton methods similarly to the experiment above with the QZ solution as the initial guess. For the parameters determining the Taylor rule reaction to inflation and the long run reaction to the output gap we iterate through a grid of 10 × 10 parameter values varying the size of the intervals considered. We span a grid over the intervals of $r_\pi \in [1.5, 1.5 \ (1 + 10^{-x})]$ and $r_Y \in [0.125, 0.125 \ (1 + 10^{-x})]$, where $x \in [-1, 8]$ (Smets and Wouters (2007) calibrate them to $r_\pi = 2.0443$ and $r_Y = 0.0882$). The algorithm iterates through the two-dimensional grid, updating the initialization to the solution of the previous iteration. A decrease in the spacing between the 100 grid points thus increases the precision of the starting guess.

Table 1 shows that all algorithms are at the median more precise than Dynare, as measured by both upper bounds on the forward error, and roughly by an order of magnitude. A decrease in the spacing between the grid points decreases run time per grid point for all of the algorithms relative to Dynare's QZ method — which does not benefit from having a nearby solution to initialize its algorithm. The Baseline

[8] See the appendix for the den Haan and Marcet (1994) statistics for the Smets and Wouters (2007) we examine in the next section.

[9] See http://www.macromodelbase.com.

[10] See Villemot (2011) Additionally, note that we follow Dynare and reduce the dimensionality of the problem by grouping variables and structuring the matrix quadratic according to the classification of "static", "purely forward", "purely backward looking", and "mixed" variables. The details are in the online appendix and are of independent interest as they supplement (Villemot, 2011) by providing a detailed block-matrix derivation of the procedure.

[11] We use `dlyap` from Matlab to solve the Sylvester equations. We also explored `gensylv` from Dynare, ACM Algorithm 705 from Gardiner et al. (1992a,b), and Hopkins (2002), ZTGSYL from LAPACK from Kågström (1994) and Kågström and Poromaa (1996), SB04QD from Slicot from Benner et al. (1999), and `gsylv` from MEPACK from Köhler (2021, 2022) to solve the generalized Sylvester equations and include these in our software implementation.

Table 1
Results: Smets-Wouters model.

Method	Grid end	Run time			Forward error 1			Forward error 2		
	(x for $(1+10^{-x})$)	Median	Min	Max	Median	Min	Max	Median	Min	Max
Dynare (QZ)	−1, 6	1	1	1	1	1	1	1	1	1
Baseline Newton Method	−1	0.9	0.76	1.2	0.16	0.011	1.2	0.059	0.021	0.18
	6	0.23	0.21	0.96	0.17	0.028	1.1	0.11	0.044	0.21
Modified Newton Method	−1	2	1.6	5.8	0.13	0.025	0.8	0.059	0.021	0.17
	6	0.23	0.21	2.4	0.15	0.015	0.82	0.1	0.04	0.21
With Šamanskii Technique	−1	1.1	0.75	1.4	0.13	0.011	0.88	0.055	0.015	0.13
	6	0.36	0.33	1	0.15	0.024	1.2	0.094	0.028	0.18
With Line Searches	−1	1	0.87	1.8	0.11	0.01	0.95	0.06	0.02	0.15
	6	0.25	0.23	1	0.19	0.021	1.2	0.11	0.037	0.21
With Occ. Line Searches	−1	1	0.85	2.1	0.16	0.008	1.2	0.062	0.022	0.15
	6	0.24	0.22	1.3	0.16	0.028	1	0.11	0.048	0.22
With Occ. LS & ŠT	−1	1.2	0.98	2	0.13	0.0076	0.87	0.057	0.022	0.14
	6	0.37	0.34	1.4	0.16	0.024	1	0.097	0.04	0.19

• For Dynare, refer to Adjemian et al. (2011).
• Run time per grid point and forward errors relative to Dynare.
• Forward error 1 and 2 are the upper bounds for the true forward error, see (47).

Table 2
Results: Model of Smets and Wouters (2007).

Method	Run time	Max Abs. Diff.	Forward errors		Iterations
			Bound 1	Bound 2	
Dynare (QZ)	1	0	5.5e−14	2.4e−11	1
Baseline Newton Method	1.2	110	9.3e−14	2e−09	10
Modified Newton Method	37	1.5e−12	2.3e−15	4.7e−12	650
With Šamanskii Technique	1.2	110	1.2e−13	5.5e−10	7
With Line Searches	1.9	7.9e−13	6.3e−15	3.6e−12	18
With Occ. Line Searches	2.1	7.8e−13	7.8e−15	3.3e−12	19
With Occ. LS & ŠT	2.1	7.8e−13	7.7e−15	3.8e−12	18

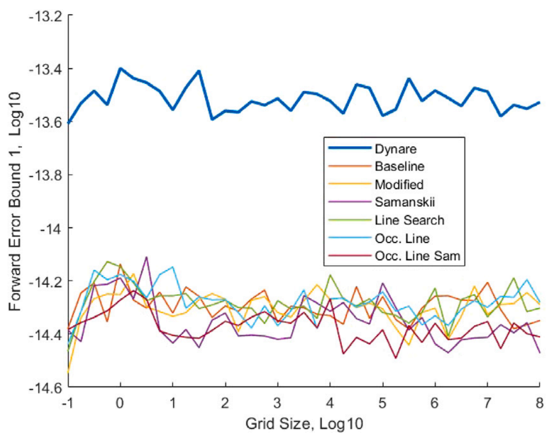
• For Dynare, refer to Adjemian et al. (2011).
• Run Time relative to Dynare (Dynare run time: 0.0021 s).
• Max Abs. Diff. measures the largest absolute difference in the computed P of each method from the P produced by Dynare.
• Forward error 1 and 2 are the upper bounds for the true forward error, see (47).

Newton method displays superior run-time performance at even the widest spacing ($x = -1$ for the end point $(1 + 10^{-x})$ times the original value) and all algorithms are faster than QZ by nearly an order of magnitude at the narrower spacing ($x = 6$). In the wider spacing, the Modified Newton algorithm is, as before, the slowest and least accurate of all algorithms. This disadvantage vanishes with a narrowing of the spacing, for which speed and accuracy become very similar for all algorithms.

