SOLVING LINEAR DSGE MODELS WITH NEWTON METHODS

ALEXANDER MEYER-GOHDE AND JOHANNA SAECKER

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

ABSTRACT. 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. *JEL classification codes*: C61, C63, E17
Keywords: Numerical accuracy; DSGE; Solution methods

E-mail address: meyer-gohde@econ.uni-frankfurt.de, saecker@hof.uni-frankfurt.de.

Date: October 5, 2022.

We are grateful to Maximilian Thomin and Raphael Abiry for invaluable research assistance. We thank participants of the International Conference of Computing in Economics and Finance 2022 and the Conference of the International Association for Applied Econometrics (IAAE) 2022 for useful comments and discussions. Any and all errors are entirely our own. Johanna Saecker thanks the IAAE for the 2022 Student Travel Grant. This research was supported by the DFG through grant nr. 465469938 "Numerical diagnostics and improvements for the solution of linear dynamic macroeconomic models".

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 (for Bernoulli iterative methods, see Meyer-Gohde (2022b)) and applying them to the solution of linear DSGE models. 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 (2022a).

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, Bastani, Juillard, Mihoubi, Perendia, Ratto, and Villemot, 2011) 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 (2022a). Long, Hu, and Zhang (2008)

[1] Such as the methods of Binder and Pesaran (1997), Anderson (2010) and the cyclic reduction method of Dynare (Adjemian, Bastani, Juillard, Mihoubi, Perendia, Ratto, and Villemot, 2011).

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.[2] First we apply the different methods to the models in the Macroeconomic Model Data Base (MMB) (see Wieland, Cwik, Müller, Schmidt, and Wolters, 2012; Wieland, Afanasyeva, Kuete, and Yoo, 2016), comparing the performance to the QZ-based 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.[3] 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

[2]We use Dynare's (Adjemian, Bastani, Juillard, Mihoubi, Perendia, Ratto, and Villemot, 2011) implementation of the QZ method, documented in Villemot (2011), to compare the Newton methods with.

[3]Our measure of accuracy is the forward error of Meyer-Gohde (2022a).

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 (Adjemian, Bastani, Juillard, Mihoubi, Perendia, Ratto, and Villemot, 2011), Gensys (Sims, 2001), (Perturbation) AIM (Anderson and Moore, 1985; Anderson, Levin, and Swanson, 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 $f : \mathbb{R}^{n_y} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_e} \to \mathbb{R}^{n_y}$; $y_t \in \mathbb{R}^{n_y}$ 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

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

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, $\overline{y} \in \mathbb{R}^{n_y}$ a vector such $\overline{y} = y(\overline{y}, 0)$ and $0 = f(\overline{y}, \overline{y}, \overline{y}, 0)$ 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,

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

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, $\overline{y}$.

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 ($E_t[\varepsilon_{t+1}] = 0$), 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)[4] 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 $f(x):\mathbb{C}^1\to\mathbb{C}^1$

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

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

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

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

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

[4]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 (2022a).

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 the 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[5] $M(\lambda) : \mathbb{C} \to \mathbb{C}^{n \times n}$ (here of degree two), are given via

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

[5]See, e.g., Dennis, Jr., Traub, and Weber (1976, p. 835) or Gantmacher (1959, vol. I, p. 228).

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. An explicit link between the quadratic matrix problem and the quadratic eigenvalue problem is given via

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

which has been reviewed extensively by Tisseur and Meerbergen (2001) and for which Hammarling, Munro, and Tisseur (2013) provide a comprehensive method to improve the accuracy of its solutions.

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

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

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

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

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. This gives the baseline Newton procedure as

Baseline Newton Method

- Given A, B, C, an initial P_0, and a convergence criterion ϵ
- While criterion$(P_j) > \epsilon$
  (1) Solve for ΔP_j in

$$A\,\Delta P_j P_j + \left(A\,P_j + B\right)\Delta P_j = -M(P_j) \tag{22}$$

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

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

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 ϵ
- While criterion$(P_j) > \epsilon$
  (1) Solve for ΔP_j in

$$A\,\Delta P_j P_0 + (A\,P_0 + B)\,\Delta P_j = -M(P_j) \tag{24}$$

  (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. This simplifies the solvability considerations, presents an opportunity to economize on computational costs, but comes at the cost of quadratic convergence (Long, Hu, and Zhang, 2008).

3.4. Newton's Method with Šamanskii Technique

Combining the two previous techniques, the Šamanskii algorithm, Algorithm 2.3 of Long, Hu, and Zhang (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 ϵ
- 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

$$A\,\Delta P_{j,i} P_j + \left(A P_j + B\right)\Delta P_{j,i} = -M(P_{j,i}) \tag{25}$$

    - (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, Hu, and Zhang (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.

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

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

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 ϵ
- While criterion$(P_j) > \epsilon$
  (1) Solve for ΔP_j in

$$A\,\Delta P_j P_j + \left(A P_j + B\right)\Delta P_j = -M(P_j) \tag{29}$$

  (2) Solve for t_j in

$$t_j = \underset{x\in[0,2]}{\operatorname{argmin}} \left\| M(P_j + x\Delta P_j)\right\|_F^2 \tag{30}$$

  (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, Hu, and Zhang (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 ϵ and ϵ_0
- While criterion$(P_j) > \epsilon$
  (1) Solve for ΔP_j in

$$A\,\Delta P_j P_j + \left(A\,P_j + B\right)\Delta P_j = -M(P_j) \tag{31}$$

  (2) if criterion$(P_j + \Delta P_j) > \epsilon_0$
    (a) Solve for t_j in

$$t_j = \underset{x\in[0,2]}{\operatorname{argmin}} \left\| M(P_j + x\Delta P_j)\right\|_F^2 \tag{32}$$

    (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, Hu, and Zhang (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 ϵ and ϵ_0
- While criterion$(P_j) > \epsilon$
  - (1) Solve for ΔP_j in

$$A\,\Delta P_j P_j + \left(A\,P_j + B\right)\Delta P_j = -M(P_j) \tag{33}$$

  - (2) if criterion$(P_j + \Delta P_j) > \epsilon_0$
    - (a) Solve for t_j in

$$t_j = \underset{x\in[0,2]}{\operatorname{argmin}} \left\| M(P_j + x\Delta P_j)\right\|_F^2 \tag{34}$$

    - (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

$$A\,\Delta P_{j,i} P_j + \left(A\,P_j + B\right)\Delta P_{j,i} = -M(P_{j,i}) \tag{35}$$

        - (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, Hu, and Zhang (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, 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 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 $AXP_j + \left(AP_j + B\right)X + M(P_j) = 0$ requires

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

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.

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, Hu, and Zhang (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 (2022a),

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

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

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

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

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

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 $\operatorname{sep}^{-1}\left[\left(A, A\hat{P} + B\right), \left(I, -\hat{P}\right)\right] = \left\| H_{\hat{P}}^{-1} \right\|_2 = \sigma_{\min}\left(H_{\hat{P}}\right)^{-1}$, which - from above - can be arbitrarily larger than the inverse of the minimal distance between the spectra of the pencils $\left(A, A\hat{P} + B\right), \left(I, -\hat{P}\right)$. This 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, and 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. This gives us two measures of the relative error of a numerically computed solution to an exact solution.

5. Applications

We conduct a number of experiments to assess the performance of the algorithms presented above. First we compare the different Newton-based methods from section 3 with Dynare's QZ-based method[6] on the suite of models in the MMB. Here we examine the convergence of the different methods to the stable solvent,

[6] 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

the accuracy of the solvents, and the associated computation times. We compare both initializing the Newton algorithms with the zero matrix (an uninformed initialization of a stable solvent) and the output from the QZ algorithm. Second, we explore 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.

5.1. MMB Suite Comparison

The Macroeconomic Model Data Base (MMB) (see Wieland, Cwik, Müller, Schmidt, and Wolters, 2012; Wieland, Afanasyeva, Kuete, and Yoo, 2016) is a model comparison initiative at the Institute for Monetary and Financial Stability (IMFS)[7] 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. 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,[8] the varying sizes of which are summarized in figure 1.

Among the models in the MMB is the model of Smets and Wouters (2007) upon which we place specific attention, particularly on the monetary policy rule in the

details are in the appendix and are of independent interest as they supplement Villemot (2011) by providing a detailed block-matrix derivation of the procedure.

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

[8]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 non-rational (e.g., adaptive) expectations and, hence, are not appropriate for our comparison here.

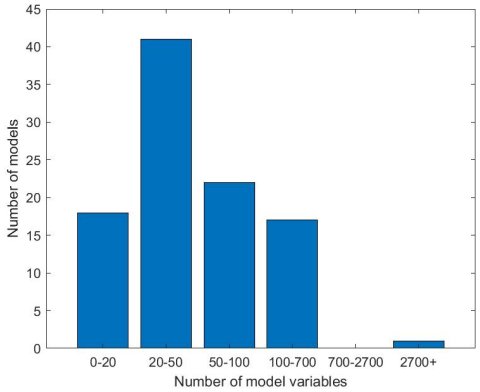


FIGURE 1. Histogram over the number of variables for the 99 MMB models

Figure 1 plots the number of model variables over the amount of MMB models.

Currently the total amount of models considered is 99.

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,

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

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.

Table 1 presents the results of the different Newton methods from above alongside the QZ-based solution of Dynare (Adjemian, Bastani, Juillard, Mihoubi, Perendia, Ratto, and Villemot, 2011) for the posterior mode parameterization of

Method	Relative Performance		Forward Errors		Iterations
	Run Time	Max Abs. Diff.	Bound 1	Bound 2	
Dynare (QZ)	0.002	0	4.32e-15	7.19e-14	1
Baseline Newton Method	1.22	108	4.00e-16	4.32e-14	10
Modified Newton Method	36.68	1.54e-12	3.02e-14	6.65e-13	650
with Šamanskii Technique	1.19	107.1	8.82e-16	8.28e-14	7
with Line Searches	1.91	7.92e-13	5.92e-16	7.1e-15	18
with Occ. Line Searches	2.07	7.79e-13	5.91e-16	8.43e-15	19
with Occ. LS & ŠT	2.1	7.79e-13	3.23e-16	8.27e-15	18

TABLE 1. Results: Model of Smets and Wouters (2007)

- For Dynare, refer to Adjemian, Bastani, Juillard, Mihoubi, Perendia, Ratto, and Villemot (2011).
- Run Time for Dynare in seconds, for all others, run time relative to Dynare.
- 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).

Smets and Wouters (2007). 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 (2022a)). 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).

Before turning to the Smets and Wouters (2007) model in more detail in the next section, we now examine the remainder of the models of the MMB and assess the various Newton-based methods relative to the QZ method. We begin by solving each of the applicable models in the MMB 100 times using the different Newton methods, initializing the methods with a zero matrix, and the QZ solver from Dynare, taking the results as the average within the middle three quintiles to reduce the effects of outliers.

Table 2 summarizes the results. 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

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	1.74	0.16	6.86	0.11	9.82e-4	7.10	0.14	1.41e-4	1.23	8
Modified Newton Method	51	61.81	7.15	329.85	7.58	0.01	1.48e6	7.97	0.04	9.42e5	686
with Šamanskii Technique	43	1.95	0.19	10.41	0.14	0.004	35.78	0.13	0.008	11.16	5
with Line Searches	67	2.25	0.64	340.54	0.15	3.89e-4	8.74	0.13	2.85e-6	16.65	9
with Occ. Line Searches	67	2.28	0.68	354.94	0.15	4.13e-4	236.57	0.15	1.98e-6	364.74	9
with Occ. LS & ŠT	67	2.51	0.68	382.05	0.08	3.97e-4	4.98	0.09	1.83e-6	8.23	9

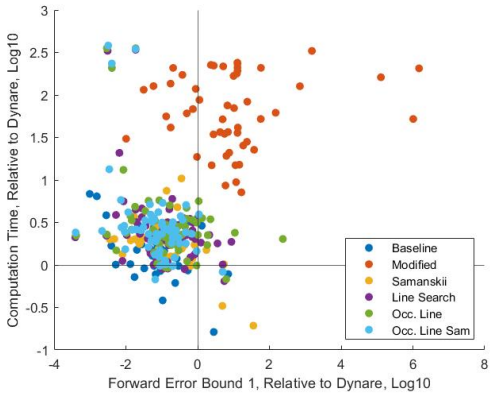
TABLE 2. Results: 99 MMB models (starting guess: zero-matrix).

- For Dynare, refer to Adjemian, Bastani, Juillard, Mihoubi, Perendia, Ratto, and Villemot (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).

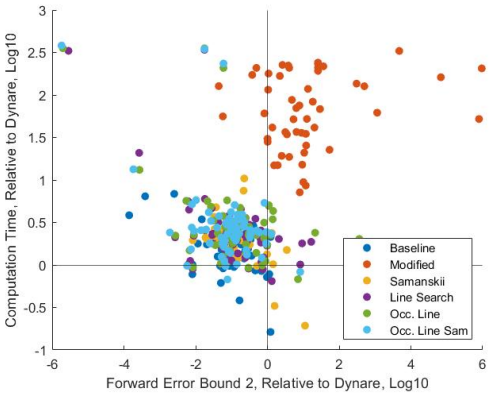
algorithm ranges between seven to 330 times as slow as Dynare. Since this algorithm only converges linearly, 686 iterations are needed until convergence to a solution, compared to a maximum of 9 for all other methods.

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 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, Hu, and Zhang (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.

