University of Kent

School of Economics Discussion Papers

Reverse mode differentiation for DSGE models

Alfred Duncan

July 2021

KDPE 2108

Reverse mode differentiation for DSGE models

Alfred Duncan[†]

University of Kent

This version: July 2021

Abstract

This paper provides a reverse mode derivative for DSGE models. Reverse mode differentiation enables the efficient computation of gradients from the model likelihood to the model parameters. These gradients can then be used by derivative based sampling algorithms including the No U-Turn Sampler. Benchmarks are provided using a small scale New Keynesian model. Our benchmarks demonstrate that MCMC chains generated using the No U-turn Sampler converge much more quickly than those generated using Metropolis Hastings.

Key words: DSGE, Reverse mode differentiation, Hamiltonian Monte Carlo, No U-Turn Sampler, Bayesian estimation.

JEL Codes: C11,C13,C32.

The author would like to thank Sylvain Barde for helpful discussions and suggestions. All errors are my own.

[†] University of Kent, School of Economics. Email a.j.m.duncan@kent.ac.uk.

1 Introduction

This paper provides a reverse mode derivative, or *pullback*, for DSGE models. Reverse mode differentiation supports efficient calculation of gradients from the model likelihood to the model parameters, and enables the use of derivative based sampling algorithms including Hamiltonian Monte Carlo and the No U-Turn Sampler (NUTS hereafter) introduced by Hoffman and Gelman (2014).

An implementation in Julia language is provided along with benchmarks.[1] Our benchmarks use the small-scale New Keynesian model described by An and Schorfheide (2007). Our benchmarks show that MCMC chains generated by NUTS, which relies on our pullback, converge much more quickly than chains generated by Metropolis Hastings (MH hereafter), a popular sampling algorithm that does not rely on derivative information.

The closest related paper is Farkas and Tatar (2020), who demonstrate that the Binder and Pesaran (1997) DSGE solution algorithm can be algorithmically differentiated by reverse differentiation libraries. The method provided in this paper is independent of the solution algorithm, and can be used in combination with efficient eigensystem based solution algorithms.

2 The problem

We start with the Blanchard and Kahn (1980) canonical form of a log-linearised DSGE model. Our solution can be easily adapted to other canonical

[1] Codes are available at `https://github.com/alfredjmduncan/ReverseDiffDSGE.jl`

forms, including the Gensys (Sims 2002) and AIM (Anderson and Moore 1985) algorithms. Our derivation does not rely on any of the internal computations of the solution method.

$$B \begin{bmatrix} x' \\ \mathbb{E}[y'] \end{bmatrix} = A \begin{bmatrix} x \\ y \end{bmatrix} + G\varepsilon$$

The algorithm described by Blanchard and Kahn (BK hereafter) solves for N, L, C, D such that

$$y = -Nx - L\varepsilon$$
$$x' = Cx + D\varepsilon$$

In order to use gradient-based Markov chain Monte Carlo methods including NUTS, we must differentiate the likelihood ll of our model, conditional upon our observed data, with respect to the structural parameters of the model, Θ.

Θ	$\longrightarrow$	B, A, G	$\longrightarrow$	C, D, N, L	$\longrightarrow$	ll
Structural parameters	CF	Canonical form	BK	Solved model	Kalman filter	log-likelihood

We will calculate our derivatives using reverse mode differentiation. Kalman filter packages with support for reverse mode differentiation are widely available, and reverse mode differentiation of the canonical form is straightforward, or can be handled by modern modelling languages. To our knowledge, this paper is the first to provide a pullback for the solved DSGE with respect to its canonical form. Without loss of generality, we denote the pullback for B by $\bar{B}$, where $\bar{B}$ is the derivative of the log-likelihood ll with respect to B, expressed in terms of $\bar{C}, \bar{D}, \bar{N}, \bar{L}$. The matrix $\bar{B}$ shares the same dimensions as B.

Following Blanchard and Kahn (1980), we partition the matrices B, A, G as follows,

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}, \qquad A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \qquad G = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}$$

where B_{11}, A_{11} are $n_x \times n_x$, and G_1 is $n_x \times n_\varepsilon$.