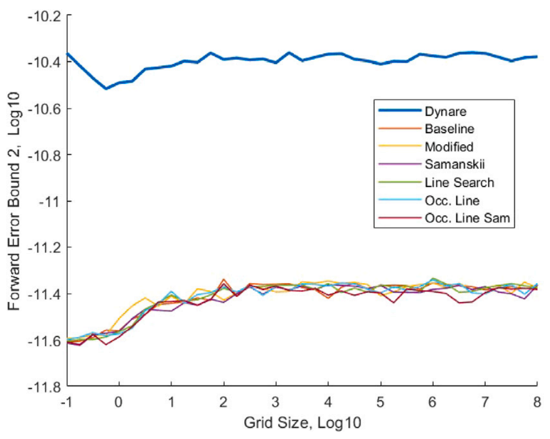
Fig. 1 summarizes the experiment graphically. Fig. 1(c) confirms a decrease in run time per grid point with a narrower grid for the Newton-based algorithms and an irrelevance of the grid spacing for QZ. As the grid becomes narrower, the iterative Newton procedures increasingly benefit from starting from the solution of the previous iteration as it becomes closer to the unknown solution of the current iteration. The QZ algorithm does not operate iteratively and, hence, demonstrates no such benefit, solving for each grid point anew. This relationship is most notable for the modified algorithm which possesses only linear convergence, thus benefiting mostly from a good starting guess. According to Figs. 1(a), 1(b), overall, all algorithms are more precise than Dynare. Thus, in summary, for iterative experiments like the one we have performed here or for the refinement of an inaccurate solution from an alternative algorithm, the iterative nature of Newton-based algorithms can be particularly advantageous, providing more accurate solutions at considerable computational savings.

We now turn to the performance at specific parameterizations and begin with the posterior mode parameterization of Smets and Wouters (2007). Here we would like to assess the promise of the different Newton methods as alternatives to the QZ method and initialize the former at the zero matrix and Table 2 presents the results. Both the Baseline Newton algorithm and the method with the Šamanskii Technique failed to converge to the same solvent as Dynare. That is, they converged to solvents with some eigenvalues outside the unit circle — as pointed out by Higham and Kim (2001) and elaborated on above, there is no known mapping of initializations to solvents to guarantee convergence to a particular solvent and our initialization with the zero matrix (obviously a guess with all eigenvalues inside the unit circle) does not guarantee that the final solvent will have the desired stability properties. The forward errors, however, confirm that they did indeed converge to a solvent (note that the numerator in the upper bound for the forward error is the norm of the residual, so a small forward error confirms that the delivered solvent is not only near to a solvent but also solves the matrix quadratic equation with a small residual, see Meyer-Gohde (2023)). The remaining methods did converge to the same stable solvent as Dynare, however with significantly larger computational costs (as measured in relative run time) compared with Dynare. This is certainly not unexpected for the Modified Newton method, which uses static coefficients in each iteration and thus only displays linear convergence. The line search methods performed more favorably, requiring about an order of magnitude more computing time, but providing roughly an order of magnitude more accuracy (as measured by forward errors).

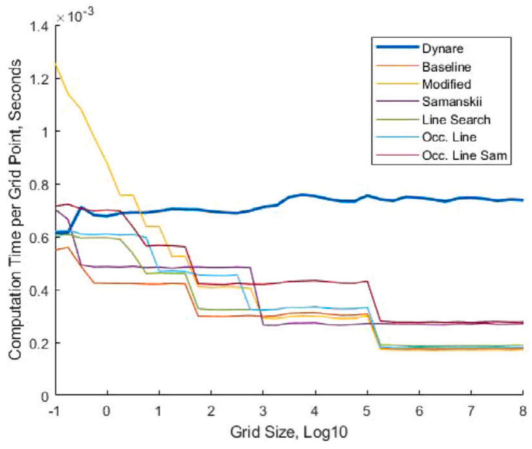
We now return to the promise of Newton methods as solution refinement techniques be examining their potential at an economically relevant numerical instability of the QZ-based solution of Dynare for the model of Smets and Wouters (2007) as examined by Meyer-Gohde (2023). Here we initialize the different Newton methods at the QZ solution and the results are in Table 3. The second column now displays



(A) Forward Error 1



(B) Forward Error 2



(C) Computation Time per Grid Point

Fig. 1. Forward Errors and Computation Time per Grid Point for different parameterizations of the model by Smets and Wouters (2007). Figs. 1(a), 1(b) plot the upper forward error bounds 1 and 2 against the grid size, log10 scale on both axes. Fig. 1(c) plots the computation per grid point against the number of grid points, log10 scale on both axes.

Table 3
Results: Model of Smets and Wouters (2007), Numerically problematic parameterization.

Method	Run time	Variance π_t	Forward errors		Iterations
			Bound 1	Bound 2	
Dynare (QZ)	1	0.28	1e−11	4.6	1
Baseline Newton Method	15	0.32	1.3e−14	0.00033	3
Modified Newton Method	4.2	0.42	1.2e−14	0.00042	3
With Šamanskii Technique	4.8	0.31	1.2e−14	0.00089	5
With Line Searches	7.8	0.26	1.3e−14	0.00033	3
With Occ. Line Searches	7	0.26	1.3e−14	0.00033	3
With Occ. LS & ŠT	3.1	0.26	1.3e−14	0.00033	3

- For Dynare, refer to Adjemian et al. (2011).
- Run Time relative to Dynare (Dynare run time: 0.0022 s).
- The Newton methods were initialized at the QZ solution.
- Forward error 1 and 2 are the upper bounds for the true forward error, see (47).

the variance of inflation as predicted by the solution.[12] At this parameterization, the QZ-based solution predicts an inflation variance of 0.28. However, even the lower of the two upper bounds on the forward error is multiple orders of magnitude above machine precision, indicating a potential numerical instability. Initializing the various Newton methods at the solution produced by Dynare, we find that all of the methods refine the solution, running 3 to 4 iterations and reducing the forward error bounds by 3 to 4 orders of magnitude. All of the methods are considerably faster than the baseline method and predict a variance of inflation of about 1.5 times greater than Dynare.

5.2. MMB suite comparison

Our focus now turns away from a specific model and we try to draw general, non model specific conclusions. For this task we adapt the Macroeconomic Model Data Base (MMB), a model comparison initiative

[12] Smets and Wouters (2007) report a variance of inflation of 0.62 for the whole sample and 0.55 and 0.25 for two subsamples.

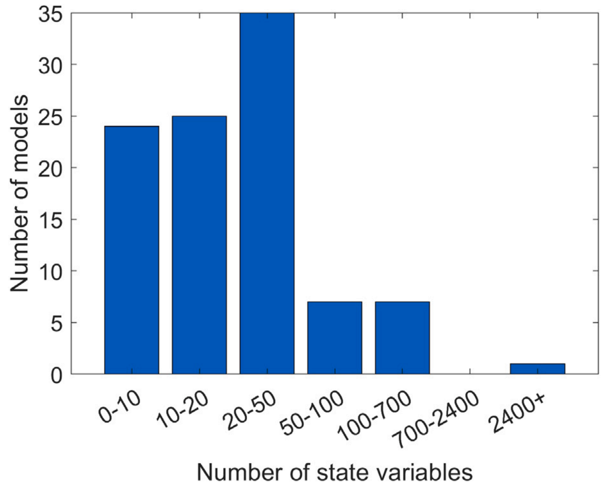


Fig. 2. Histogram over the number of state variables for the 99 MMB models. Fig. 2 plots the number of model state variables over the amount of MMB models. Currently the total amount of models considered is 99.

at the Institute for Monetary and Financial Stability (IMFS)[13] traditionally used to compare the predicted outcomes of different policies across a broad set of macroeconomic models. Version 3.1 contains 151 different models, ranging from small scale, pedagogical models to large scale, estimated models of the US, EU, and multi-country economies.[14] While certainly invaluable for exploring the possible outcomes of policy interventions, we see this database additionally as a useful tool for assessing the potential of different solution methods in a more model-robust context than is currently done in the DSGE literature. Accordingly, we apply the methods of this paper to the set of models appropriate for reproduction,[15] the varying sizes of which – measured by the number of state variables or "purely backward looking" and "mixed" variables in Dynare nomenclature – are summarized in Fig. 2.