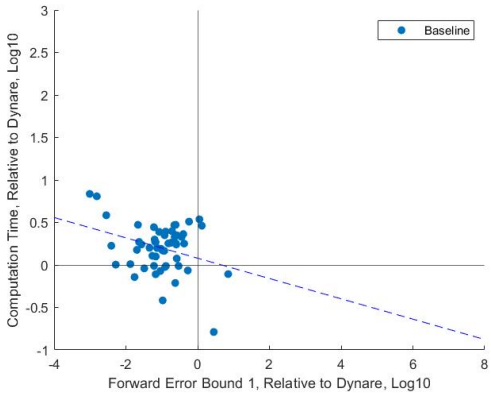
Figure 2 compares all different Newton methods with Dynare for the models in the MMB that converge to the unique stable solvent. The figures confirm that all algorithms except the Modified Newton Method are generally slower but more accurate than Dynare's QZ algorithm, with their clouds of points (each corresponding to a model within the database) being in the upper left quadrant (corresponding to higher run times, but lower forward errors). Figure 2c and 2d 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 appendix). This points to a logical tradeoff: increasing the number of Newton steps (by, say, lowering



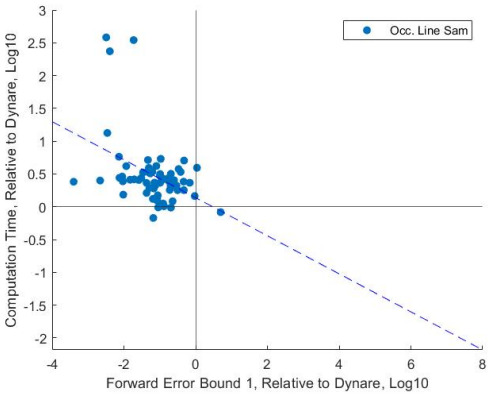
(A) Forward Error 1, Relative to Dynare



(B) Forward Error 2, Relative to Dynare



(C) Computation Time, Relative to Dynare



(D) Computation Time, Relative to Dynare

FIGURE 2. Forward Errors, Computation Time and Number of Variables for the Macroeconomic Model Data Base (MMB)

Figures 2a, 2b plot the computation times against the upper bounds of the forward error 1 and 2 for all methods, log10 scale on both axes.

the convergence threshold or altering the criterion) increases the accuracy of the solution at the price of increasing the necessary computation cost.

This negative relationship implies that the Newton-based methods from above might also be useful as solution refinements, that is, given a solvent from another algorithm - such as QZ - Newton methods might be used to further improve the accuracy of these solutions. Accordingly, we re-run the experiment starting now with Dynare (QZ)'s solution as the starting guess instead of the zero-matrix. Table 3 shows that all algorithms converge now to the unique stable solution for all models. With a precise starting guess, all algorithms need only one iteration to satisfy the convergence criterion and none of the algorithms diverge to a different

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	0.1	7.7e-16	1.8	0.11	2.5e-15	1.5	1
Modified Newton Method	99	0.34	0.031	25	0.1	7.7e-16	1.8	0.11	2.5e-15	1.5	1
with Šamanskii Technique	99	0.49	0.055	70	0.087	3e-30	1.6	0.086	9.6e-30	1.1	1
with Line Searches	99	0.34	0.033	30	0.11	0.00012	1.8	0.1	7.8e-06	1.5	1
with Occ. Line Searches	99	0.33	0.032	63	0.1	7.7e-16	1.8	0.11	2.5e-15	1.5	1
with Occ. LS & ŠT	99	0.54	0.058	71	0.091	3e-30	1.6	0.089	9.6e-30	1.1	1

TABLE 3. Results: 99 MMB models (starting guess: solution Dynare (QZ)).

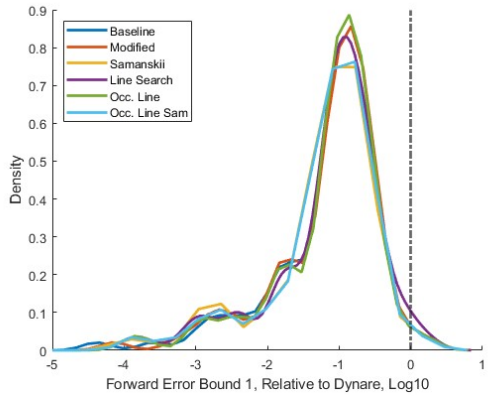
- For Dynare, refer to Adjemian, Bastani, Juillard, Mihoubi, Perendia, Ratto, and Villemot (2011).
- Run Time and foward errors relative to Dynare.
- Forward error 1 and 2 are the upper bounds for the true forward error, see (47).

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, Hu, and Zhang (2008), as they do not perform systematically better than the Baseline Newton algorithm along these dimensions. In sum, this experiment provides strong evidence that Newton-based methods can be used at minimal additional cost to refine the solutions provided by QZ.

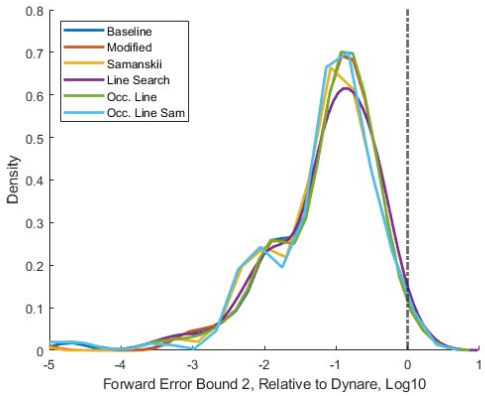
Figure 3 provides an overview of the entire distribution of forward errors, the upper row relative to those from Dynare's QZ method and the lower in absolute terms, using the different Newton-based methods presented here. Forward errors left of the vertical line are thus smaller than Dynare for both figures in the upper row. For both the first, figure 3a, and second, figure 3b, 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. Yet again, the various Newton methods - though now even the Modified algorithm that performed so poorly above - demonstrate no considerable differences when initializing with Dynare's QZ solution.

5.2. Policy Comparison Experiment

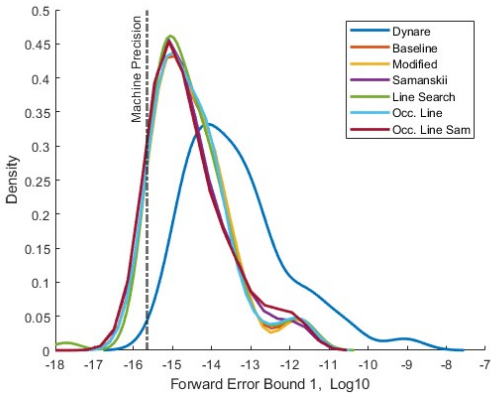
In this experiment we use the Newton-based algorithms to solve the model of Smets and Wouters (2007) iteratively for different parameterizations of the Taylor rule. The goal here is to explore whether solutions from previous, nearby parameterizations can be used to efficiently initialize the Newton methods 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



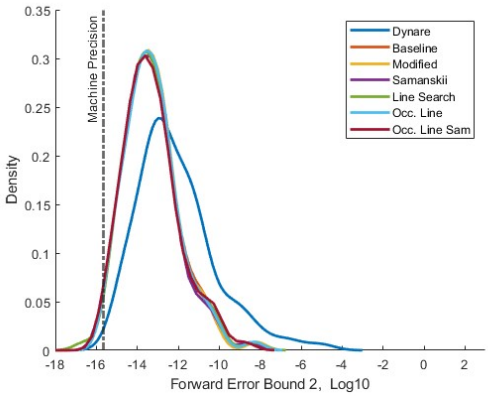
(A) Forward Error 1, Relative to Dynare



(B) Forward Error 2, Relative to Dynare



(C) Forward Error 1



(D) Forward Error 2

FIGURE 3. Distribution of forward error bounds relative to Dynare for the Macroeconomic Model Data Base (MMB)

Figures 3a, 3b 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)).

varying the size of the interval considered. We pick $r_\pi \in [1.5, 1.5\,(1+10^{-x})]$ 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 taking the solution under the previous parameterization as the initialization for the next iteration. A decrease in the spacing between the 100 grid points thus increases the precision of the starting guess, the solution from the previous parameterization.

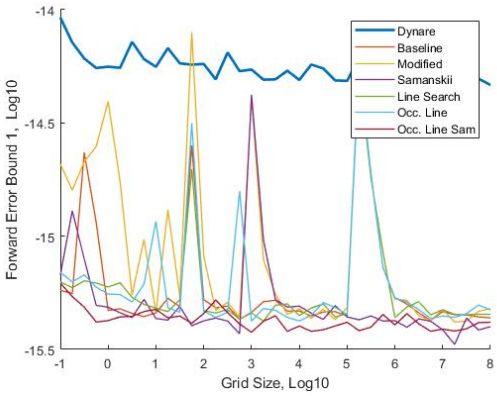
Table 4 shows that all algorithms are at the median more precise than Dynare, as measured by both upper bounds on the forward error, and roughly by an order of

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	0.90	0.76	1.19	0.06	0.01	0.86	0.07	0.03	0.22
	6	0.23	0.21	0.96	0.09	0.01	1.65	0.11	0.04	1.85
Modified Newton Method	-1	2.01	1.58	5.83	0.23	0.17	6.66	0.5	0.04	3.16
	6	0.23	0.21	2.41	0.11	0.01	1.68	0.11	0.05	1.90
with Šamanskii Technique	-1	1.13	0.75	1.41	0.06	0.01	7.73	0.08	0.03	2.21
	6	0.36	0.33	1.04	0.07	0.01	0.53	0.10	0.03	0.23
with Line Searches	-1	1.00	0.87	1.84	0.07	0.01	1.24	0.08	0.02	0.85
	6	0.25	0.23	1.02	0.09	0.01	0.92	0.11	0.05	1.83
with Occ. Line Searches	-1	1.02	0.85	2.07	0.09	0.01	1.14	0.08	0.02	2.51
	6	0.24	0.22	1.35	0.10	0.02	1.61	0.12	0.04	1.83
with Occ. LS & ŠT	-1	1.17	0.98	2.03	0.07	0.01	0.42	0.07	0.02	0.26
	6	0.37	0.34	1.38	0.08	0.01	0.36	0.09	0.03	0.20

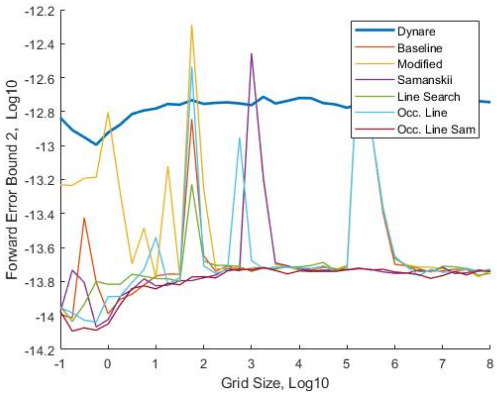
TABLE 4. Results: Smets-Wouters model.

- For Dynare, refer to Adjemian, Bastani, Juillard, Mihoubi, Perendia, Ratto, and Villemot (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).

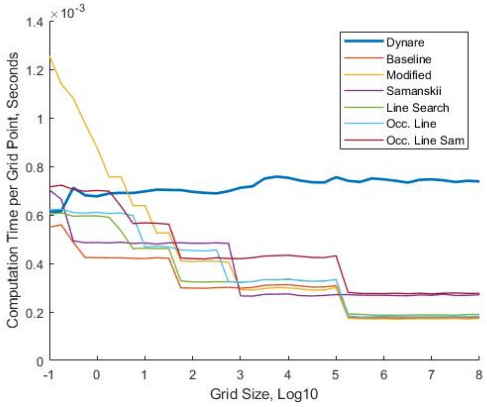
magnitude. A decrease in the spacing between the grid points decreases run time per grid point for all of the algorithms relative to Dynare's QZ method - which does not benefit from having a nearby solution to initialize its algorithm. The Baseline Newton method displays superior run-time performance 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.



(A) Forward Error 1, Relative to Dynare



(B) Forward Error 2, Relative to Dynare



(C) Computation Time per Grid Point

FIGURE 4. Forward Errors and Computation Time per Grid Point for different parameterizations of the model by Smets and Wouters (2007).

Figures 4a, 4b plot the upper forward error bounds 1 and 2 against the grid size, log10 scale on both axes. Figure 4c plots the computation per grid point against the number of grid points, log10 scale on both axes.

Figure 4 summarizes the experiment graphically. Figure 4c 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 posesses only linear convergence, thus benefiting mostly from a good starting guess. According to figures 4a, 4b, overall, all algorithms are more precise than Dynare. Thus, in summary, for iterative experiments like the one we have performed here, the iterative nature of Newton-based algorithms can be particularly advantageous, providing more accurate solutions at considerable computational savings.

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 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, exactly the dependence of convergence speed and properties on an initial matrix of Newton-based methods presents significant potential, particularly in iterative environments or when a solution refinement is sought. 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 (2022a) 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.

