Solving Linear DSGE Models with Newton Methods

Alexander Meyer-Gohde
Johanna Saecker

Goethe University Frankfurt
Institute for Monetary and Financial Stability (IMFS)

saecker@hof.uni-frankfurt.de
meyer-gohde@econ.uni-frankfurt.de

ESEM-EEA 2023, Barcelona

August 30, 2023

Overview

Problem Statement

Newton-based Methods

Results

Problem Statement

Existing methods:

- using QZ/generalized Schur decomposition: Adjemian et al. (2011), Sims (2001), Uhlig (1999), Klein (2000)
- other: Anderson (2010), Binder and Pesaran (1997), Adjemian et al. (2011)

Research question:
Can Newton-based methods be an interesting alternative for solving linear DSGE models?

Results: **YES!**

- gains in **accuracy** compared to Dynare's QZ-based solution
- if good starting guess → gains in **speed**

Problem Statement

Nonlinear DSGE model approximated linearly around the steady state

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

A, B, C, D: matrices with derivatives, dimensions n_y x n_y
y_t: vector of n_y endogenous variables in (log) deviations from steady states
ε_t: vector of n_e exogenous shocks with a known distribution

Goal: find recursive linear solution of the form

$$y_t = P\ y_{t-1} + Q\ \varepsilon_t \tag{2}$$

Problem Statement

Restrictions:

$$0 = A\,\mathbf{P}^2 + B\,\mathbf{P} + C \tag{3}$$

$$0 = (A\,\mathbf{P} + B)\,Q + D \tag{4}$$

$\Rightarrow$ looking for unique **P** with eigenvalues inside the closed unit circle

Given P, unique Q can be found (Lan and Meyer-Gohde, 2014)

2 Newton-based Methods

Newton's Method (Higham and Kim, 2001)

Goal: find P which fulfills

$$M(P) \equiv AP^2 + BP + C = 0 \tag{5}$$

- iterate through different P_j by updating $P_{j+1} = P_j + \Delta P$
- calculate ΔP: Newton-step in each iteration j
- stop when $M(P)$ close enough to zero

$$M(P + \Delta P) = A(P + \Delta P)^2 + B(P + \Delta P) + C = 0 \tag{6}$$

$$\ldots$$

$$M(P + \Delta P) = M(P) + \underbrace{(A\,\Delta P P + (A\,P + B\,)\,\Delta P)}_{\mathscr{D}_P(\Delta P)} + A\,\Delta P^2 = 0$$

$$A\,\Delta P P + (A\,P + B\,)\,\Delta P = -M(P) \tag{7}$$

Baseline Newton algorithm (Higham and Kim, 2001)

(1) Baseline (Higham and Kim, 2001)

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

$$A\,\Delta P_j P_j + (A\,P_j + B\,)\,\Delta P_j = -M(P_j) \tag{8}$$

  2. Set $P_{j+1} = P_j + \Delta P_j$
  3. Advance $j = j + 1$
- Return P_j

$\Rightarrow$ quadratic convergence, computationally intense/iteration

Newton algorithms

(2) Modified:

- never update P_j $\Rightarrow$ save time each iteration, linear convergence

(3) With Šamanskii technique (ŠT):

- add an iteration like in (2) before updating all P_j $\Rightarrow$ cubic convergence for good starting guess

...

1. Solve for ΔP_j in

$$A\,\Delta P_j P_j + (A\,P_j + B\,)\,\Delta P_j = -M(P_j) \tag{9}$$

2. Set $P_{j+1} = P_j + \Delta P_j$

...

Newton algorithms: line searches

$\Rightarrow$ improve global convergence

(4) With **exact** line searches (LS) (Higham and Kim, 2001):

- calculate multiple t_j of step size ΔP_j for updating based on norm criterium in each iteration

(5) With **occasional** line searches (Long et al., 2008)

- exact line searches only if far away from convergence criterium

(6) With **occ.** line searches or **Šamanskii** T. (Long et al., 2008):

- criterium-based line searches, otherwise Šamanskii step

1. Solve for ΔP_j in

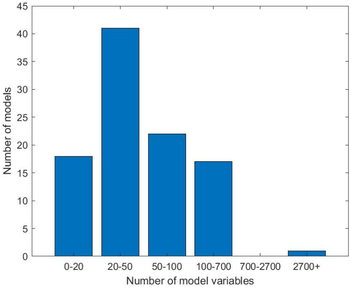
$$A \, \Delta P_j P_j + (A \, P_j + B \,) \, \Delta P_j = -M(P_j) \tag{10}$$