We now examine the models of the MMB and assess the various Newton-based methods relative to the QZ method. As above, we begin by demonstrating the potential of the different algorithms as solution refinements, initializing at the QZ solution, and then present the more mixed results of the methods as replacements for QZ by initializing at the zero matrix. Each of the models is taken as is — we maintain the parameterization recorded in the reproduction database of the MMB which is either the original calibration or the estimate(d posterior mode) of the original study. We solve each of the applicable models in the MMB 100 times using the different Newton methods and the QZ method of Dynare for comparison, taking the results as the average within the middle three quintiles to reduce the effects of outliers in measuring the computation time.[16]

Table 4 summarizes the results initiating at the solution provided by QZ.[17] The first column of results counts the number of models for which the method in question converged to the unique stable solution, highlighting that all algorithms operate in the vicinity of the unique stable solution provided by QZ for all models. With this precise QZ solution as a starting guess, all algorithms need only one iteration to satisfy the convergence criterion and none of the algorithms diverge to a different solvent. All algorithms perform this additional iteration at a fraction of the computational cost of the original solution provided by Dynare. Roughly one order of magnitude of additional accuracy is provided by all algorithms as measured by the two forward error upper bounds. Again, the performance of line search and Šamanskii methods in terms of run time and accuracy are disappointing for the set of DSGE models in the MMB in the context of the results of Higham and Kim (2001) and Long et al. (2008), as they do not perform systematically better than the Baseline Newton algorithm along these dimensions. In sum, this experiment provides further strong evidence that Newton-based methods can be used at minimal additional cost to refine the solutions provided by QZ.

Fig. 3 provides an overview of the entire distribution of forward errors, the upper row relative to those from Dynare's QZ method and the lower in absolute terms, using the different Newton-based methods presented here when initialized at Dynare's QZ solution. Forward errors left of the vertical line are thus smaller than Dynare for both figures in the upper row. For both the first, Fig. 3(a), and second, Fig. 3(b), upper bounds on the forward error, we see an obvious shift to the left on a log scale of about one order of magnitude for all the Newton methods and, from the lower row, we see that this entails tightening the distributions as well as shifting them closer to machine precision - a lower convergence criterion would allow additional Newton steps and bring yet more solutions below machine precision. The various Newton methods demonstrate no considerable differences when initializing with Dynare's QZ solution, underscoring that it is the Newton step that provides the additional accuracy.

Table 5 summarizes the results when initiating at the zero matrix. The first column of results counts the number of models for which the method in question converged to the unique stable solution, highlighting a well-known (Higham and Kim, 2001) drawback of Newton methods, namely the unpredictability of which solution the algorithm will converge to. The convergence to the unique stable solvent ranges from 43 models for the Newton method with Šamanskii Technique to 67 models for all of the line-search methods. For the remaining models, the algorithms generally converged to a solvent as the forward errors are roughly the same magnitude as those of Dynare's QZ (this can be seen by examining the maximal relative forward errors, which for all algorithms but the Modified and Occasional Line searches are at worst about one order of magnitude higher than Dynare's QZ), however, just not to the stable one. Hence, even initializing the algorithms at the zero matrix (arguably the appropriate uninformed prior for a stable solution) manages at best to recover the unique stable solution for two-thirds of the models.

Overall, the Newton methods are on the one hand slower than Dynare (QZ), but on the other more accurate than Dynare. While the minimal run times for all algorithms except the Modified Newton algorithm are one order of magnitude less than Dynare, maximum run times are up to two orders of magnitude higher than Dynare, with the median run time being around two times as high as Dynare for the five algorithms. In line with the findings from above, the Modified Newton algorithm is the slowest and least accurate. For all 99 models, run time for this algorithm ranges between seven to 330 times as slow as Dynare. Since this algorithm only converges linearly, 686 iterations are needed until convergence to a solution, compared to a maximum of 9 for all other methods.

All three line search methods perform relatively similar in terms of convergence to the stable solution, run times and iterations needed. In all of these dimensions they perform slightly worse than the Baseline algorithm. The solutions of all algorithms except the Modified algorithm are at least one order of magnitude more precise than the solution of

[13] See http://www.macromodelbase.com.

[14] The model of Smets and Wouters (2007) examined in detail in the previous section is among these models.

[15] Currently, this is 99 models, ranging from small scale DSGE models to models from policy institutions containing hundreds of variables. Some of the models in the database are deterministic and/or use nonlinear or nonrational (e.g., adaptive) expectations and, hence, are not appropriate for our comparison here.