APPENDIX

6.1. Detailed Dynare Topology - Underlying Equations

Here we follow Villemot (2011) and develop in detail the matrices involved using the typology of variables from Dynare. In contrast to Villemot (2011), however, we develop the matrices explicitly, detailing the submatrices and their dimensions. While this first subsection contains nothing new, this alternate presentation might be of independent interest, hopefully increasing the approachability of the dimension reductions associated with the typology developed for Dynare. Additionally, it lays down the underlying typology needed to bring the matrix quadratic and elements of the Newton algorithms from the main text in line with Dynare. The first-order approximation of (1) at the steady state, where we only derive the homogenous - that is, in y_t - component necessary for the solution of the matrix quadratic equation (5), is

$$\mathbf{f}_{\mathbf{y}_{t+1}}\mathbf{y}_{t+1} + \mathbf{f}_{\mathbf{y}_t}\mathbf{y}_t + \mathbf{f}_{\mathbf{y}_{t+1}}\mathbf{y}_{t-1} = \mathbf{0}$$

The vector y_t is subdivided into $\mathbf{y}_t^s$, "static" variables with only nonzero derivatives at t, $\mathbf{y}_t^{--}$, "purely backward looking" variables with only nonzero derivatives at t and $t-1$, $\mathbf{y}_t^m$, "mixed" variables with nonzero derivatives at $t+1$, t, and $t-1$, and $\mathbf{y}_t^{++}$, "purely forward looking" variables with only nonzero derivatives at $t+1$ and t. The lengths of the subvectors in y_t satisfy the following equalities

$$n^d = n^{--} + n^m + n^{++}, \quad n^+ = n^m + n^{++}, \quad n^- = n^{--} + n^m, \quad n = n^s + n^d = n^s + n^{--} + n^m + n^{++}$$

where n^d is the number of dynamic variables, the sum of number of purely backward-looking, n^{--}, mixed n^m, and purely forward-looking variables, n^{++}. The number of forward-looking variables, n^+, is the sum of the number of mixed, n^m, and purely forward-looking variables, n^{++}, and the number of backward-looking variables, n^-, is the sum of the number of purely backward-looking, n^{--} and mixed variables n^m. Hence, the number of endogenous variables is the sum of the number of static, n^s, and dynamic variables, n^d, or the sum of the number of static, n^s, purely backward-looking, n^{--}, mixed n^m, and purely forward-looking variables, n^{++}. Arranging the matrices $\mathbf{f}_{\mathbf{y}_{t+1}}$, $\mathbf{f}_{\mathbf{y}_t}$, and $\mathbf{f}_{\mathbf{y}_{t-1}}$ accordingly gives

$$\begin{array}{c} \begin{array}{c} \\ n^s \\ n^{--} \\ n^m \\ n^{++} \end{array} \begin{array}{c} \begin{array}{cccc} n^s & n^{--} & n^m & n^{++} \end{array} \\ \left[\begin{array}{cc|cc} \mathbf{0} & \mathbf{0} & & \\ \mathbf{0} & \mathbf{0} & & \\ \mathbf{0} & \mathbf{0} & & \\ \mathbf{0} & \mathbf{0} & & \end{array}\right] \end{array} \underset{n\times n^+}{\mathbf{f}_{\mathbf{y}^+}} \begin{array}{c} \begin{array}{c} 1 \end{array} \\ \left[\begin{array}{c|c} n^s & \mathbf{y}^s_{t+1} \\ n^{--} & \mathbf{y}^{--}_{t+1} \\ n^m & \mathbf{y}^m_{t+1} \\ n^{++} & \mathbf{y}^{++}_{t+1} \end{array}\right] \end{array} + \begin{array}{c} \begin{array}{cccc} n^s & n^{--} & n^m & n^{++} \end{array} \\ \left[\begin{array}{cc|cc} & & & \\ & & & \\ & & & \\ & & & \end{array}\right] \end{array} \underset{n\times n}{\mathbf{f}_{\mathbf{y}^0}} \begin{array}{c} \begin{array}{c} 1 \end{array} \\ \left[\begin{array}{c|c} n^s & \mathbf{y}^s_{t} \\ n^{--} & \mathbf{y}^{--}_{t} \\ n^m & \mathbf{y}^m_{t} \\ n^{++} & \mathbf{y}^{++}_{t} \end{array}\right] \end{array} \\ + \begin{array}{c} \\ n^s \\ n^{--} \\ n^m \\ n^{++} \end{array} \begin{array}{c} \begin{array}{cccc} n^s & n^{--} & n^m & n^{++} \end{array} \\ \left[\begin{array}{cc|cc} \mathbf{0} & & \mathbf{0} & \\ \mathbf{0} & & \mathbf{0} & \\ \mathbf{0} & & \mathbf{0} & \\ \mathbf{0} & & \mathbf{0} & \end{array}\right] \end{array} \underset{n\times n^-}{\mathbf{f}_{\mathbf{y}^-}} \begin{array}{c} \begin{array}{c} 1 \end{array} \\ \left[\begin{array}{c|c} n^s & \mathbf{y}^s_{t-1} \\ n^{--} & \mathbf{y}^{--}_{t-1} \\ n^m & \mathbf{y}^m_{t-1} \\ n^{++} & \mathbf{y}^{++}_{t-1} \end{array}\right] \end{array} = \underset{n\times 1}{\mathbf{0}} \end{array}$$

Subdividing the matrix into the columns associated with static variables and the remaining variables, also referred to as "dynamic" variables $\mathbf{y}^d_t$ - having nonzero at $t+1$ and/or $t-1$ yields

$$\mathbf{f}_{\mathbf{y}^0} = \begin{array}{c} \begin{array}{cc} n^s & n^{--}\ n^m\ n^{++} \end{array} \\ \left[\begin{array}{c|c} \underset{n\times n^s}{\mathbf{S}} & \underset{n\times n^d}{\mathbf{S}^-} \end{array}\right] \end{array}$$

Performing a QR decomposition on $\mathbf{S}$, $\underset{n\times n^s}{\mathbf{S}} = \underset{n\times n}{\mathbf{Q}}\ \underset{n\times n^s}{\mathbf{R}}$, where $\mathbf{R} = \begin{array}{c} \begin{array}{c} n^s \end{array} \\ \left[\begin{array}{c} \breve{\mathbf{A}}^{0s} \\ \hline \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{array}\right] \end{array}$ and premultiplying the system of equations with the inverse of the unitary Q, Q^*, gives

$$\mathbf{Q}^*\mathbf{f}_{\mathbf{y}_{t+1}}\mathbf{y}_{t+1} + \mathbf{Q}^*\mathbf{f}_{\mathbf{y}_t}\mathbf{y}_t + \mathbf{Q}^*\mathbf{f}_{\mathbf{y}_{t+1}}\mathbf{y}_{t+1} = \mathbf{0}$$

$$\begin{array}{c} \\ n^s \\ n^{--} \\ n^m \\ n^{++} \end{array}\!\!\begin{array}{c} \begin{array}{cc} n^s & n^{--} \end{array} \\ \left[\begin{array}{cc} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{array}\right. \end{array}\!\!\begin{array}{c} n^m \quad n^{++} \\ \left.\begin{array}{c} \\ \\ \mathbf{A}^+_{n\times n^+} \\ \\ \end{array}\right] \end{array}\!\!\begin{array}{c} 1 \\ \left[\begin{array}{c} \mathbf{y}^s_{t+1} \\ \mathbf{y}^{--}_{t+1} \\ \mathbf{y}^m_{t+1} \\ \mathbf{y}^{++}_{t+1} \end{array}\right] \end{array} + \begin{array}{c} \\ n^s \\ n^{--} \\ n^m \\ n^{++} \end{array}\!\!\begin{array}{c} n^s \quad n^{--} \quad n^m \quad n^{++} \\ \left[\begin{array}{cccc} & & & \\ & & & \\ & & \mathbf{A}^0_{n\times n} & \\ & & & \end{array}\right] \end{array}\!\!\begin{array}{c} 1 \\ \left[\begin{array}{c} \mathbf{y}^s_t \\ \mathbf{y}^{--}_t \\ \mathbf{y}^m_t \\ \mathbf{y}^{++}_t \end{array}\right] \end{array}$$

$$+ \begin{array}{c} \\ n^s \\ n^{--} \\ n^m \\ n^{++} \end{array}\!\!\begin{array}{c} n^s \quad n^{--} \quad n^m \quad n^{++} \\ \left[\begin{array}{cccc} \mathbf{0} & & \mathbf{0} & \\ \mathbf{0} & & \mathbf{0} & \\ \mathbf{0} & \mathbf{A}^-_{n\times n^-} & \mathbf{0} & \\ \mathbf{0} & & \mathbf{0} & \end{array}\right] \end{array}\!\!\begin{array}{c} 1 \\ \left[\begin{array}{c} \mathbf{y}^s_{t-1} \\ \mathbf{y}^{--}_{t-1} \\ \mathbf{y}^m_{t-1} \\ \mathbf{y}^{++}_{t-1} \end{array}\right] \end{array} = \underset{n\times 1}{\mathbf{0}}$$

where $\mathbf{A}^+ = \mathbf{Q}^*\mathbf{f}_{\mathbf{y}^+}$, $\mathbf{A}^0 = \mathbf{Q}^*\mathbf{f}_{\mathbf{y}^0} = \underset{n\times n^s}{\left[\mathbf{Q}^*\mathbf{S}\right.} \quad \underset{n\times n^d}{\left.\mathbf{Q}^*\mathbf{S}^-\right]} = \begin{array}{c} \\ n^s \\ n^{--} \\ n^m \\ n^{++} \end{array}\!\!\begin{array}{c} n^s \quad n^{--} \quad n^m \quad n^{++} \\ \left[\begin{array}{cccc} \breve{\mathbf{A}}^{0s} & & & \\ \mathbf{0} & & & \\ \mathbf{0} & & \mathbf{Q}^*\mathbf{S}^-_{n\times n^d} & \\ \mathbf{0} & & & \end{array}\right] \end{array}$, and $\mathbf{A}^- = \mathbf{Q}^*\mathbf{f}_{\mathbf{y}^-}$. Subdividing the system of equations in accordance with the QR decomposition yields

$$\begin{array}{c} \\ n^s \\ n^{--} \\ n^m \\ n^{++} \end{array}\!\!\begin{array}{c} n^s \quad n^{--} \quad n^m \quad n^{++} \\ \left[\begin{array}{cccc} \mathbf{0} & \mathbf{0} & \breve{\mathbf{A}}^+_{n^s\times n^+} & \\ \mathbf{0} & \mathbf{0} & & \\ \mathbf{0} & \mathbf{0} & \tilde{\mathbf{A}}^+_{n^d\times n^+} & \\ \mathbf{0} & \mathbf{0} & & \end{array}\right] \end{array}\!\!\begin{array}{c} 1 \\ \left[\begin{array}{c} \mathbf{y}^s_{t+1} \\ \mathbf{y}^{--}_{t+1} \\ \mathbf{y}^m_{t+1} \\ \mathbf{y}^{++}_{t+1} \end{array}\right] \end{array} + \begin{array}{c} \\ n^s \\ n^{--} \\ n^m \\ n^{++} \end{array}\!\!\begin{array}{c} n^s \quad n^{--} \quad n^m \quad n^{++} \\ \left[\begin{array}{cccc} \breve{\mathbf{A}}^{0s} & & \breve{\mathbf{A}}^{0d}_{n^s\times n^d} & \\ \mathbf{0} & & & \\ \mathbf{0} & & \tilde{\mathbf{A}}^0_{n^d\times n^d} & \\ \mathbf{0} & & & \end{array}\right] \end{array}\!\!\begin{array}{c} 1 \\ \left[\begin{array}{c} \mathbf{y}^s_t \\ \mathbf{y}^{--}_t \\ \mathbf{y}^m_t \\ \mathbf{y}^{++}_t \end{array}\right] \end{array}$$

$$+ \begin{array}{c} \\ n^s \\ n^{--} \\ n^m \\ n^{++} \end{array}\!\!\begin{array}{c} n^s \quad n^{--} \quad n^m \quad n^{++} \\ \left[\begin{array}{cccc} \mathbf{0} & \breve{\mathbf{A}}^-_{n^s\times n^-} & \mathbf{0} & \\ \mathbf{0} & & \mathbf{0} & \\ \mathbf{0} & \tilde{\mathbf{A}}^-_{n^d\times n^-} & \mathbf{0} & \\ \mathbf{0} & & \mathbf{0} & \end{array}\right] \end{array}\!\!\begin{array}{c} 1 \\ \left[\begin{array}{c} \mathbf{y}^s_{t-1} \\ \mathbf{y}^{--}_{t-1} \\ \mathbf{y}^m_{t-1} \\ \mathbf{y}^{++}_{t-1} \end{array}\right] \end{array} = \underset{n\times 1}{\mathbf{0}}$$

The matrix $\tilde{\mathbf{A}}^0$ is

$$\underset{n^d \times n^d}{\tilde{\mathbf{A}}^0} \underset{n^d \times 1}{\mathbf{y}_t^d} = \begin{bmatrix} \underset{n^d \times n^{--}}{\tilde{\mathbf{A}}^{0--}} & \underset{n^d \times n^m}{\tilde{\mathbf{A}}^{0m}} & \underset{n^d \times n^{++}}{\tilde{\mathbf{A}}^{0++}} \end{bmatrix} \begin{bmatrix} \underset{n^{--} \times 1}{\mathbf{y}_t^{--}} \\ \underset{n^m \times 1}{\mathbf{y}_t^m} \\ \underset{n^{++} \times 1}{\mathbf{y}_t^{++}} \end{bmatrix}$$