Proposition 1 The DSGE pullback can be expressed as follows:

$\bar{A}$

$$\bar{A}_{11} = -\beta_1^{-1\prime}\left(\beta_2' W - \bar{C}\right) \qquad \bar{A}_{12} = -\beta_1^{-1\prime}\left(\beta_2' \bar{A}_{22} + \bar{C}N' + \bar{D}L'\right)$$

$$\bar{A}_{21} = W \qquad \bar{A}_{22} = -WN' - QL'$$

$\bar{B}$

$$\bar{B}_{11} = \beta_1^{-1\prime}\left(\beta_2' \bar{B}_{21} - (\bar{C}C' + \bar{D}D')\right) \qquad \bar{B}_{12} = -\bar{B}_{11}N'$$

$$\bar{B}_{21} = -WC' - QD' \qquad \bar{B}_{22} = -\bar{B}_{21}N'$$

$\bar{G}$

$$\bar{G}_1 = -\beta_1^{-1\prime}\left(\beta_2' Q + \bar{D}\right)$$

$$\bar{G}_2 = Q$$

where

$$\beta_1 = B_{11} - B_{12}N, \qquad \beta_2 = B_{21} - B_{22}N$$

and where W and Q can be expressed as follows:

$$\Gamma = -\left(C' \otimes (B_{22} - \beta_2\beta_1^{-1}B_{12}) + I_{n_x} \otimes (\beta_2\beta_1^{-1}A_{12} - A_{22})\right)'$$
$$w = \Gamma \backslash vec\left(\bar{N} + (B_{22} - \beta_2\beta_1^{-1}B_{12})'QD' + B_{12}'\beta_1^{-1\prime}(\bar{C}C' + \bar{D}D') - A_{12}'\beta_1^{-1\prime}\bar{C}\right)$$

$$W = \texttt{reshape}(w, n_y, n_x)$$

$$Q = -(\beta_2\beta_1^{-1}A_{12} - A_{22})' \backslash \left(\bar{L} - (\beta_1^{-1}A_{12})'\bar{D}\right)$$

The proof of Proposition 1 can be found in Appendix A.

3 Implementation and Benchmarks

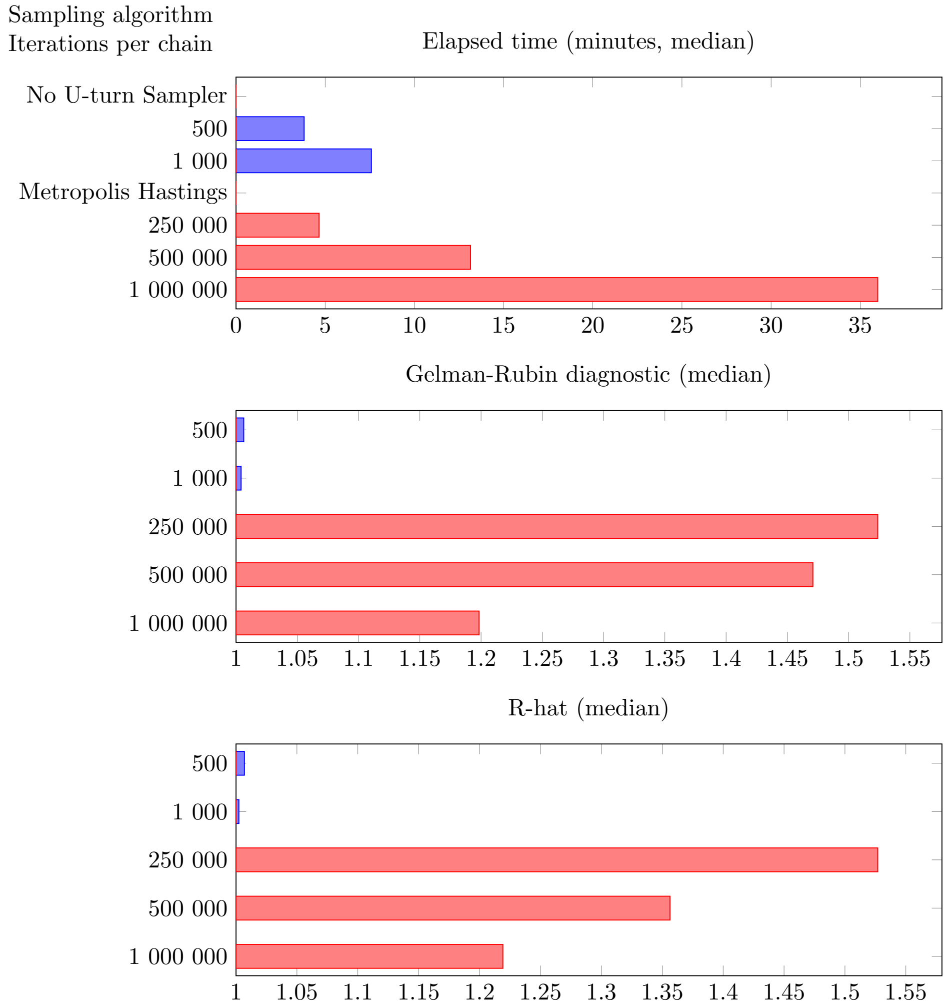
Our example implementation is written in Julia, using Zygote.jl for reverse mode differentiation, and Turing.jl for MCMC sampling. Our solution can be easily adapted to other programming languages with reverse differentiation libraries. The model used for all of our benchmarks is the small scale New Keynesian model of An and Schorfheide (2007). A full description is provided in Appendix B. Codes are available at `https://github.com/alfredjmduncan/ReverseDiffDSGE.jl`.