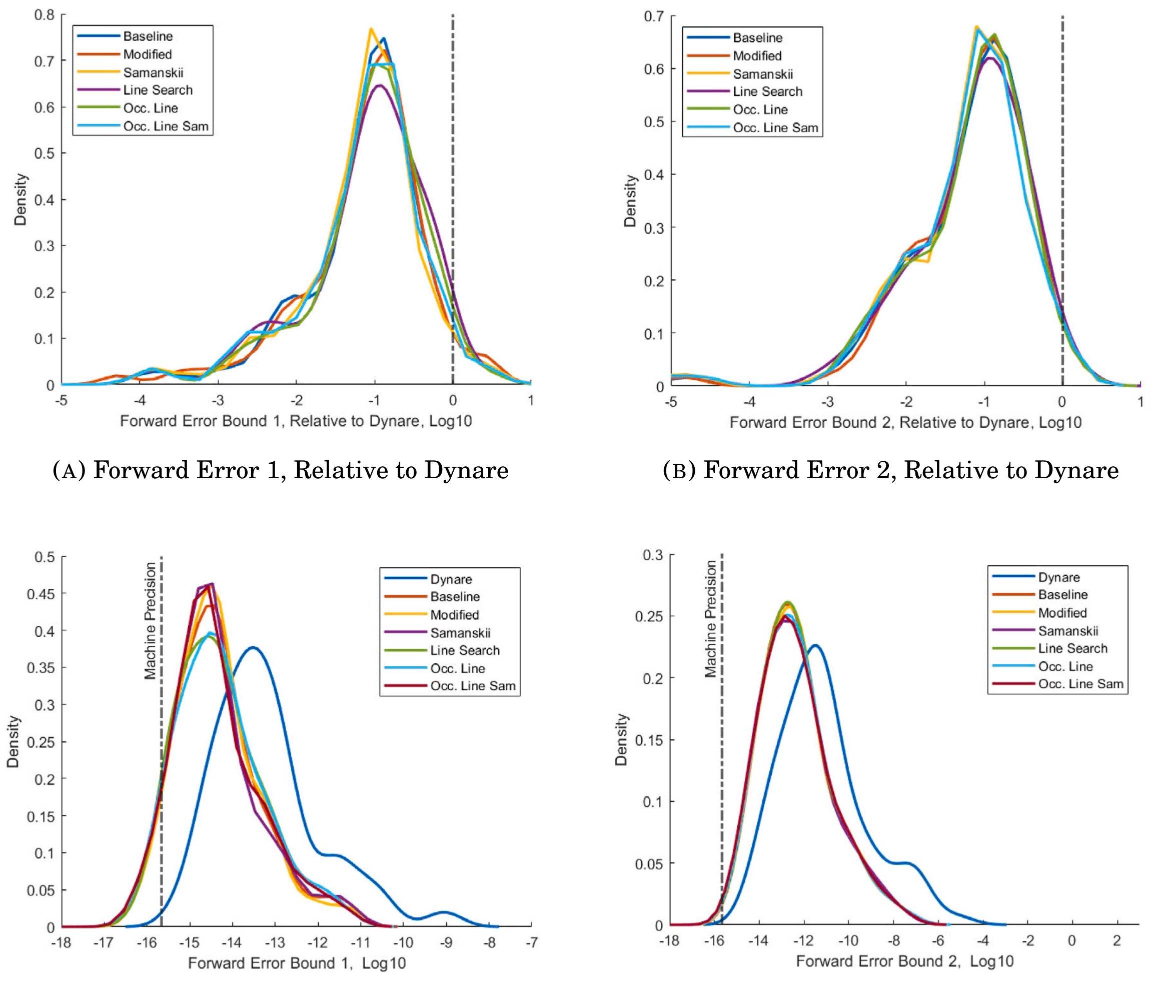
[16] Due to parallel uncontrollable demands on working memory run times can differ between runs so that our approach ensures accurate measurement.

[17] See the appendix for the den Haan and Marcet (1994) statistics for the (Smets and Wouters, 2007) contained in this experiment. In contrast to the results here using the forward errors of Meyer-Gohde (2023), the den Haan and Marcet (1994) statistic is unable to distinguish between the different method's solutions.

Table 4
Results relative to Dynare (QZ): 99 MMB models (starting guess: solution Dynare (QZ)).

Method	Convergence	Run time			Forward error 1			Forward error 2			Iterations
		Median	Min	Max	Median	Min	Max	Median	Min	Max	
Dynare (QZ)	99	1	1	1	1	1	1	1	1	1	1
Baseline Newton Method	99	0.34	0.032	29	0.099	3.5e−15	2.9	0.1	2.5e−15	1.5	1
Modified Newton Method	99	0.34	0.031	25	0.099	3.5e−15	2.9	0.1	2.5e−15	1.5	1
With Šamanskii Technique	99	0.49	0.055	70	0.087	1.3e−29	3.2	0.082	9.6e−30	1.1	1
With Line Searches	99	0.34	0.033	30	0.099	0.00012	3.1	0.093	7.8e−06	1.5	1
With Occ. Line Searches	99	0.33	0.032	63	0.096	3.5e−15	3.1	0.093	2.5e−15	1.5	1
With Occ. LS & ŠT	99	0.54	0.058	71	0.094	1.3e−29	3.1	0.082	9.6e−30	1.1	1

• For Dynare, refer to Adjemian et al. (2011).
• Run Time and foward errors relative to Dynare.
• Forward error 1 and 2 are the upper bounds for the true forward error, see (47).



(A) Forward Error 1, Relative to Dynare

(B) Forward Error 2, Relative to Dynare

(C) Forward Error 1

(D) Forward Error 2

Fig. 3. Distribution of forward error bounds relative to Dynare for the Macroeconomic Model Data Base (MMB). Figs. 3(a), 3(b) plot the distribution of model solutions against the upper bounds of the forward error 1 and 2 for all algorithms, log10 scale on the x axis, 99 MMB models (starting guess: solution Dynare(QZ)).

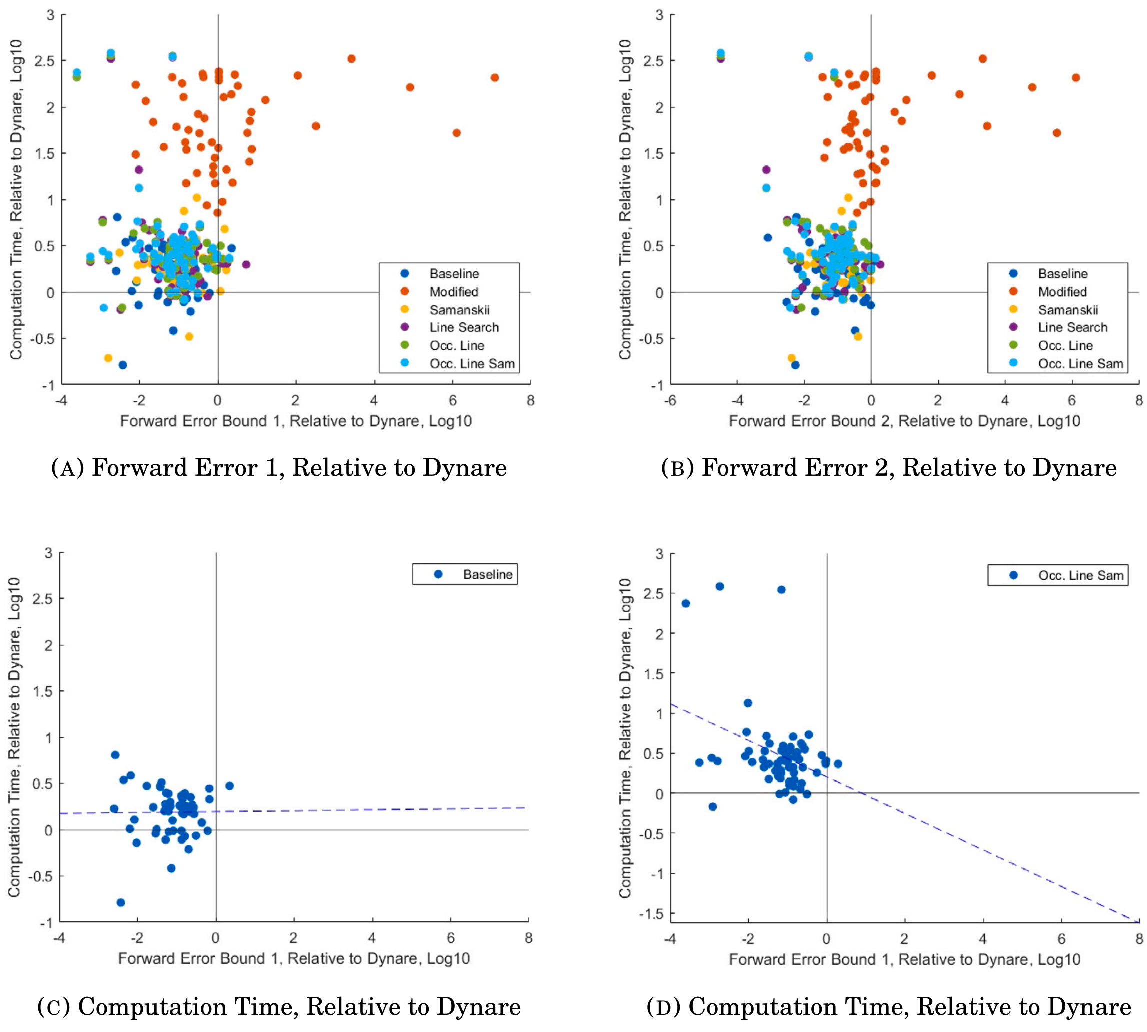
Dynare in terms of the median of the forward error bounds. The Occasional Line Search Method with Šamanskii Technique algorithm is most accurate being at the median two orders of magnitude more precise than Dynare. With this algorithm improving on global convergence by combining the strengths of line searches and the Šamanskii Technique, it is surprising that this is not visible compared to the other two line search algorithms. Hence, the performance of line search and Šamanskii methods in terms of run time and accuracy are disappointing for the set of DSGE models in the MMB in the context of the results of Higham and Kim (2001) and Long et al. (2008), as the Baseline Newton algorithm surprisingly appears to perform equally well and arguably better along these dimensions. The inclusion of line searches and its mitigation of excessively large Newton steps, however, succeeds in improving the convergence of Newton methods to the stable solvent when the zero matrix is used as the initialization.