The vectors for forward and backward-looking variables can be assembled depending on how the mixed variables are assigned according to either the first or second equality in the following

$$\begin{cases} \overset{1}{=} \underbrace{\begin{bmatrix} \underset{n^d \times n^{--}}{\tilde{\mathbf{A}}^{0--}} & \underset{n^d \times n^m}{\tilde{\mathbf{A}}^{0m}} \end{bmatrix}}_{\underset{n^d \times n^-}{\tilde{\mathbf{A}}^{0-}}} \underbrace{\begin{bmatrix} \underset{n^{--} \times 1}{\mathbf{y}_t^{--}} \\ \underset{n^m \times 1}{\mathbf{y}_t^m} \end{bmatrix}}_{\underset{n^- \times 1}{\mathbf{y}_t^-}} + \underbrace{\begin{bmatrix} \underset{n^d \times n^m}{\mathbf{0}} & \underset{n^d \times n^{++}}{\tilde{\mathbf{A}}^{0++}} \end{bmatrix}}_{\underset{n^d \times n^+}{\tilde{\mathbf{A}}^{0+}}} \underbrace{\begin{bmatrix} \underset{n^m \times 1}{\mathbf{y}_t^m} \\ \underset{n^{++} \times 1}{\mathbf{y}_t^{++}} \end{bmatrix}}_{\underset{n^+ \times 1}{\mathbf{y}_t^+}} \\ \overset{2}{=} \underbrace{\begin{bmatrix} \underset{n^d \times n^{--}}{\tilde{\mathbf{A}}^{0--}} & \underset{n^d \times n^m}{\mathbf{0}} \end{bmatrix}}_{\underset{n^d \times n^-}{\tilde{\mathbf{A}}^{0-}}} \underbrace{\begin{bmatrix} \underset{n^{--} \times 1}{\mathbf{y}_t^{--}} \\ \underset{n^m \times 1}{\mathbf{y}_t^m} \end{bmatrix}}_{\underset{n^- \times 1}{\mathbf{y}_t^-}} + \underbrace{\begin{bmatrix} \underset{n^d \times n^m}{\tilde{\mathbf{A}}^{0m}} & \underset{n^d \times n^{++}}{\tilde{\mathbf{A}}^{0++}} \end{bmatrix}}_{\underset{n^d \times n^+}{\tilde{\mathbf{A}}^{0+}}} \underbrace{\begin{bmatrix} \underset{n^m \times 1}{\mathbf{y}_t^m} \\ \underset{n^{++} \times 1}{\mathbf{y}_t^{++}} \end{bmatrix}}_{\underset{n^+ \times 1}{\mathbf{y}_t^+}} \end{cases}$$

The mixed variables can then be selected out of the vectors of forward and backward-looking variables via

$$\mathbf{y}_t^m = \mathbf{y}_t^m$$

$$\underbrace{\begin{bmatrix} \underset{n^m \times n^{--}}{\mathbf{0}} & \underset{n^m \times n^m}{\mathbf{I}} \end{bmatrix}}_{\underset{n^m \times n^-}{\mathbf{I}^-}} \underbrace{\begin{bmatrix} \underset{n^{--} \times 1}{\mathbf{y}_t^{--}} \\ \underset{n^m \times 1}{\mathbf{y}_t^m} \end{bmatrix}}_{\underset{n^- \times 1}{\mathbf{y}_t^-}} = \underbrace{\begin{bmatrix} \underset{n^m \times n^m}{\mathbf{I}} & \underset{n^m \times n^{++}}{\mathbf{0}} \end{bmatrix}}_{\underset{n^m \times n^+}{\mathbf{I}^+}} \underbrace{\begin{bmatrix} \underset{n^m \times 1}{\mathbf{y}_t^m} \\ \underset{n^{++} \times 1}{\mathbf{y}_t^{++}} \end{bmatrix}}_{\underset{n^+ \times 1}{\mathbf{y}_t^+}}$$

$$\underset{n^m \times n^-}{\mathbf{I}^-} \underset{n^- \times 1}{\mathbf{y}_t^-} = \underset{n^m \times n^+}{\mathbf{I}^+} \underset{n^+ \times 1}{\mathbf{y}_t^+}$$

These are the "selection" matrices of Villemot (2011).

6.2. Detailed Dynare Topology - Matrix Quadratic

We now continue with the topology from Dynare and apply it to the underlying matrix quadratic. The transition matrix, P, from (4) that solves the matrix

equation (13) can be subdivided in accordance to Dynare's typology as

$$\mathbf{P} = \begin{array}{c} \\ n^s \\ n^{--} \\ n^m \\ n^{++} \end{array} \begin{array}{c} \begin{array}{cccc} n^s & n^{--} & n^m & n^{++} \end{array} \\ \begin{bmatrix} \mathbf{P}_{s,s} & \mathbf{P}_{s,--} & \mathbf{P}_{s,m} & \mathbf{P}_{s,++} \\ \mathbf{P}_{--,s} & \mathbf{P}_{--,--} & \mathbf{P}_{--,m} & \mathbf{P}_{--,++} \\ \mathbf{P}_{m,s} & \mathbf{P}_{m,--} & \mathbf{P}_{m,m} & \mathbf{P}_{m,++} \\ \mathbf{P}_{++,s} & \mathbf{P}_{++,--} & \mathbf{P}_{++,m} & \mathbf{P}_{++,++} \end{bmatrix} \end{array} = \begin{array}{c} n \\ \begin{bmatrix} \mathbf{P}_{\bullet,s} & \mathbf{P}_{\bullet,--} & \mathbf{P}_{\bullet,m} & \mathbf{P}_{\bullet,++} \end{bmatrix} \end{array} = \begin{array}{c} \\ n^s \\ n^{--} \\ n^m \\ n^{++} \end{array} \begin{array}{c} \begin{array}{c} n \end{array} \\ \begin{bmatrix} \mathbf{P}_{s,\bullet} \\ \mathbf{P}_{--,\bullet} \\ \mathbf{P}_{m,\bullet} \\ \mathbf{P}_{++,\bullet} \end{bmatrix} \end{array}$$

The matrix quadratic can be expressed as

$$\begin{aligned} \underset{n\times n}{\mathbf{M}}(\underset{n\times n}{\mathbf{P}}) &= \underset{n\times n}{\mathbf{A}}\mathbf{P}^2 + \underset{n\times n}{\mathbf{B}}\mathbf{P} + \underset{n\times n}{\mathbf{C}} \\ &= \underbrace{(\mathbf{AP}+\mathbf{B})}_{\equiv\mathbf{G}}\mathbf{P} + \mathbf{C} \end{aligned}$$

For a solvent P of the matrix quadratic, taking the structure of C from the Dynare typology above into account yields

$$\mathbf{M}(\mathbf{P}) = 0 = \mathbf{GP} + \mathbf{C}$$

$$= \mathbf{G} \begin{array}{c} n \\ \begin{bmatrix} \mathbf{P}_{\bullet,s} & \mathbf{P}_{\bullet,--} & \mathbf{P}_{\bullet,m} & \mathbf{P}_{\bullet,++} \end{bmatrix} \end{array} + \begin{array}{c} \\ n^s \\ n^{--} \\ n^m \\ n^{++} \end{array} \begin{array}{c} \begin{array}{ccc} n^s & n^{--} & n^m & n^{++} \end{array} \\ \begin{bmatrix} \mathbf{0} & \underset{n^s\times n^-}{\breve{\mathbf{A}}^-} & \mathbf{0} \\ \mathbf{0} & & \mathbf{0} \\ \mathbf{0} & \underset{n^d\times n^-}{\tilde{\mathbf{A}}^-} & \mathbf{0} \\ \mathbf{0} & & \mathbf{0} \end{bmatrix} \end{array}$$

From corollary 4.5 of Lan and Meyer-Gohde (2014), $\mathbf{G}$ has full rank if P is the unique solvent of $M(P)$ stable with respect to the closed unit circle, hence the columns of P associated with nonzero columns in C, the static and forward-looking variables are zero $\rightarrow \mathbf{P}_{\bullet,s} = \underset{n\times n^s}{\mathbf{0}}$, $\mathbf{P}_{\bullet,++} = \underset{n\times n^{++}}{\mathbf{0}}$, whence P is

$$\mathbf{P} = \begin{array}{c} n \\ \begin{bmatrix} \mathbf{0} & \mathbf{P}_{\bullet,--} & \mathbf{P}_{\bullet,m} & \mathbf{0} \end{bmatrix} \end{array} \begin{array}{c} \begin{array}{cccc} n^s & n^{--} & n^m & n^{++} \end{array} \end{array}$$

and $\mathbf{M}(\mathbf{P}) = \begin{bmatrix} \underset{n\times n^s}{\mathbf{0}} & \underset{n\times n^{--}}{\mathbf{M}(\mathbf{P})^{--}} & \underset{n\times n^m}{\mathbf{M}(\mathbf{P})^m} & \underset{n\times n^{++}}{\mathbf{0}} \end{bmatrix}$.
Consequentially, the first n^s rows of the matrix quadratic are

$$\underset{n^s\times n^+}{\breve{\mathbf{A}}^+} \begin{array}{c} \\ n^m \\ n^{++} \end{array} \begin{array}{c} \begin{array}{cc} n^{--} & n^m \end{array} \\ \begin{bmatrix} \mathbf{P}_{m,--} & \mathbf{P}_{m,m} \\ \mathbf{P}_{++,--} & \mathbf{P}_{++,m} \end{bmatrix} \end{array} \begin{array}{c} n^{--} \\ n^m \end{array} \begin{bmatrix} \mathbf{P}_{--,--} & \mathbf{P}_{--,m} \\ \mathbf{P}_{m,--} & \mathbf{P}_{m,m} \end{bmatrix} + \underset{n^s\times n^d}{\breve{\mathbf{A}}^{0d}} \begin{array}{c} \\ n^{--} \\ n^m \\ n^{++} \end{array} \begin{array}{c} \begin{array}{cc} n^{--} & n^m \end{array} \\ \begin{bmatrix} \mathbf{P}_{--,--} & \mathbf{P}_{--,m} \\ \mathbf{P}_{m,--} & \mathbf{P}_{m,m} \\ \mathbf{P}_{++,--} & \mathbf{P}_{++,m} \end{bmatrix} \end{array}$$

$$+ \underset{n^s \times n^-}{\breve{\mathbf{A}}^-} + \underset{n^s \times n^s}{\breve{\mathbf{A}}^{0s}} \overset{\begin{matrix} n^{--} & n^m \end{matrix}}{n^s \begin{bmatrix} \mathbf{P}_{s,--} & \mathbf{P}_{s,m} \end{bmatrix}} = \underset{n^s \times n^-}{\mathbf{0}}$$

Given $\overset{n}{\begin{matrix} n^{--} \\ n^m \\ n^{++} \end{matrix}} \begin{bmatrix} \mathbf{P}_{--,\bullet} \\ \mathbf{P}_{m,\bullet} \\ \mathbf{P}_{++,\bullet} \end{bmatrix}$, $\overset{\begin{matrix} n^{--} & n^m \end{matrix}}{n^s \begin{bmatrix} \mathbf{P}_{s,--} & \mathbf{P}_{s,m} \end{bmatrix}}$ solves

$$\overset{\begin{matrix} n^{--} & n^m \end{matrix}}{n^s \begin{bmatrix} \mathbf{P}_{s,--} & \mathbf{P}_{s,m} \end{bmatrix}} = - \left[\underset{n^s \times n^s}{\breve{\mathbf{A}}^{0s}} \right]^{-1} \left(\underset{n^s \times n^+}{\breve{\mathbf{A}}^+} \overset{\begin{matrix} n^{--} & n^m \end{matrix}}{\begin{matrix} n^m \\ n^{++} \end{matrix} \begin{bmatrix} \mathbf{P}_{m,--} & \mathbf{P}_{m,m} \\ \mathbf{P}_{++,--} & \mathbf{P}_{++,m} \end{bmatrix}} \overset{\begin{matrix} n^{--} & n^m \end{matrix}}{\begin{matrix} n^{--} \\ n^m \end{matrix} \begin{bmatrix} \mathbf{P}_{--,--} & \mathbf{P}_{--,m} \\ \mathbf{P}_{m,--} & \mathbf{P}_{m,m} \end{bmatrix}} \right.$$

$$\left. + \underset{n^s \times n^d}{\breve{\mathbf{A}}^{0d}} \overset{\begin{matrix} n^{--} & n^m \end{matrix}}{\begin{matrix} n^{--} \\ n^m \\ n^{++} \end{matrix} \begin{bmatrix} \mathbf{P}_{--,--} & \mathbf{P}_{--,m} \\ \mathbf{P}_{m,--} & \mathbf{P}_{m,m} \\ \mathbf{P}_{++,--} & \mathbf{P}_{++,m} \end{bmatrix}} + \underset{n^s \times n^-}{\breve{\mathbf{A}}^-} \right)$$

and the first n^s rows of P are $\underset{n^s \times n}{\mathbf{P}_{s,\bullet}} = \overset{\begin{matrix} n^s & n^{--} & n^m & n^{++} \end{matrix}}{n^s \begin{bmatrix} \mathbf{0} & \mathbf{P}_{s,--} & \mathbf{P}_{s,m} & \mathbf{0} \end{bmatrix}}$.