2. Set $P_{j+1} = P_j + t_j \, \Delta P_j$

3 Results

1. MMB Comparison: Setup

- Macroeconomic Model Data Base (MMB) (Wieland et al., 2012)
- test Newton methods for 99 models
- compare to Dynare's QZ-based algorithm
  - **accuracy**
  - speed
- starting guess
  1. zero matrix
  2. Dynare's P



1. MMB Comparison: Speed & Convergence

Method	Conv.	Run Time			Iterations
		Median	Min	Max	
Dynare (QZ)	99	1	1	1	1
Baseline	53	1.74	0.16	6.86	8
Modified	51	61.81	7.15	329.85	686
Šamanskii T.	43	1.95	0.19	10.41	5
LS	67	2.25	0.64	340.54	9
Occ. LS	67	2.28	0.68	354.94	9
Occ. LS & ŠT	67	2.51	0.68	382.05	9

Initial guess: zero-matrix. Conv.: models that converged to the stable solution. Run time relative to Dynare.

Initial guess: zero-matrix

$\Rightarrow$ no guarantee of convergence to unique stable solution ($< 68\%$)

$\Rightarrow$ modestly slower

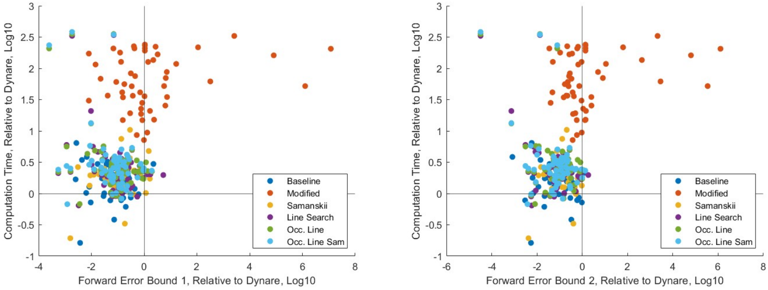
1. MMB Comparison: Accuracy

Method	Forward Error 1			Forward Error 2		
	Median	Min	Max	Median	Min	Max
Dynare (QZ)	1	1	1	1	1	1
Baseline	0.076	2.5e-3	2.3	0.1	0.00082	1.2
Modified	0.85	0.008	1.2e+7	0.65	0.035	1.3e+6
Šamanskii T.	0.13	1.6e-3	1.6	0.1	0.0042	1.1
LS	0.11	2.5e-4	5.3	0.086	3.2e-5	1.8
Occ. LS	0.096	2.5e-4	2	0.094	3.2e-5	0.99
Occ. LS & ŠT	0.09	2.5e-4	2	0.082	3.2e-5	1.4

Initial guess: zero-matrix. Forward error calculation according to Meyer-Gohde (2022).

$\Rightarrow$ one order of magnitude more accurate

$$\underbrace{\frac{\left\|P-\hat{P}\right\|_F}{\|P\|_F}}_{\text{forward error}} \leq \underbrace{\frac{\left\|H_{\hat{P}}^{-1}\operatorname{vec}\left(R_{\hat{P}}\right)\right\|_2}{\left\|\hat{P}\right\|_F}}_{\text{bound 1}} \leq \underbrace{\left\|H_{\hat{P}}^{-1}\right\|_2 \frac{\left\|R_{\hat{P}}\right\|_F}{\left\|\hat{P}\right\|_F}}_{\text{bound 2}}$$



Initial guess: zero-matrix. Only models converging to unique stable solution.

Initial guess: zero-matrix

$\Rightarrow$ slower

$\Rightarrow$ more accurate

2. MMB Comparison: Speed & Convergence

Method	Convergence	Run Time Median	Run Time Min	Run Time Max	Iterations
Dynare (QZ)	99	1	1	1	1
Baseline Newton Method	99	0.34	0.032	29	1
Modified Newton Method	99	0.34	0.031	25	1
with Šamanskii Technique	99	0.49	0.055	70	1
with Line Searches	99	0.34	0.033	30	1
with Occ. Line Searches	99	0.33	0.032	63	1
with Occ. LS & ŠT	99	0.54	0.058	71	1

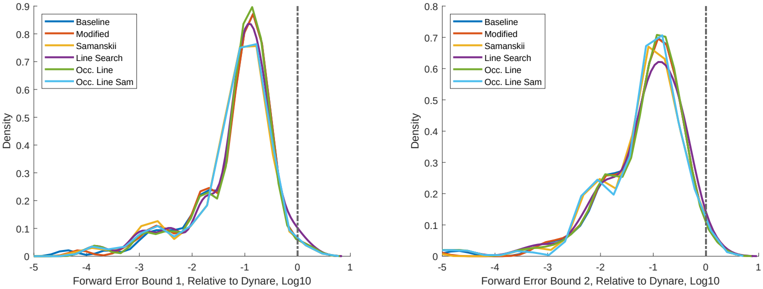
Initial guess: QZ-based solution. Run time relative to Dynare.