Fig. 4 compares the accuracy and computation time of all different Newton methods with Dynare for the models in the MMB that converge to the unique stable solvent. The figures confirm that all algorithms except the Modified Newton Method are generally slower but more accurate than Dynare's QZ algorithm, with their clouds of points (each

Table 5
Results: 99 MMB models (starting guess: zero-matrix).

Method	Convergence	Run time			Forward error 1			Forward error 2			Iterations
		Median	Min	Max	Median	Min	Max	Median	Min	Max	
Dynare (QZ)	99	1	1	1	1	1	1	1	1	1	1
Baseline Newton Method	53	1.7	0.16	6.9	0.076	0.0025	2.3	0.1	0.00082	1.2	8
Modified Newton Method	51	62	7.2	329.85	0.85	0.008	1.2e+07	0.65	0.035	1.3e+06	686
With Šamanskii Technique	43	1.9	0.19	10.41	0.13	0.0016	1.6	0.1	0.0042	1.1	5
With Line Searches	67	2.2	0.64	340.54	0.11	0.00025	5.3	0.086	3.2e−05	1.8	9
With Occ. Line Searches	67	2.3	0.68	354.94	0.096	0.00025	2	0.094	3.2e−05	0.99	9
With Occ. LS & ŠT	67	2.5	0.68	382.05	0.09	0.00025	2	0.082	3.2e−05	1.4	9

• For Dynare, refer to Adjemian et al. (2011).
• Run time and forward errors relative to Dynare, number of models converging to the stable solution and median of number of iterations in absolute terms.
• Forward error 1 and 2 are the upper bounds for the true forward error, see (47).



(A) Forward Error 1, Relative to Dynare

(B) Forward Error 2, Relative to Dynare

(C) Computation Time, Relative to Dynare

(D) Computation Time, Relative to Dynare

Fig. 4. Forward Errors and Computation Time for the Macroeconomic Model Data Base (MMB). Figs. 4(a), 4(b) plot the computation times against the upper bounds of the forward error 1 and 2 for all methods, log10 scale on both axes.

corresponding to a model within the database) being in the upper left quadrant (corresponding to higher run times, but lower forward errors). Figs. 4(c) and 4(d) focus on the Baseline and occasional line search method including a regression line. A negative relationship between relative speed and accuracy is present (and holds for all methods except the Modified Newton method for which the relationship seems to be positive, see the additional figures in the online appendix). This points to a logical tradeoff: increasing the number of Newton steps (by, say, lowering the convergence threshold or altering the criterion) increases the accuracy of the solution at the price of increasing the necessary computation cost.

Fig. 5 compares the different Newton methods with Dynare for the models in the MMB that converge to the unique stable solvent but now with a focus on the size of the model, which is taken as the number of state variables — "purely backward looking" and "mixed" variables in Dynare nomenclature, see the appendix for more on the declination of variables. Fig. 5(c) displays no trend in the relative computation time and the size of the model, implying that the methods here face increases