The last n^d columns and rows of P solve the reduced matrix quadratic equation

$$\overset{\begin{matrix} n^{--} & n^m & n^{++} \end{matrix}}{\begin{matrix} n^{--} \\ n^m \\ n^{++} \end{matrix} \begin{bmatrix} \mathbf{0} & & \\ \mathbf{0} & \underset{n^d \times n^+}{\tilde{\mathbf{A}}^+} & \\ \mathbf{0} & & \end{bmatrix}} \underbrace{\overset{\begin{matrix} n^{--} & n^m & n^{++} \end{matrix}}{\begin{matrix} n^{--} \\ n^m \\ n^{++} \end{matrix} \begin{bmatrix} \mathbf{P}_{--,--} & \mathbf{P}_{--,m} & \mathbf{P}_{--,++} \\ \mathbf{P}_{m,--} & \mathbf{P}_{m,m} & \mathbf{P}_{m,++} \\ \mathbf{P}_{++,--} & \mathbf{P}_{++,m} & \mathbf{P}_{++,++} \end{bmatrix}}}_{\underset{n^d \times n^d}{\tilde{\mathbf{P}}}} \cdot \underset{n^d \times n^d}{\tilde{\mathbf{P}}} + \underset{n^d \times n^d}{\tilde{\mathbf{A}}^0} \underset{n^d \times n^d}{\tilde{\mathbf{P}}}$$

$$+ \overset{\begin{matrix} n^{--} & n^m & n^{++} \end{matrix}}{\begin{matrix} n^{--} \\ n^m \\ n^{++} \end{matrix} \begin{bmatrix} & & \mathbf{0} \\ & \underset{n^d \times n^-}{\tilde{\mathbf{A}}^-} & \mathbf{0} \\ & & \mathbf{0} \end{bmatrix}}$$

$$\underset{n^d \times n^d}{=\tilde{\mathbf{M}}(\tilde{\mathbf{P}})} = n^d \begin{bmatrix} \overset{n^{--}}{\tilde{\mathbf{M}}(\tilde{\mathbf{P}})^{--}} & \overset{n^m}{\tilde{\mathbf{M}}(\tilde{\mathbf{P}})^m} & \overset{n^{++}}{\mathbf{0}} \end{bmatrix} = \underset{n^d \times n^d}{\mathbf{0}}$$

6.3. Detailed Dynare Topology - Newton Step

The Newton-based methods in the main text all require solving a Sylvester equation for the iterative Newton step, $d\mathbf{P}$. This can be broken down using the typology from above as follows:

$$\mathbf{A} \cdot d\mathbf{P} \cdot \mathbf{P} + (\mathbf{AP} + \mathbf{B}) d\mathbf{P} + \mathbf{M}(\mathbf{P}) = \mathbf{0}$$

As was shown above, $\underset{n \times n}{\mathbf{M}(\mathbf{P})} = n \begin{bmatrix} \overset{n^s}{\mathbf{0}} & \overset{n^{--}}{\mathbf{M}(\mathbf{P})^{\bullet,--}} & \overset{n^m}{\mathbf{M}(\mathbf{P})^{\bullet,m}} & \overset{n^{++}}{\mathbf{0}} \end{bmatrix}$ and $\mathbf{P} = n \begin{bmatrix} \overset{n^s}{\mathbf{0}} & \overset{n^{--}}{\mathbf{P}_{\bullet,--}} & \overset{n^m}{\mathbf{P}_{\bullet,m}} & \overset{n^{++}}{\mathbf{0}} \end{bmatrix}$, hence it follows that

$$d\mathbf{P} = n \begin{bmatrix} \overset{n^s}{\mathbf{0}} & \overset{n^{--}}{d\mathbf{P}_{\bullet,--}} & \overset{n^m}{d\mathbf{P}_{\bullet,m}} & \overset{n^{++}}{\mathbf{0}} \end{bmatrix}$$

To see this, develop the expression $\mathbf{A} \cdot d\mathbf{P} \cdot \mathbf{P}$

$$d\mathbf{PP} = n \begin{bmatrix} \overset{n^s}{\mathbf{0}} & \overset{n^{--}}{d\mathbf{PP}_{\bullet,--}} & \overset{n^m}{d\mathbf{PP}_{\bullet,m}} & \overset{n^{++}}{\mathbf{0}} \end{bmatrix}$$

$$\mathbf{A}d\mathbf{PP} = n \begin{bmatrix} \overset{n^s}{\mathbf{0}} & \overset{n^{--}}{\mathbf{A}d\mathbf{PP}_{\bullet,--}} & \overset{n^m}{\mathbf{A}d\mathbf{PP}_{\bullet,m}} & \overset{n^{++}}{\mathbf{0}} \end{bmatrix}$$

The first and last block columns give

$$n \begin{bmatrix} \overset{n^s}{\mathbf{0}} & \overset{n^{++}}{\mathbf{0}} \end{bmatrix} + (\mathbf{AP} + \mathbf{B}) \, n \begin{bmatrix} \overset{n^s}{d\mathbf{P}^{\bullet,s}} & \overset{n^{++}}{d\mathbf{P}^{\bullet,++}} \end{bmatrix} + n \begin{bmatrix} \overset{n^s}{\mathbf{0}} & \overset{n^{++}}{\mathbf{0}} \end{bmatrix} = n \begin{bmatrix} \overset{n^s}{\mathbf{0}} & \overset{n^{++}}{\mathbf{0}} \end{bmatrix}$$

and $\mathbf{AP} + \mathbf{B} = \mathbf{G}$ is full rank (see above), $d\mathbf{P}^{\bullet,s}$ and $d\mathbf{P}^{\bullet,++}$ are zero matrices.

As the first block columns of $\mathbf{A}$ and $d\mathbf{P}$ are zero and the first block column of $\mathbf{B} = \begin{array}{c} \overset{n^s}{} \\ \begin{matrix} n^s \\ n^{--} \\ n^m \\ n^{--} \end{matrix} \end{array} \begin{bmatrix} \check{\mathbf{A}}^{0s} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}$, $d\mathbf{P}_{s,\bullet}$ is given by $d\mathbf{P}_{s,\bullet} = n^s \begin{bmatrix} \overset{n^s}{\mathbf{0}} & \overset{n^{--}}{d\mathbf{P}^{s,--}} & \overset{n^m}{d\mathbf{P}^{s,m}} & \overset{n^{++}}{\mathbf{0}} \end{bmatrix}$,

where

$$
\begin{array}{c} n^{--} \quad n^{m} \\ n^{s}\left[\begin{array}{c|c} d\mathbf{P}^{s,--} & d\mathbf{P}^{s,m} \end{array}\right] \end{array} = -\underset{n^s \times n^s}{\left[\check{\mathbf{A}}^{0s}\right]^{-1}} \left(\begin{array}{c} n^{--} \quad n^{m} \\ n^{s}\left[\begin{array}{c|c} \mathbf{M}^{s,--}(\mathbf{P}) & \mathbf{M}^{s,m}(\mathbf{P}) \end{array}\right] \end{array} + \underset{n^s \times n^d}{\check{\mathbf{A}}^{0d}} \begin{array}{c} n^{--} \quad n^{m} \\ n^{m}\left[\begin{array}{c|c} d\mathbf{P}_{--,--} & d\mathbf{P}_{--,m} \\ \hline d\mathbf{P}_{m,--} & d\mathbf{P}_{m,m} \end{array}\right] \\ n^{++}\left[\begin{array}{c|c} d\mathbf{P}_{++,--} & d\mathbf{P}_{++,m} \end{array}\right] \end{array} \right.
$$

$$
+ \underset{n^d \times n^+}{\check{\mathbf{A}}^{+}} \left\{ \begin{array}{c} n^{--} \quad n^{m} \\ n^{m}\left[\begin{array}{c|c} d\mathbf{P}_{m,--} & d\mathbf{P}_{m,m} \end{array}\right] \\ n^{++}\left[\begin{array}{c|c} d\mathbf{P}_{++,--} & d\mathbf{P}_{++,m} \end{array}\right] \end{array} \begin{array}{c} n^{--} \quad n^{m} \\ n^{--}\left[\begin{array}{c|c} \mathbf{P}_{--,--} & \mathbf{P}_{--,m} \\ \hline \mathbf{P}_{m,--} & \mathbf{P}_{m,m} \end{array}\right] \end{array} \right.
$$

$$
\left. + \begin{array}{c} n^{--} \quad n^{m} \\ n^{m}\left[\begin{array}{c|c} \mathbf{P}_{m,--} & \mathbf{P}_{m,m} \end{array}\right] \\ n^{++}\left[\begin{array}{c|c} \mathbf{P}_{++,--} & \mathbf{P}_{++,m} \end{array}\right] \end{array} \begin{array}{c} n^{--} \quad n^{m} \\ n^{--}\left[\begin{array}{c|c} d\mathbf{P}_{--,--} & d\mathbf{P}_{--,m} \\ \hline d\mathbf{P}_{m,--} & d\mathbf{P}_{m,m} \end{array}\right] \end{array} \right\}
$$

given a solution for $\begin{array}{c} n^{--} \quad n^{m} \\ n^{--}\left[\begin{array}{c|c} d\mathbf{P}_{--,--} & d\mathbf{P}_{--,m} \\ \hline d\mathbf{P}_{m,--} & d\mathbf{P}_{m,m} \\ \hline d\mathbf{P}_{++,--} & d\mathbf{P}_{++,m} \end{array}\right] \\ n^{m} \\ n^{++} \end{array}$

Hence the remaining equations are (where zero columns of $\mathbf{P}$, $d\mathbf{P}$, $\mathbf{M}(\mathbf{P})$ have been eliminated where appropriate)

$$
\underset{n^d \times n^+}{\tilde{\mathbf{A}}^{+}} \begin{array}{c} n^{--} \quad n^{m} \\ n^{m}\left[\begin{array}{c|c} d\mathbf{P}_{m,--} & d\mathbf{P}_{m,m} \end{array}\right] \\ n^{++}\left[\begin{array}{c|c} d\mathbf{P}_{++,--} & d\mathbf{P}_{++,m} \end{array}\right] \end{array} \begin{array}{c} n^{--} \quad n^{m} \\ n^{--}\left[\begin{array}{c|c} \mathbf{P}_{--,--} & \mathbf{P}_{--,m} \\ \hline \mathbf{P}_{m,--} & \mathbf{P}_{m,m} \end{array}\right] \end{array}
$$

$$
+ \left(\begin{array}{c} n^{--} \quad n^{m} \quad n^{++} \\ n^{m}\left[\begin{array}{c|c|c} \mathbf{P}_{m,--} & \mathbf{P}_{m,m} & \mathbf{0} \end{array}\right] \\ n^{++}\left[\begin{array}{c|c|c} \mathbf{P}_{++,--} & \mathbf{P}_{++,m} & \mathbf{0} \end{array}\right] \end{array} + \underset{n^d \times n^d}{\tilde{\mathbf{A}}^{0}} \right) \begin{array}{c} n^{--} \quad n^{m} \\ n^{--}\left[\begin{array}{c|c} d\mathbf{P}_{--,--} & d\mathbf{P}_{--,m} \\ \hline d\mathbf{P}_{m,--} & d\mathbf{P}_{m,m} \\ \hline d\mathbf{P}_{++,--} & d\mathbf{P}_{++,m} \end{array}\right] \\ n^{m} \\ n^{++} \end{array}
$$

$$
\begin{aligned}
&\overset{n^{--}}{} \qquad \overset{n^{m}}{} \\
&+n^{d}\left[\begin{array}{c|c} \tilde{\mathbf{M}}(\tilde{\mathbf{P}})^{--} & \tilde{\mathbf{M}}(\tilde{\mathbf{P}})^{m} \end{array}\right] \\
&\overset{n^{--}}{} \quad \overset{n^{m}}{} \\
=&n^{d}\left[\begin{array}{c|c} \mathbf{0} & \mathbf{0} \end{array}\right]
\end{aligned}
$$

Defining $\underset{n^d\times n^d}{\tilde{\mathbf{A}}^0} = n^d\left[\begin{array}{c|c|c} \tilde{\mathbf{A}}^{0--} & \tilde{\mathbf{A}}^{0m} & \tilde{\mathbf{A}}^{0++} \end{array}\right]$, the foregoing is

$$
\begin{aligned}
&\underbrace{\left(\begin{array}{l} \overset{n^{--}}{} \\ \underset{n^d\times n^+}{\tilde{\mathbf{A}}^{+}}\ \overset{n^m}{\left[\begin{array}{c} \mathbf{P}_{m,--} \\ \hline \mathbf{P}_{++,--} \end{array}\right]} + \underset{n^d\times n^{--}}{\tilde{\mathbf{A}}^{0--}} \end{array}\right)}_{\equiv\ \underset{n^d\times n^{--}}{\boldsymbol{\alpha}}} n^{--}\left[\begin{array}{c|c} d\mathbf{P}_{--,--} & d\mathbf{P}_{--,m} \end{array}\right] \\
&\qquad \overset{n^m}{} \qquad\qquad \overset{n^{++}}{} \qquad\qquad \overset{n^{--}}{} \qquad \overset{n^m}{} \\
&+\underbrace{n^{d}\left[\begin{array}{c|c} \underset{n^d\times n^+}{\tilde{\mathbf{A}}^{+}}\left[\begin{array}{c} \mathbf{P}_{m,m} \\ \hline \mathbf{P}_{++,m} \end{array}\right] + \underset{n^d\times n^m}{\tilde{\mathbf{A}}^{0m}} & \underset{n^d\times n^{++}}{\tilde{\mathbf{A}}^{0++}} \end{array}\right]}_{\equiv\ \underset{n^d\times n^+}{\boldsymbol{\beta}}} \begin{array}{l} n^m \\ n^{++} \end{array}\left[\begin{array}{c|c} d\mathbf{P}_{m,--} & d\mathbf{P}_{m,m} \\ \hline d\mathbf{P}_{++,--} & d\mathbf{P}_{++,m} \end{array}\right] \\
&\qquad \overset{n^{--}}{} \qquad \overset{n^m}{} \qquad\qquad \overset{n^{--}}{} \qquad \overset{n^m}{} \\
&+\underbrace{\underset{n^d\times n^+}{\tilde{\mathbf{A}}^{+}}}_{\equiv\ \underset{n^d\times n^+}{\boldsymbol{\gamma}}} \begin{array}{l} n^m \\ n^{++} \end{array}\left[\begin{array}{c|c} d\mathbf{P}_{m,--} & d\mathbf{P}_{m,m} \\ \hline d\mathbf{P}_{++,--} & d\mathbf{P}_{++,m} \end{array}\right] \underbrace{\begin{array}{l} n^{--} \\ n^m \end{array}\left[\begin{array}{c|c} \mathbf{P}_{--,--} & \mathbf{P}_{--,m} \\ \hline \mathbf{P}_{m,--} & \mathbf{P}_{m,m} \end{array}\right]}_{\equiv\ \underset{n^-\times n^-}{\boldsymbol{\delta}}} \\
&\qquad \overset{n^{--}}{} \qquad \overset{n^m}{} \\
&+\underbrace{n^{d}\left[\begin{array}{c|c} \tilde{\mathbf{M}}(\tilde{\mathbf{P}})^{--} & \tilde{\mathbf{M}}(\tilde{\mathbf{P}})^{m} \end{array}\right]}_{\equiv\ \underset{n^d\times n^-}{-\boldsymbol{\theta}}} \\
&\overset{n^{--}}{} \quad \overset{n^m}{} \\
=&n^{d}\left[\begin{array}{c|c} \mathbf{0} & \mathbf{0} \end{array}\right]
\end{aligned}
$$

written more compactly as

$$\underset{n^d \times n^-}{\boldsymbol{\theta}} = \underset{n^d \times n^{--}}{\boldsymbol{\alpha}} \begin{matrix} n^{--} \\ n^{--} \end{matrix} \left[\begin{matrix} n^{--} & n^m \\ d\mathbf{P}_{--,--} & d\mathbf{P}_{--,m} \end{matrix} \right] + \underset{n^d \times n^+ n^{++}}{\boldsymbol{\beta}} \begin{matrix} n^m \\ n^{++} \end{matrix} \left[\begin{matrix} n^{--} & n^m \\ d\mathbf{P}_{m,--} & d\mathbf{P}_{m,m} \\ d\mathbf{P}_{++,--} & d\mathbf{P}_{++,m} \end{matrix} \right]$$

$$+ \underset{n^d \times n^+ n^{++}}{\boldsymbol{\gamma}} \begin{matrix} n^m \\ n^{++} \end{matrix} \left[\begin{matrix} n^{--} & n^m \\ d\mathbf{P}_{m,--} & d\mathbf{P}_{m,m} \\ d\mathbf{P}_{++,--} & d\mathbf{P}_{++,m} \end{matrix} \right] \underset{n^- \times n^-}{\boldsymbol{\delta}}$$