We provide an example estimation, drawing parameter values from the prior distribution of the An and Schorfheide (2007) model. We compare NUTS with MH, where NUTS utilises our pullback. For each sample, we draw parameter values from the prior distributions (based on those given by Herbst and Schorfheide (2016)), we then generate 100 periods of sample data, and estimate the model parameters on the sample data.

We record the Gelman-Rubin and R-hat convergence diagnostics (Gelman and Rubin 1992). The Gelman-Rubin diagnostic is a measure of the convergence of multiple parallel chains. The R-hat diagnostic is a test for non-stationarity within an MCMC chain. For both measures, values close to one indicate convergence of the MCMC chains. Figure 1 shows the results of our benchmarks. The No U-Turn Sampler takes much more time per sample than MH, but converges much more quickly.

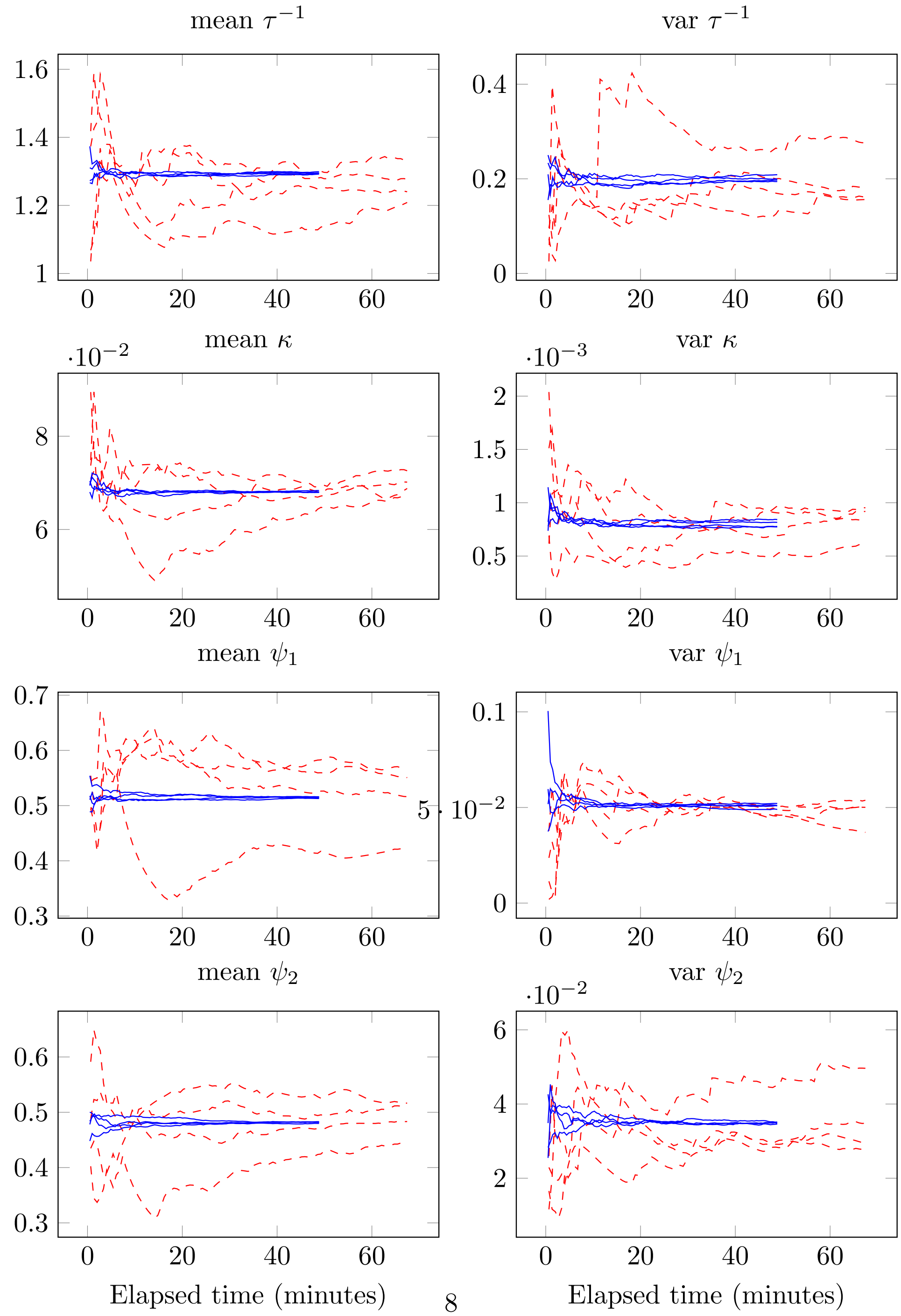
Next, we provide an example estimation based on a single draw from the prior distribution of the model. We generate 100 periods of simulated data, and estimate the model parameters using NUTS (with 10 000 iterations per chain) and MH (with 1 500 000 iterations per chain). Each estimation is computed with 4 chains. We plot the mean and variance of parameter estimates, by elapsed time, and by chain. The results of the exercise are shown in are shown Figure 2. The results give a visual representation of what we have seen in Figure 1; even after 1.5 million iterations, the MH chains have not converged. The No U-turn sampler chains converge very quickly.