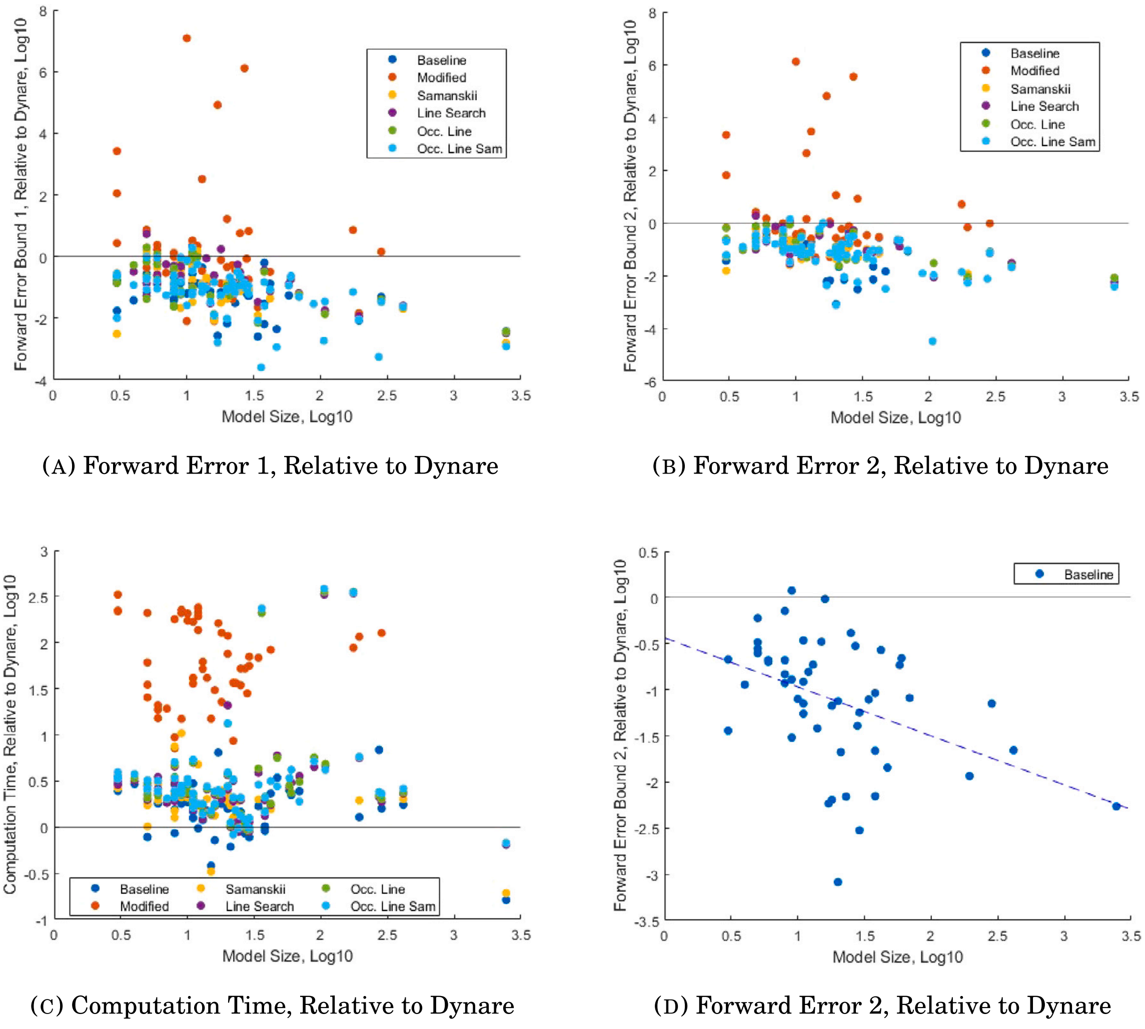


Fig. 5. Forward Errors, Computation Time and Number of Variables for the Macroeconomic Model Data Base (MMB). Figs. 5(a), 5(b) plot the upper bounds of the forward error 1 and 2 against model size as measured by the number of state variable for all methods, log10 scale on both axes.

in the computational demands proportional to those of Dynare. There is one important exception: the largest model studied — for this model with 2450 endogenous state variables, all of the Newton methods except the Modified algorithm are faster than Dynare and converge to the same solution. The Figs. 5(a) and 5(b) again confirm that all algorithms except the Modified Newton Method are generally more accurate than Dynare's QZ algorithm, with their clouds of points (each corresponding to a model within the database) being below the x-axis. Fig. 5(d) focuses in on the Baseline method including a regression line and finds a negative relationship between model size and accuracy (reexamining Figs. 5(a) and 5(b), this relationship can be seen for all Newton algorithms), indicating that as the model becomes larger, the accuracy advantage of Newton methods increases.

Fig. 6 provides an overview of the entire distribution of forward errors, the upper row relative to those from Dynare's QZ method and the lower in absolute terms, using the different Newton-based methods presented here. Forward errors left of the vertical line are thus smaller than Dynare for both figures in the upper row. For both the first, Fig. 6(a), and second, Fig. 6(b), upper bounds on the forward error, we see an obvious shift to the left on a log scale of slightly more than one order of magnitude for all the Newton methods apart from the Modified algorithm and, from the lower row, we see that this entails tightening the distributions as well as shifting them closer to machine precision — a lower convergence criterion would allow additional Newton steps and bring yet more solutions below machine precision. In particular we see in Fig. 6(a) that the baseline and occasional line search with Šamanskii methods drive the distribution of forward errors against machine precision, highlighting the advantage in terms of accuracy of our methods.

6. Conclusion

We have presented and applied Newton-based methods from the recent applied mathematics literature for solving the matrix quadratic equation underlying the solution of linear DSGE models as an alternative to the current standard of a generalized Schur or QZ decomposition (Moler and Stewart, 1973; Golub and van Loan, 2013). Applying the methods to the suite of models in the Macroeconomic Model Data Base (MMB), we find that although Newton-based methods might appear to be a competitive alternative, offering up to several orders of magnitude smaller forward errors at computational costs of the same order of magnitude, they are not guaranteed to converge to the unique stable solution that is generally required in the DSGE literature. While line-search methods improved the frequency of convergence to this solution from about 50% of the models to about 66%, this still poses a prohibitive hurdle for considering Newton-based methods as a replacement for the current generalized Schur or QZ decomposition standard. Indeed, Higham and Kim (2001) note that determining to