Performing a QR decomposition $\underset{n^d \times n^{--}}{\boldsymbol{\alpha}} = \underset{n^d \times n^d}{\mathbf{U}} \begin{matrix} n^{--} \\ n^+ \end{matrix} \left[\begin{matrix} \mathbf{T} \\ \mathbf{0} \end{matrix} \right]$, $\tilde{\mathbf{U}} = \mathbf{U}^* = \begin{matrix} n^{--} & n^d \\ n^+ \end{matrix} \left[\begin{matrix} \tilde{\mathbf{U}}_1 \\ \tilde{\mathbf{U}}_2 \end{matrix} \right]$ and premultiplying with $\tilde{\mathbf{U}}$ gives two sets of equations. First

$$\underset{n^+ \times n^d}{\tilde{\mathbf{U}}_2} \underset{n^d \times n^+ n^{++}}{\boldsymbol{\beta}} \begin{matrix} n^m \\ n^{++} \end{matrix} \left[\begin{matrix} n^{--} & n^m \\ d\mathbf{P}_{m,--} & d\mathbf{P}_{m,m} \\ d\mathbf{P}_{++,--} & d\mathbf{P}_{++,m} \end{matrix} \right] + \underset{n^+ \times n^d}{\tilde{\mathbf{U}}_2} \underset{n^d \times n^+ n^{++}}{\boldsymbol{\gamma}} \begin{matrix} n^m \\ n^{++} \end{matrix} \left[\begin{matrix} n^{--} & n^m \\ d\mathbf{P}_{m,--} & d\mathbf{P}_{m,m} \\ d\mathbf{P}_{++,--} & d\mathbf{P}_{++,m} \end{matrix} \right] \underset{n^- \times n^-}{\boldsymbol{\delta}} = \underset{n^+ \times n^d}{\tilde{\mathbf{U}}_2} \underset{n^d \times n^-}{\boldsymbol{\theta}}$$

A generalized Sylvester equation in $\begin{matrix} n^m \\ n^{++} \end{matrix} \left[\begin{matrix} n^{--} & n^m \\ d\mathbf{P}_{m,--} & d\mathbf{P}_{m,m} \\ d\mathbf{P}_{++,--} & d\mathbf{P}_{++,m} \end{matrix} \right]$. Given its solution, the remaining elements of $d\mathbf{P}$ are given by

$$\begin{matrix} n^{--} \\ n^{--} \end{matrix} \left[\begin{matrix} n^{--} & n^m \\ d\mathbf{P}_{--,--} & d\mathbf{P}_{--,m} \end{matrix} \right] = \underset{n^{--} \times n^{--}}{\mathbf{T}^{-1}} \underset{n^{--} \times n^d}{\tilde{\mathbf{U}}_1} \left(\underset{n^d \times n^-}{\boldsymbol{\theta}} - \underset{n^d \times n^+ n^{++}}{\boldsymbol{\gamma}} \begin{matrix} n^m \\ n^{++} \end{matrix} \left[\begin{matrix} n^{--} & n^m \\ d\mathbf{P}_{m,--} & d\mathbf{P}_{m,m} \\ d\mathbf{P}_{++,--} & d\mathbf{P}_{++,m} \end{matrix} \right] \underset{n^- \times n^-}{\boldsymbol{\delta}} \right.$$

$$\left. - \underset{n^d \times n^+ n^{++}}{\boldsymbol{\beta}} \begin{matrix} n^m \\ n^{++} \end{matrix} \left[\begin{matrix} n^{--} & n^m \\ d\mathbf{P}_{m,--} & d\mathbf{P}_{m,m} \\ d\mathbf{P}_{++,--} & d\mathbf{P}_{++,m} \end{matrix} \right] \right)$$

6.4. Detailed Dynare Topology - Line Search

The line search methods in the text require finding zeros of the polynomial

$$g'(x) = 2\alpha(x-1) + \beta\left(2x - 3x^2\right) + 4\gamma x^3 \tag{A1}$$

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 $\gamma = \left\|A(\Delta P)^2\right\|_F$.

Using the typology from Dynare and the results above

$$\alpha = ||\mathbf{M(P)}||_F^2 = \mathrm{tr}\left(\mathbf{M(P)}^*\mathbf{M(P)}\right)$$

$$= \mathrm{tr}\left(\begin{array}{c} \\ n^s \\ n^{--} \\ n^m \\ n^{++} \end{array} \begin{array}{c} \begin{array}{c} n \end{array} \\ \left[\begin{array}{c} \mathbf{0} \\ \hline \mathbf{M(P)}^*_{--} \\ \hline \mathbf{M(P)}^*_m \\ \hline \mathbf{0} \end{array}\right] \end{array} n \left[\begin{array}{c|c|c|c} n^s & n^{--} & n^m & n^{++} \\ \mathbf{0} & \mathbf{M(P)}_{--} & \mathbf{M(P)}_m & \mathbf{0} \end{array}\right] \right)$$

$$= \mathrm{tr}\left(\mathbf{M(P)}^*_{--}\mathbf{M(P)}_{--}\right) + \mathrm{tr}\left(\mathbf{M(P)}^*_m\mathbf{M(P)}_m\right)$$

$$\underset{n\times n}{\mathbf{M}}(\underset{n\times n}{\mathbf{P}}) = \underset{n\times n}{\mathbf{A}}\mathbf{P}^2 + \underset{n\times n}{\mathbf{B}}\mathbf{P} + \underset{n\times n}{\mathbf{C}}$$

$$= \begin{array}{c} \\ n^s \\ n^{--} \\ n^m \\ n^{++} \end{array} \begin{array}{c} \begin{array}{cccc} n^s & n^{--} & n^m & n^{++} \end{array} \\ \left[\begin{array}{c|c|c} \mathbf{0} & \mathbf{0} & \underset{n^s\times n^+}{\breve{\mathbf{A}}^+} \\ \hline \mathbf{0} & \mathbf{0} & \\ \hline \mathbf{0} & \mathbf{0} & \underset{n^d\times n^+}{\tilde{\mathbf{A}}^+} \\ \hline \mathbf{0} & \mathbf{0} & \end{array}\right] \end{array} n \left[\begin{array}{c|c|c|c} n^s & n^{--} & n^m & n^{++} \\ \mathbf{0} & \mathbf{PP}_{\bullet,--} & \mathbf{PP}_{\bullet,m} & \mathbf{0} \end{array}\right]$$

$$+ \begin{array}{c} \\ n^s \\ n^{--} \\ n^m \\ n^{++} \end{array} \begin{array}{c} \begin{array}{cccc} n^s & n^{--} & n^m & n^{++} \end{array} \\ \left[\begin{array}{c|c|c} \breve{\mathbf{A}}^{0s} & \underset{n^s\times n^d}{\breve{\mathbf{A}}^{0d}} & \\ \hline \mathbf{0} & & \\ \hline \mathbf{0} & \underset{n^d\times n^d}{\tilde{\mathbf{A}}^0} & \\ \hline \mathbf{0} & & \end{array}\right] \end{array} n \left[\begin{array}{c|c|c|c} n^s & n^{--} & n^m & n^{++} \\ \mathbf{0} & \mathbf{P}_{\bullet,--} & \mathbf{P}_{\bullet,m} & \mathbf{0} \end{array}\right]$$

$$+ \begin{array}{c} \\ n^s \\ n^{--} \\ n^m \\ n^{++} \end{array} \begin{array}{c} \begin{array}{cccc} n^s & n^{--} & n^m & n^{++} \end{array} \\ \left[\begin{array}{c|c|c} \mathbf{0} & \underset{n^s\times n^-}{\breve{\mathbf{A}}^-} & \mathbf{0} \\ \hline \mathbf{0} & & \mathbf{0} \\ \hline \mathbf{0} & \underset{n^d\times n^-}{\tilde{\mathbf{A}}^-} & \mathbf{0} \\ \hline \mathbf{0} & & \mathbf{0} \end{array}\right] \end{array}$$

$$= \underset{n}{\overset{n^s}{\left[\begin{array}{c|c|c}\mathbf{0} & \begin{bmatrix}\underset{n^s\times n^s}{\mathbf{0}} & \underset{n^s\times n^{--}}{\mathbf{0}} & \underset{n^s\times n^+}{\breve{\mathbf{A}}^+}\\ \underset{n^{--}\times n^s}{\mathbf{0}} & \underset{n^{--}\times n^{--}}{\mathbf{0}} & \\ \underset{n^m\times n^s}{\mathbf{0}} & \underset{n^m\times n^{--}}{\mathbf{0}} & \underset{n^d\times n^+}{\tilde{\mathbf{A}}^+}\\ \underset{n^{++}\times n^s}{\mathbf{0}} & \underset{n^{++}\times n^{--}}{\mathbf{0}} & \end{bmatrix} & \mathbf{0}\end{array}\right]}}\overset{n^-}{\overset{n^{++}}{}}\underset{n\times n}{\mathbf{P}}\underset{n\times n^{--}}{\left[\mathbf{P}_{\bullet,--}\right.}\ \underset{n\times n^m}{\left.\mathbf{P}_{\bullet,m}\right]}$$

$$+ \underset{n}{\overset{n^s}{\left[\begin{array}{c|c|c}\mathbf{0} & \begin{bmatrix}\underset{n^s\times n^s}{\breve{\mathbf{A}}^{0s}} & \underset{n^s\times n^d}{\breve{\mathbf{A}}^{0d}}\\ \underset{n^{--}\times n^s}{\mathbf{0}} & \\ \underset{n^m\times n^s}{\mathbf{0}} & \underset{n^d\times n^d}{\tilde{\mathbf{A}}^0}\\ \underset{n^{++}\times n^s}{\mathbf{0}} & \end{bmatrix} & \mathbf{0}\end{array}\right]}}\overset{n^-}{\overset{n^{++}}{}}\underset{n\times n^{--}}{\left[\mathbf{P}_{\bullet,--}\right.}\ \underset{n\times n^m}{\left.\mathbf{P}_{\bullet,m}\right]}\underset{n}{\overset{n^s}{\left[\begin{array}{c|c|c}\mathbf{0} & \mathbf{0} & \mathbf{0}\end{array}\right]}}\overset{n^-}{\overset{n^{++}}{}}+\underset{n}{\overset{n^s}{\left[\begin{array}{c|c|c}\mathbf{0} & \begin{bmatrix}\underset{n^s\times n^-}{\breve{\mathbf{A}}^-}\\ \underset{n^d\times n^-}{\tilde{\mathbf{A}}^-}\end{bmatrix} & \mathbf{0}\end{array}\right]}}\overset{n^-}{\overset{n^{++}}{}}$$

$$\begin{bmatrix}\underset{n^s\times n^s}{\mathbf{0}} & \underset{n^s\times n^{--}}{\mathbf{0}} & \underset{n^s\times n^+}{\breve{\mathbf{A}}^+}\\ \underset{n^{--}\times n^s}{\mathbf{0}} & \underset{n^{--}\times n^{--}}{\mathbf{0}} & \\ \underset{n^m\times n^s}{\mathbf{0}} & \underset{n^m\times n^{--}}{\mathbf{0}} & \underset{n^d\times n^+}{\tilde{\mathbf{A}}^+}\\ \underset{n^{++}\times n^s}{\mathbf{0}} & \underset{n^{++}\times n^{--}}{\mathbf{0}} & \end{bmatrix}\underset{n\times n}{\mathbf{P}}\underset{n\times n^{--}}{\left[\mathbf{P}_{\bullet,--}\right.}\ \underset{n\times n^m}{\left.\mathbf{P}_{\bullet,m}\right]}=\begin{bmatrix}\underset{n^s\times n^+}{\breve{\mathbf{A}}^+}\\ \\ \underset{n^d\times n^+}{\tilde{\mathbf{A}}^+}\end{bmatrix}\underset{n^+\times n}{\mathbf{P}_{m/++,\bullet}}\ \underset{n\times n^-}{\mathbf{P}_{\bullet,--/m}}$$

where

$$\underset{n\times n^-}{\mathbf{P}_{\bullet,--/m}}=\underset{n}{\overset{n^{--}}{\left[\mathbf{P}_{\bullet,--}\right.}}\ \overset{n^m}{\left.\mathbf{P}_{\bullet,m}\right]}\qquad\text{and}\qquad\underset{n\times n^+}{\mathbf{P}_{\bullet,m/++}}=\underset{n}{\overset{n^m}{\left[\mathbf{P}_{\bullet,m}\right.}}\ \overset{n^{++}}{\left.\mathbf{P}_{\bullet,++}\right]}$$