Initial guess: Dynare's solution

$\Rightarrow$ convergence rate significantly improved

$\Rightarrow$ one order of magnitude quicker

2. MMB Comparison: Distribution



Initial guess: Dynare's solution

$\Rightarrow$ **all** methods more accurate than Dynare(QZ)

Summary

What we did:

- use 6 Newton algorithms to solve P-matrix
- solve 99 models of Macro Modelbase

What we found:

- gains in **accuracy** compared to Dynare's QZ solution
- good **starting guess** necessary to a) guarantee convergence to desirable solution, b) gains in speed

Paper:

- Newton methods strong in iterative environments
- solve different parameterizations of Taylor rule in Smets and Wouters (2007) model

Part of a research agenda

- Solving linear DSGE models with Bernoulli iterations (Meyer-Gohde, 2023)
- Solving linear DSGE models with Structure-Preserving Doubling methods (Huber, Meyer-Gohde, Saecker)
- Backward Error and Condition Number Analysis of Linear DSGE Solutions (Meyer-Gohde, 2022)

Bibliography I

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.

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, 472–489.

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, 877–88.

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, 499–519.

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

Lan, H. and A. Meyer-Gohde (2014): "Solvability of Perturbation Solutions in DSGE Models," *Journal of Economic Dynamics and Control*, 45, 366–388.

Bibliography II

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, 645–654.

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

——— (2023): "Solving linear DSGE models with Bernoulli iterations," Tech. rep., IMFS Working Paper Series.

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

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

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, Oxford University Press, chap. 3, 30–61.

Bibliography III

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, 523–541.

1 Appendix

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
  $$A\,\Delta P_j P_j + (A\,P_j + B\,)\,\Delta P_j = -M(P_j) \tag{11}$$
  2. Set $P_{j+1} = P_j + \Delta P_j$
  3. Advance $j = j + 1$
- Return P_j

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

$$A\,\Delta P_j P_0 + (A\,P_0 + B\,)\,\Delta P_j = -M(P_j) \qquad (12)$$

  2. Set $P_{j+1} = P_j + \Delta P_j$
  3. Advance $j = j + 1$
- Return P_j

Newton's Method with 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 $i < m$
     1.1 Solve for $\Delta P_{j,i}$ in
     $$A\,\Delta P_{j,i} P_j + (A\,P_j + B\,)\,\Delta P_{j,i} = -M(P_{j,i}) \tag{13}$$
     1.2 Set $P_{j,i+1} = P_{j,i} + \Delta P_{j,i}$
     1.3 Advance $i = i + 1$
  2. Set $P_{j+1} = P_{j,m}$
  3. Advance $j = j + 1$
- Return P_j

Newton-Based Method with 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
  $$A\,\Delta P_j P_j + (A\,P_j + B\,)\,\Delta P_j = -M(P_j) \tag{14}$$
  2. Solve for t_j in
  $$t_j = \operatorname*{argmin}_{x\in[0,2]} \|M(P_j + x\Delta P_j)\|_F^2 \tag{15}$$
  3. Set $P_{j+1} = P_j + t_j\Delta P_j$
  4. Advance $j = j + 1$
- Return P_j

Newton-Based Method with 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
  $$A\,\Delta P_j P_j + (A\,P_j + B\,)\,\Delta P_j = -M(P_j) \tag{16}$$
  2. if criterion$(P_j + \Delta P_j) > \epsilon_0$
     2.1 Solve for t_j in
     $$t_j = \underset{x\in[0,2]}{\operatorname{argmin}} \|M(P_j + x\Delta P_j)\|_F^2 \tag{17}$$
     2.2 Set $P_{j+1} = P_j + t_j \Delta P_j$
  3. else
     3.1 Set $P_{j+1} = P_j + \Delta P_j$
  4. Advance $j = j + 1$
- Return P_j

Newton-Based Method with 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

$$A\,\Delta P_j P_j + (A\,P_j + B\,)\,\Delta P_j = -M(P_j) \tag{18}$$

  2. if criterion$(P_j + \Delta P_j) > \epsilon_0$
     - line search
  3. else
     - Šamanskii step
  4. Advance $j = j + 1$
- Return P_j