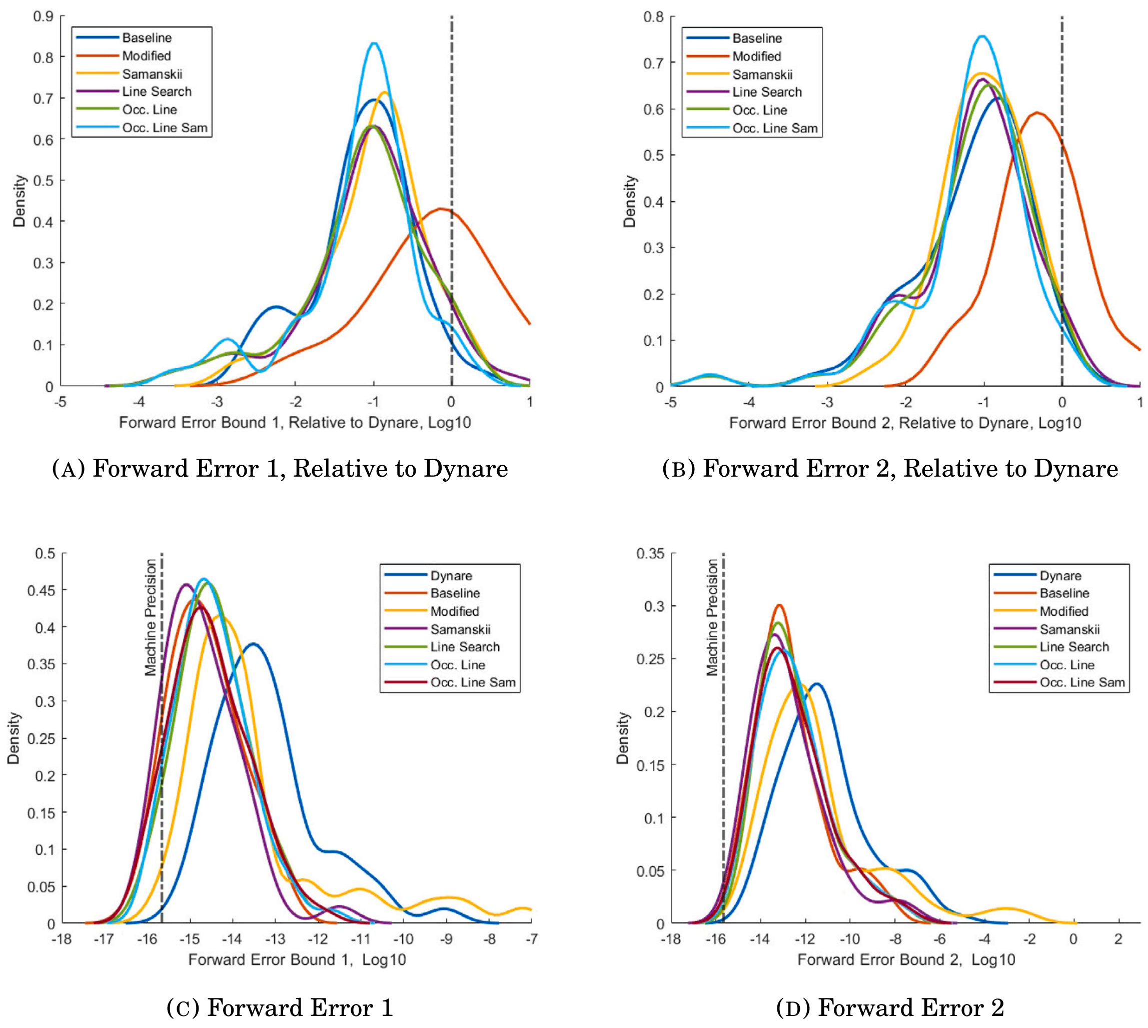


Fig. 6. Distribution of forward error bounds relative to Dynare for the Macroeconomic Model Data Base (MMB). Figs. 6(a), 6(b) plot the distribution of model solutions against the upper bounds of the forward error 1 and 2 for all algorithms, log10 scale on the x axis, 99 MMB models (starting guess: zero matrix).

which solvent the method will converge a priori (as can be done via basins of convergence for scalar quadratic equations) remains an open problem for the mathematics literature.

That being said, our Newton-based methods perform quite successfully in iterative environments or to refine existing solutions — in line with this importance of the initialization for the convergence speed and properties. In filling in an increasingly dense grid of parameterizations for the Taylor rule in the model of Smets and Wouters (2007), Newton-based methods can initialize with the solution from the previous parameterization and significantly outperform the current generalized Schur or QZ method both in terms of computational costs and forward error. Taking the solution from QZ as the initialization, all of the Newton methods provide roughly an order of magnitude improvement in the accuracy of the solution at a fraction of the original computational cost. This initialization and iteration makes applying our collection of Newton methods to improve the accuracy of solutions to linear DSGE models a fruitful avenue of application, as is done in Meyer-Gohde (2023) where QZ based methods from the literature are shown to generate inaccuracies of economic consequence in several macro-finance models.

Iterative Newton-based methods like we have presented here could analogously reduce the computational burden associated with solving the model for iterative estimation procedures and might be adapted to more quickly and/or accurately perform likelihood calculations or solve heterogenous agent models. We leave this, however, to future research.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Johanna Saecker reports travel was provided by International Association for Applied Econometrics IAAE, Germany via the 2022 Student Travel Grant. Alexander Meyer-Gohde and Johanna Saecker report financial support provided by DFG, Germany through grant nr. 465469938 “Numerical diagnostics and improvements for the solution of linear dynamic macroeconomic models”.

Data availability

Solving Linear DSGE Models with Newton Methods (Replication) (Original data) (Mendeley Data)

.

Acknowledgments

We are grateful to Maximilian Thomin and Raphael Abiry for invaluable research assistance. We thank Angus Chu, the editor, two anonymous referees and participants of the International Conference of Computing in Economics and Finance 2022, the Conference of the International Association for Applied Econometrics 2022 as well as the researchers of Sveriges Riksbank for useful comments and discussions.

Any and all errors are entirely our own. The authors declare the following financial interests: Johanna Saecker thanks the IAAE, Germany for the 2022 Student Travel Grant. This research was supported by the DFG, Germany through grant nr. 465469938 “Numerical diagnostics and improvements for the solution of linear dynamic macroeconomic models”.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.econmod.2024.106670.

References

Adjemian, S., Bastani, H., Juillard, M., Mihoubi, F., Perendia, G., Ratto, M., Villemot, S., 2011. Dynare: Reference Manual, Version 4. Dynare Working Papers 1, CEPREMAP.

Al-Sadoon, M.M., 2018. The linear systems approach to linear rational expectations models. Econom. Theory 34, 628–658.

Al-Sadoon, M.M., 2023. The Spectral Approach to Linear Rational Expectations Models. Discussion Paper 2007.13804, arXiv.

An, S., Schorfheide, F., 2007. Bayesian analysis of DSGE models. Econometric Rev. 26 (2–4), 113–172.

Anderson, G.S., 2010. A reliable and computationally efficient algorithm for imposing the saddle point property in dynamic models. J. Econom. Dynam. Control 34 (3), 472–489.

Anderson, G.S., Levin, A., Swanson, E., 2006. Higher-Order Perturbation Solutions to Dynamic Discrete-Time Rational Expectations Models. Discussion Paper 2006–01, Federal Reserve Bank of San Francisco Working Paper Series.

Anderson, G.S., Moore, G., 1985. A linear algebraic procedure for solving linear perfect foresight models. Econom. Lett. 17 (3), 247–252.

Aruoba, S.B., Fernández-Villaverde, J., Rubio-Ramírez, J.F., 2006. Comparing solution methods for dynamic equilibrium economies. J. Econom. Dynam. Control 30 (12), 2477–2508.

Benner, P., Mehrmann, V., Sima, V., Van Huffel, S., Varga, A., 1999. SLICOT—A subroutine library in systems and control theory. In: Datta, B.N. (Ed.), Applied and Computational Control, Signals, and Circuits: Volume 1. Boston, MA, pp. 499–539.

Binder, M., Pesaran, M.H., 1997. Multivariate linear rational expectations models: Characterization of the nature of the solutions and their fully recursive computation. Econom. Theory 13 (6), 877–888.

Blanchard, O.J., 1979. Backward and forward solutions for economies with rational expectations. Am. Econ. Rev. 69 (2), 114–118.

Blanchard, O.J., Kahn, C.M., 1980. The solution of linear difference models under rational expectations. Econometrica 48 (5), 1305–1311.

Cayley, A., 1879. Desiderata and suggestions: No. 3. The Newton-Fourier imaginary problem. Amer. J. Math. 2 (1), 97.

Chen, X.S., Lv, P., 2018. On estimating the separation between (A,B) and (C,D) associated with the generalized Sylvester equation $AXD - BXC = E$. J. Comput. Appl. Math. 330, 128–140.