$$\underset{n\times n}{\mathbf{M}(\mathbf{P})}=\underset{n}{\overset{n^s}{\left[\begin{array}{c|c|c}\mathbf{0} & \left(\begin{bmatrix}\underset{n^s\times n^+}{\breve{\mathbf{A}}^+}\\ \\ \underset{n^d\times n^+}{\tilde{\mathbf{A}}^+}\end{bmatrix}\underset{n^+\times n}{\mathbf{P}_{m/++,\bullet}}+\begin{bmatrix}\underset{n^s\times n^s}{\breve{\mathbf{A}}^{0s}} & \underset{n^s\times n^d}{\breve{\mathbf{A}}^{0d}}\\ \underset{n^{--}\times n^s}{\mathbf{0}} & \\ \underset{n^m\times n^s}{\mathbf{0}} & \underset{n^d\times n^d}{\tilde{\mathbf{A}}^0}\\ \underset{n^{++}\times n^s}{\mathbf{0}} & \end{bmatrix}\right)\underset{n\times n^-}{\mathbf{P}_{\bullet,--/m}}+\begin{bmatrix}\underset{n^s\times n^-}{\breve{\mathbf{A}}^-}\\ \underset{n^d\times n^-}{\tilde{\mathbf{A}}^-}\end{bmatrix} & \mathbf{0}\end{array}\right]}}\overset{n^-}{\overset{n^{++}}{}}$$

$$=\underset{n}{\overset{n^s}{\left[\begin{array}{c|c|c}\mathbf{0} & \mathbf{X} & \mathbf{0}\end{array}\right]}}\overset{n^-}{\overset{n^{++}}{}}$$

$$\underset{n\times n^-}{\mathbf{X}} = \begin{matrix} n^s \\ n^d \end{matrix}\overset{n^-}{\left[\begin{matrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{matrix}\right]} = \begin{matrix} n^s \\ n^d \end{matrix}\overset{n^-}{\left[\begin{matrix} \left(\underset{n^s\times n^+}{\breve{\mathbf{A}}^+}\underset{n^+\times n}{\mathbf{P}_{m/++,\bullet}} + \left[\underset{n^s\times n^s}{\breve{\mathbf{A}}^{0s}}\ \underset{n^s\times n^d}{\breve{\mathbf{A}}^{0d}}\right]\right)\underset{n\times n^-}{\mathbf{P}_{\bullet,--/m}} + \underset{n^s\times n^-}{\breve{\mathbf{A}}^-} \\ \underset{n^d\times n^+}{\tilde{\mathbf{A}}^+}\underset{n^+\times n}{\mathbf{P}_{m/++,\bullet}}\underset{n\times n^-}{\mathbf{P}_{\bullet,--/m}} + \left[\underset{n^d\times n^s}{\mathbf{0}}\ \underset{n^d\times n^d}{\tilde{\mathbf{A}}^0}\right]\underset{n\times n^-}{\mathbf{P}_{\bullet,--/m}} + \underset{n^d\times n^-}{\tilde{\mathbf{A}}^-} \end{matrix}\right]}$$

$$= \begin{matrix} n^s \\ n^d \end{matrix}\overset{n^-}{\left[\begin{matrix} \left(\underset{n^s\times n^+}{\breve{\mathbf{A}}^+}\underset{n^+\times n}{\mathbf{P}_{m/++,\bullet}} + \left[\underset{n^s\times n^s}{\breve{\mathbf{A}}^{0s}}\ \underset{n^s\times n^d}{\breve{\mathbf{A}}^{0d}}\right]\right)\underset{n\times n^-}{\mathbf{P}_{\bullet,--/m}} + \underset{n^s\times n^-}{\breve{\mathbf{A}}^-} \\ \underset{n^d\times n^+}{\tilde{\mathbf{A}}^+}\underset{n^+\times n}{\mathbf{P}_{m/++,\bullet}}\underset{n\times n^-}{\mathbf{P}_{\bullet,--/m}} + \underset{n^d\times n^d}{\tilde{\mathbf{A}}^0}\underset{n^d\times n^-}{\mathbf{P}_{d,--/m}} + \underset{n^d\times n^-}{\tilde{\mathbf{A}}^-} \end{matrix}\right]}$$

$$= \begin{matrix} n^s \\ n^d \end{matrix}\overset{n^-}{\left[\begin{matrix} \underset{n^s\times n^s}{\breve{\mathbf{A}}^{0s}}\underset{n^s\times n^-}{\mathbf{P}_{s,--/m}} + \left(\underset{n^s\times n^+}{\breve{\mathbf{A}}^+}\underset{n^+\times n^d}{\mathbf{P}_{m/++,d}} + \underset{n^s\times n^d}{\breve{\mathbf{A}}^{0d}}\right)\underset{n^d\times n^-}{\mathbf{P}_{d,--/m}} + \underset{n^s\times n^-}{\breve{\mathbf{A}}^-} \\ \underset{n^d\times n^+}{\tilde{\mathbf{A}}^+}\underset{n^+\times n}{\mathbf{P}_{m/++,\bullet}}\underset{n\times n^-}{\mathbf{P}_{\bullet,--/m}} + \underset{n^d\times n^d}{\tilde{\mathbf{A}}^0}\underset{n^d\times n^-}{\mathbf{P}_{d,--/m}} + \underset{n^d\times n^-}{\tilde{\mathbf{A}}^-} \end{matrix}\right]}$$

$$\mathrm{tr}\left(\mathbf{M}(\mathbf{P})^*\mathbf{M}(\mathbf{P})\right) = \mathrm{tr}(\mathbf{X}^*\mathbf{X}) = \mathrm{tr}(\mathbf{X}_1^*\mathbf{X}_1) + \mathrm{tr}(\mathbf{X}_2^*\mathbf{X}_2) \qquad \text{by construction, } \underset{n^s\times n^-}{\mathbf{X}_1} = \underset{n^s\times n^-}{\mathbf{0}}$$

$$= \mathrm{tr}(\mathbf{X}_2^*\mathbf{X}_2) = \mathrm{tr}\left(\tilde{\mathbf{M}}(\tilde{\mathbf{P}})^*\tilde{\mathbf{M}}(\tilde{\mathbf{P}})\right)$$

$$\gamma = \left|\left|\mathbf{A}d\mathbf{P}^2\right|\right|_F^2 = \mathrm{tr}\left(\left(\mathbf{A}d\mathbf{P}^2\right)^*\mathbf{A}d\mathbf{P}^2\right)$$

$$\mathbf{A}d\mathbf{P}^2 = \begin{matrix} n \end{matrix}\overset{n^s}{\overset{n^-}{\overset{n^{++}}{\left[\begin{matrix} \mathbf{0} & \underset{n\times n}{\mathbf{A}}\ d\ \underset{n\times n}{\mathbf{P}}\ d\underset{n\times n^-}{\mathbf{P}_{\bullet,--/m}} & \mathbf{0} \end{matrix}\right]}}}$$

$$= \begin{matrix} n \end{matrix}\overset{n^s}{\overset{n^-}{\overset{n^{++}}{\left[\begin{matrix} \mathbf{0} & \left[\begin{matrix} \underset{n^s\times n^+}{\breve{\mathbf{A}}^+} \\ \underset{n^d\times n^+}{\tilde{\mathbf{A}}^+} \end{matrix}\right] d\underset{n^+\times n}{\mathbf{P}_{m/++,\bullet}}d\underset{n\times n^-}{\mathbf{P}_{\bullet,--/m}} & \mathbf{0} \end{matrix}\right]}}}$$

$$= \begin{matrix} n \end{matrix}\overset{n^s}{\overset{n^-}{\overset{n^{++}}{\left[\begin{matrix} \mathbf{0} & \left[\begin{matrix} \underset{n^s\times n^+}{\breve{\mathbf{A}}^+} \\ \underset{n^d\times n^+}{\tilde{\mathbf{A}}^+} \end{matrix}\right] d\underset{n^+\times n^-}{\mathbf{P}_{m/++,--/m}}d\underset{n^-\times n^-}{\mathbf{P}_{--/m,--/m}} & \mathbf{0} \end{matrix}\right]}}}$$

$$= \begin{matrix} n \end{matrix}\overset{n^s}{\overset{n^-}{\overset{n^{++}}{\left[\begin{matrix} \mathbf{0} & \left[\begin{matrix} \underset{n^s\times n^-}{\mathbf{Y}_1} \\ \underset{n^d\times n^-}{\mathbf{Y}_2} \end{matrix}\right] & \mathbf{0} \end{matrix}\right]}}}$$

$$\delta\mathbf{P} = d\mathbf{P}_{\substack{m/++,--/m \\ n^+\times n^-}} d\mathbf{P}_{\substack{--/m,--/m \\ n^-\times n^-}}$$

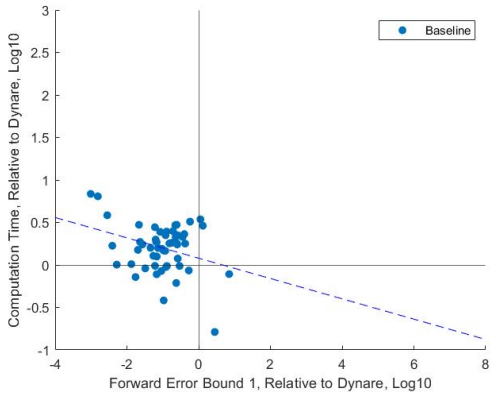
$$\begin{aligned}
\gamma &= \operatorname{tr}(\mathbf{Y}_1^*\mathbf{Y}_1) + \operatorname{tr}(\mathbf{Y}_2^*\mathbf{Y}_2) \\
&= \operatorname{tr}\left(\underset{n^s\times n^+}{\breve{\mathbf{A}}^+}\ \underset{n^+\times n^-}{\delta\mathbf{P}}\ \underset{n^-\times n^+}{\delta\mathbf{P}^*}\ \underset{n^+\times n^s}{\breve{\mathbf{A}}^{+*}}\right) + \operatorname{tr}\left(\underset{n^d\times n^+}{\tilde{\mathbf{A}}^+}\ \underset{n^+\times n^-}{\delta\mathbf{P}}\ \underset{n^-\times n^+}{\delta\mathbf{P}^*}\ \underset{n^+\times n^d}{\tilde{\mathbf{A}}^{+*}}\right) \\
&= \operatorname{tr}\left(\underset{n^-\times n^+}{\delta\mathbf{P}^*}\left[\underset{n^+\times n^s}{\breve{\mathbf{A}}^{+*}}\ \underset{n^s\times n^+}{\breve{\mathbf{A}}^+} + \underset{n^+\times n^d}{\tilde{\mathbf{A}}^{+*}}\ \underset{n^d\times n^+}{\tilde{\mathbf{A}}^+}\right]\underset{n^+\times n^-}{\delta\mathbf{P}}\right)
\end{aligned}$$

$$\operatorname{tr}\left(\mathbf{A}d\mathbf{P}^2\cdot\mathbf{M}(\mathbf{P})^*\right) = \operatorname{tr}\left(\begin{array}{c|c|c} & \overset{n^s}{} & \overset{n^-}{} \quad \overset{n^{++}}{} \\ \hline n & \mathbf{0} & \begin{bmatrix}\mathbf{Y}_1 \\ n^s\times n^-\end{bmatrix} \\ & & \begin{bmatrix}\mathbf{Y}_2 \\ n^d\times n^-\end{bmatrix} \quad \mathbf{0} \end{array}\right) \cdot \begin{array}{c} \overset{n}{} \\ \begin{matrix} n^s \\ n^- \\ n^{++} \end{matrix} \end{array} \left[\begin{array}{c} \mathbf{0} \\ \hline \begin{bmatrix}\mathbf{0} & \mathbf{X}_2^* \\ n^-\times n^s & n^-\times n^d\end{bmatrix} \\ \hline \mathbf{0} \end{array}\right]$$

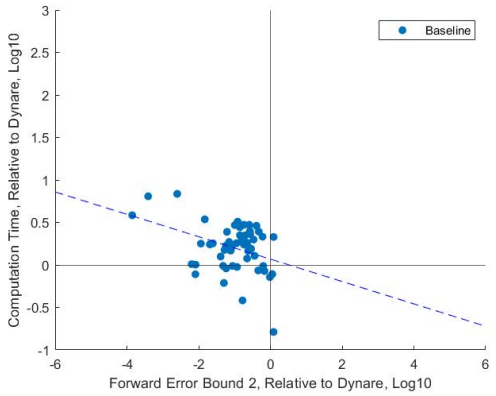
$$= \operatorname{tr}\left(\begin{bmatrix}\mathbf{Y}_1 \\ n^s\times n^-\\ \mathbf{Y}_2 \\ n^d\times n^-\end{bmatrix}\begin{bmatrix}\mathbf{0} & \mathbf{X}_2^* \\ n^-\times n^s & n^-\times n^d\end{bmatrix}\right)$$

$$= \operatorname{tr}\left(\underset{n^d\times n^-}{\mathbf{Y}_2}\ \underset{n^-\times n^d}{\mathbf{X}_2^*}\right)$$

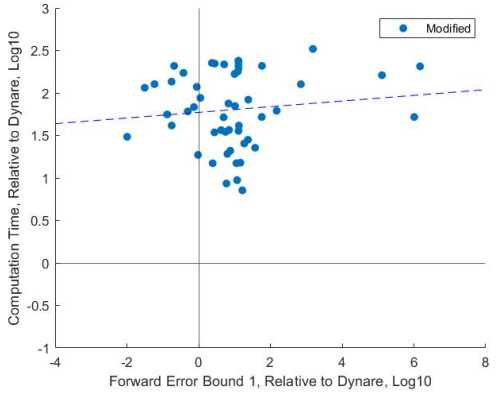
6.5. Additional Figures



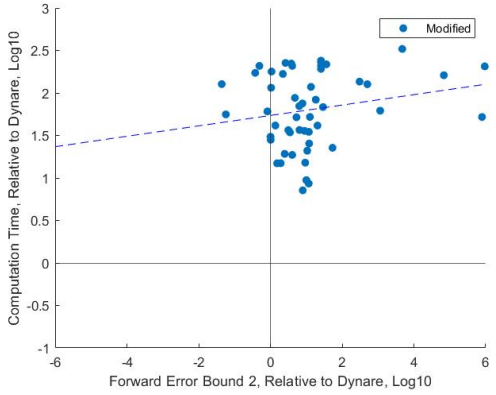
(A) Forward Error 1, Baseline Relative to Dynare



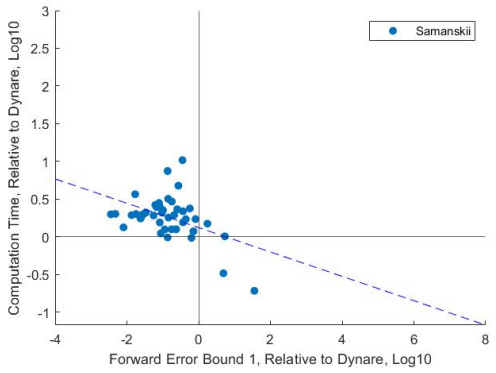
(B) Forward Error 2, Baseline Relative to Dynare



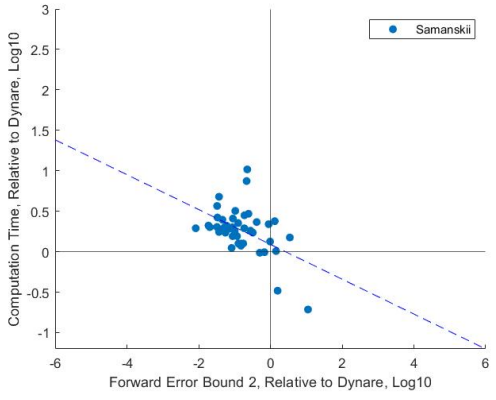
(C) Forward Error 1, Modified Relative to Dynare



(D) Forward Error 2, Modified Relative to Dynare

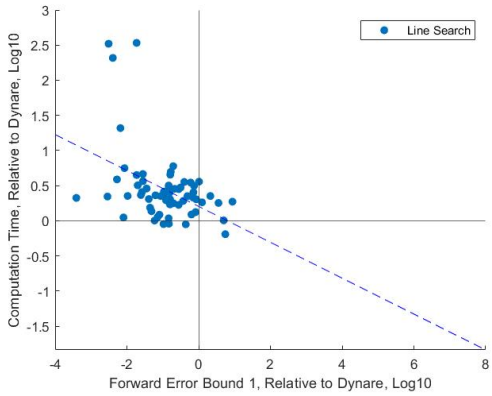


(E) Forward Error 1, Šamanskii Relative to Dynare

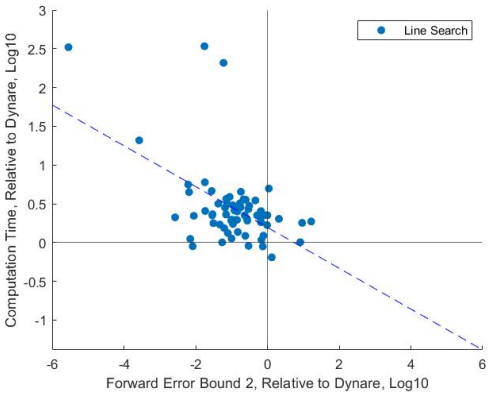


(F) Forward Error 2, Šamanskii Relative to Dynare

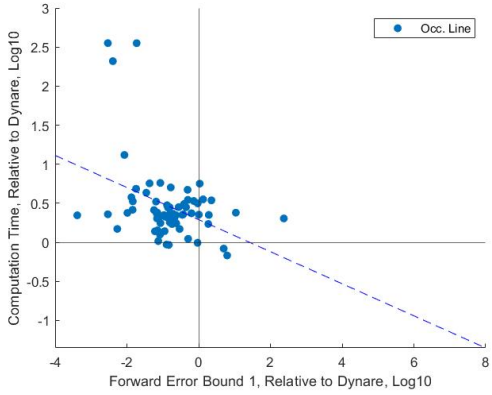
FIGURE 5. Forward Errors and Computation Time for the Macroeconomic Model Data Base (MMB)



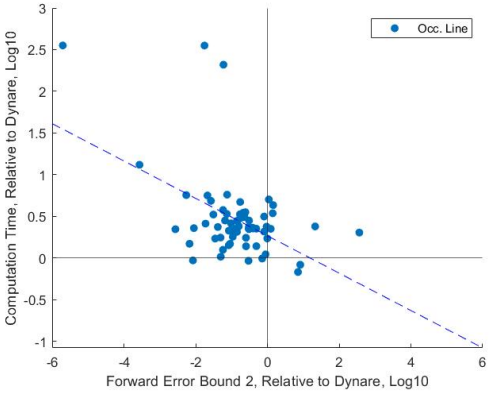
(A) Forward Error 1, Line Searches Relative to Dynare



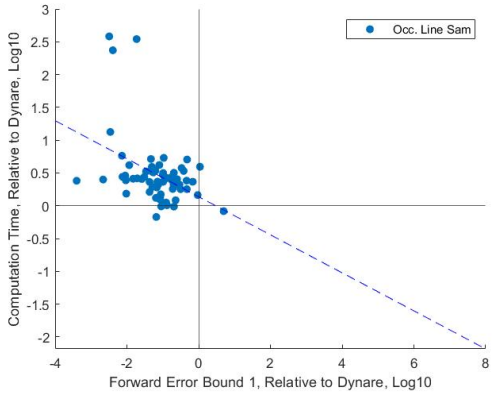
(B) Forward Error 2, Line Searches Relative to Dynare



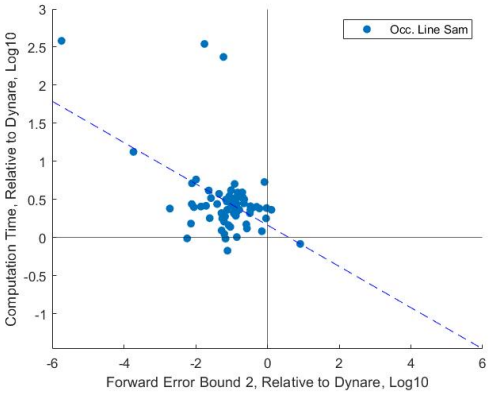
(C) Forward Error 1, Occ. Line Searches Relative to Dynare



(D) Forward Error 2, Occ. Line Searches to Dynare



(E) Forward Error 1, Occ. LS & Šamanskii Relative to Dynare



(F) Forward Error 2, Occ. LS & Šamanskii Relative to Dynare

FIGURE 6. Forward Errors and Computation Time for the Macroeconomic Model Data Base (MMB), Continued

REFERENCES

ADJEMIAN, S., H. BASTANI, M. JUILLARD, F. MIHOUBI, G. PERENDIA, M. RATTO, AND S. VILLEMOT (2011): "Dynare: Reference Manual, Version 4," Dynare Working Papers 1, CEPREMAP.

AN, S., AND F. SCHORFHEIDE (2007): "Bayesian Analysis of DSGE Models," *Econometric Reviews*, 26(2-4), 113–172.

ANDERSON, G. S. (2010): "A Reliable and Computationally Efficient Algorithm for Imposing the Saddle Point Property in Dynamic Models," *Journal of Economic Dynamics and Control*, 34(3), 472–489.

ANDERSON, G. S., A. LEVIN, AND E. SWANSON (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., AND G. MOORE (1985): "A Linear Algebraic Procedure for Solving Linear Perfect Foresight Models," *Economics Letters*, 17(3), 247–252.

BINDER, M., AND M. H. PESARAN (1997): "Multivariate Linear Rational Expectations Models: Characterization of the Nature of the Solutions and Their Fully Recursive Computation," *Econometric Theory*, 13(6), 877–88.

BLANCHARD, O. J. (1979): "Backward and Forward Solutions for Economies with Rational Expectations," *The American Economic Review*, 69(2), 114–118.

CAYLEY, A. (1879): "Desiderata and Suggestions: No. 3. The Newton-Fourier Imaginary Problem," *American Journal of Mathematics*, 2(1), 97–97.

CHEN, X. S., AND P. LV (2018): "On estimating the separation between (A,B) and (C,D) associated with the generalized Sylvester equation $AXD - BXC = E$," *Journal of Computational and Applied Mathematics*, 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 and its Applications*, 93, 93–105.

CORLESS, R. M., AND N. FILLION (2013): *A Graduate Introduction to Numerical Methods*. Springer.

DENNIS, JR., J. E., J. F. TRAUB, AND R. P. WEBER (1976): "The Algebraic Theory of Matrix Polynomials," *SIAM Journal on Numerical Analysis*, 13(6), 831–845.

GANTMACHER, F. R. (1959): *The Theory of Matrices*, vol. I&II. Chelsea Publishing Company, New York, NY.

GOLUB, G. H., AND C. F. VAN LOAN (2013): *Matrix Computations*. The Johns Hopkins University Press, fourth edn.

HAMMARLING, S., C. J. MUNRO, AND F. TISSEUR (2013): "An Algorithm for the Complete Solution of Quadratic Eigenvalue Problems," *ACM Transactions On Mathematical Software*, 39(3), 18:1–18:19.

HIGHAM, N. J. (2002): *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, second edn.

HIGHAM, N. J., AND H.-M. KIM (2001): "Solving a Quadratic Matrix Equation by Newton's Method with Exact Line Searches," *SIAM Journal on Matrix Analysis and Applications*, 23(2), 499–519.

JUDD, K. L. (1992): "Projection Methods for Solving Aggregate Growth Models," *Journal of Economic Theory*, 58(2), 410–452.

KLEIN, P. (2000): "Using the Generalized Schur Form to Solve a Multivariate Linear Rational Expectations Model," *Journal of Economic Dynamics and Control*, 24(10), 1405–1423.

KÅGSTRÖM, B. (1994): "A Perturbation Analysis of the Generalized Sylvester Equation $(AR-LB, DR-LE) = (C, F)$," *SIAM Journal on Matrix Analysis and Applications*, 15(4), 1045–1060.

KÅGSTRÖM, B., AND P. POROMAA (1996): "LAPACK-style algorithms and software for solving the generalized Sylvester equation and estimating the separation between regular matrix pairs," *ACM Transactions on Mathematical Software (TOMS)*, 22(1), 78–103.

LAN, H., AND A. MEYER-GOHDE (2014): "Solvability of Perturbation Solutions in DSGE Models," *Journal of Economic Dynamics and Control*, 45, 366–388.

LONG, J., X. HU, AND L. ZHANG (2008): "Improved Newton's method with exact line searches to solve quadratic matrix equation," *Journal of Computational and Applied Mathematics*, 222(2), 645–654.

MEYER-GOHDE, A. (2022a): "Backward Error and Condition Number Analysis of Linear DSGE Solutions," mimeo, Goethe University Frankfurt, Institute for Monetary and Financial Stability (IMFS).

——— (2022b): "Solving Linear DSGE Models with Bernoulli Iterations," mimeo, Goethe University Frankfurt, Institute for Monetary and Financial Stability (IMFS).

MOLER, C. B., AND G. W. STEWART (1973): "An Algorithm for Generalized Matrix Eigenvalue Problems," *SIAM Journal on Numerical Analysis*, 10(2), 241–256.

SCHRÖDER, E. (1870): "Ueber unendlich viele Algorithmen zur Auflösung der Gleichungen," *Mathematische Annalen*, 2, 317–365.

SIMS, C. A. (2001): "Solving Linear Rational Expectations Models," *Computational Economics*, 20(1-2), 1–20.

SMETS, F., AND R. WOUTERS (2007): "Shocks and Frictions in US Business Cycles: A Bayesian DSGE Approach," *The American Economic Review*, 97(3), 586–606.

STEWART, G. W. (1971): "Error Bounds for Approximate Invariant Subspaces of Closed Linear Operators," *SIAM Journal on Numerical Analysis*, 8(4), 796–808.

TISSEUR, F., AND K. MEERBERGEN (2001): "The Quadratic Eigenvalue Problem," *SIAM Review*, 43(2), 235–286.

UHLIG, H. (1999): "A Toolkit for Analysing Nonlinear Dynamic Stochastic Models Easily," in *Computational Methods for the Study of Dynamic Economies*, ed. by R. Marimon, and A. Scott, chap. 3, pp. 30–61. Oxford University Press.

VILLEMOT, S. (2011): "Solving Rational Expectations Models at First Order: What Dynare Does," Dynare Working Papers 2, CEPREMAP.

WIELAND, V., E. AFANASYEVA, M. KUETE, AND J. YOO (2016): "New Methods for Macro-Financial Model Comparison and Policy Analysis," in *Handbook of Macroeconomics*, ed. by J. B. Taylor, and H. Uhlig, vol. 2 of *Handbook of Macroeconomics*, pp. 1241–1319. Elsevier.

WIELAND, V., T. CWIK, G. J. MÜLLER, S. SCHMIDT, AND M. WOLTERS (2012): "A new comparative approach to macroeconomic modeling and policy analysis," *Journal of Economic Behavior & Organization*, 83(3), 523–541.