Chu, K.-w.E., 1987. The solution of the matrix equations $AXB - CXD = E$ and $(YA - DZ, YC - BZ) = (E, F)$. Linear Algebra Appl. 93, 93–105.

Corless, R.M., Fillion, N., 2013. A Graduate Introduction to Numerical Methods. Springer.

Dennis, Jr., J.E., Traub, J.F., Weber, R.P., 1976. The algebraic theory of matrix polynomials. SIAM J. Numer. Anal. 13 (6), 831–845.

Gantmacher, F.R., 1959. The Theory of Matrices. Vol. I&II, Chelsea Publishing Company, New York, NY.

Gardiner, J.D., Laub, A.J., Amato, J.J., Moler, C.B., 1992a. Solution of the Sylvester matrix equation AXBT + CXDT = E. ACM Trans. Math. Software 18 (2), 223–231.

Gardiner, J.D., Wette, M.R., Laub, A.J., Amato, J.J., Moler, C.B., 1992b. Algorithm 705; a FORTRAN-77 software package for solving the Sylvester matrix equation AXBT + CXDT = E. ACM Trans. Math. Software 18 (2), 232–238.

Golub, G.H., van Loan, C.F., 2013. Matrix Computations, fourth ed. The Johns Hopkins University Press.

den Haan, W.J., Marcet, A., 1994. Accuracy in simulations. Rev. Econom. Stud. 61 (1), 3–17.

Hammarling, S., Munro, C.J., Tisseur, F., 2013. An algorithm for the complete solution of quadratic eigenvalue problems. ACM Trans. Math. Software 39 (3), 18:1–18:19.

Higham, N.J., 2002. Accuracy and Stability of Numerical Algorithms, second ed. Society for Industrial and Applied Mathematics.

Higham, N.J., Kim, H.-M., 2001. Solving a quadratic matrix equation by Newton's method with exact line searches. SIAM J. Matrix Anal. Appl. 23 (2), 499–519.

Hopkins, T., 2002. Remark on algorithm 705: A Fortran-77 software package for solving the Sylvester matrix equation AXBT + CXDT = E. ACM Trans. Math. Software 28 (3), 372–375.

Huber, J., Meyer-Gohde, A., Saecker, J., 2023. Solving Linear DSGE Models With Structure-Preserving Doubling Methods. IMFS Working Paper Series 195, Goethe University Frankfurt, Institute for Monetary and Financial Stability (IMFS).

Judd, K.L., 1992. Projection methods for solving aggregate growth models. J. Econom. Theory 58 (2), 410–452.

Juillard, M., Villemot, S., 2011. Multi-country real business cycle models: Accuracy tests and test bench. J. Econom. Dynam. Control 35 (2), 178–185.

Kågström, B., Poromaa, P., 1996. Computing eigenspaces with specified eigenvalues of a regular matrix pair (A,B) and condition estimation: theory, algorithms and software. Numer. Algorithms (12), 369–407.

Klein, P., 2000. Using the generalized Schur form to solve a multivariate linear rational expectations model. J. Econom. Dynam. Control 24 (10), 1405–1423.

Köhler, M., 2021. Approximate Solution of Non-Symmetric Generalized Eigenvalue Problems and Linear Matrix Equation on HPC Platforms. Logos Verlag Berlin, Magdeburg, Germany.

Köhler, M., 2022. Matrix Equations PACKage – A Fortran Library for the Solution of Sylvester-Like Matrix Equations. Zendo, http://dx.doi.org/10.5281/zenodo.10016456.

Kågström, B., 1994. A perturbation analysis of the generalized sylvester equation $(AR - LB, DR - LE) = (C, F)$. SIAM J. Matrix Anal. Appl. 15 (4), 1045–1060.

Kågström, B., Poromaa, P., 1996. LAPACK-style algorithms and software for solving the generalized sylvester equation and estimating the separation between regular matrix pairs. ACM Trans. Math. Softw. 22 (1), 78–103.

Lan, H., Meyer-Gohde, A., 2014. Solvability of perturbation solutions in DSGE models. J. Econom. Dynam. Control 45, 366–388.

Long, J., Hu, X., Zhang, L., 2008. Improved Newton's method with exact line searches to solve quadratic matrix equation. J. Comput. Appl. Math. 222 (2), 645–654.

Meyer-Gohde, A., 2022. Solving Linear DSGE Models with Bernoulli Iterations. mimeo, Goethe University Frankfurt, Institute for Monetary and Financial Stability (IMFS).

Meyer-Gohde, A., 2023. Numerical Stability Analysis of Linear DSGE Models - Backward Errors, Forward Errors and Condition Numbers. IMFS Working Paper Series 193, Goethe University Frankfurt, Institute for Monetary and Financial Stability (IMFS).

Moler, C.B., Stewart, G.W., 1973. An algorithm for generalized matrix eigenvalue problems. SIAM J. Numer. Anal. 10 (2), 241–256.

Rendahl, P., 2017. Linear Time Iteration. Discussion paper, IHS Economics Series.

Schröder, E., 1870. Ueber unendlich viele algorithmen zur auflösung der gleichungen. Math. Ann. 2, 317–365.

Sims, C.A., 2001. Solving linear rational expectations models. Comput. Econ. 20 (1–2), 1–20.

Smets, F., Wouters, R., 2007. Shocks and frictions in US business cycles: A Bayesian DSGE approach. Am. Econ. Rev. 97 (3), 586–606.

Stewart, G.W., 1971. Error bounds for approximate invariant subspaces of closed linear operators. SIAM J. Numer. Anal. 8 (4), 796–808.

Tisseur, F., 2000. Backward error and condition of polynomial eigenvalue problems. Linear Algebra Appl. 309 (1), 339–361.

Tisseur, F., Meerbergen, K., 2001. The quadratic eigenvalue problem. SIAM Rev. 43 (2), 235–286.

Uhlig, H., 1999. A toolkit for analysing nonlinear dynamic stochastic models easily. In: Marimon, R., Scott, A. (Eds.), Computational Methods for the Study of Dynamic Economies. Oxford University Press, pp. 30–61, chap. 3.

Villemot, S., 2011. Solving Rational Expectations Models at First Order: What Dynare Does. Dynare Working Papers 2, CEPREMAP.

Wieland, V., Afanasyeva, E., Kuete, M., Yoo, J., 2016. New methods for macro-financial model comparison and policy analysis. In: Taylor, J.B., Uhlig, H. (Eds.), Handbook of Macroeconomics. In: Handbook of Macroeconomics, vol. 2, Elsevier, pp. 1241–1319.

Wieland, V., Cwik, T., Müller, G.J., Schmidt, S., Wolters, M., 2012. A new comparative approach to macroeconomic modeling and policy analysis. J. Econ. Behav. Organ. 83 (3), 523–541.