Figure 1: Convergence diagnostics for NUTS and MH



Notes: Each sampler is run with four parallel chains. N = 20. For both the Gelman-Rubin diagnostic and the R-hat diagnostic, values close to one indicate convergence.

Figure 2: Mean and variance of posterior distributions for four parallel MCMC chains generated by NUTS (blue, solid) and MH (red, dashed).



References

An, Sungbae, and Frank Schorfheide. 2007. "Bayesian Analysis of DSGE Models." *Econometric Reviews* 26 (2-4): 113–72. https://doi.org/10.1080/07474930701220071.

Anderson, Gary, and George Moore. 1985. "A Linear Algebraic Procedure for Solving Linear Perfect Foresight Models." *Economics Letters* 17 (3): 247–52. https://EconPapers.repec.org/RePEc:eee:ecolet:v:17:y:1985:i:3:p:247-252.

Binder, Michael and Pesaran, M, (1997), Multivariate Linear Rational Expectations Models, Econometric Theory, 13, issue 6, p. 877-888.

Blanchard, Olivier Jean, and Charles M. Kahn. 1980. "The Solution of Linear Difference Models Under Rational Expectations." *Econometrica* 48 (5). [Wiley, Econometric Society]: 1305–11. http://www.jstor.org/stable/1912186.

Chen, Tianqi, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. 2015. "MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems."

Farkas, Mátyás and Tatar, Balint, 2020. "Bayesian estimation of DSGE models with Hamiltonian Monte Carlo," IMFS Working Paper Series 144, Goethe University Frankfurt, Institute for Monetary and Financial Stability (IMFS).

Ge, Hong, Kai Xu, and Zoubin Ghahramani. 2018. "Turing: A Language for

Flexible Probabilistic Inference." In *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, 1682–90. http://proceedings.mlr.press/v84/ge18b.html.

Gelman, Andrew, and Donald B. Rubin. 1992. "Inference from Iterative Simulation Using Multiple Sequences." *Statistical Science* 7 (4). Institute of Mathematical Statistics: 457–72. https://doi.org/10.1214/ss/1177011136.

Herbst, Edward, and Frank Schorfheide. 2016. *Bayesian Estimation of Dsge Models*. 1st ed. Princeton University Press.

Hoffman, Matthew D., and Andrew Gelman. 2014. "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo." *Journal of Machine Learning Research* 15 (47): 1593–1623. http://jmlr.org/papers/v15/hoffman14a.html.

Sims, Christopher A. 2002. "Solving Linear Rational Expectations Models." *Computational Economics* 20 (1-2): 1–20. https://ideas.repec.org/a/kap/compec/v20y2002i1-2p1-20.html.

Appendix A

Proof of Proposition 1

The DSGE has both a stable and an unstable solution. We rely on both to generate a set of necessary conditions for the solution of BK.

The stable solution is

$$x' = \underbrace{[B_{11} - B_{12}N]^{-1}[A_{11} - A_{12}N]}_{=C} x + \underbrace{[B_{11} - B_{12}N]^{-1}[G_1 - A_{12}L]}_{=D} \varepsilon.$$

The unstable solution is

$$[B_{21} - B_{22}N]x' = [A_{21} - A_{22}N]x + [G_2 - A_{22}L]\varepsilon.$$

Substituting the stable into the unstable solution we get

$$\begin{aligned}&[B_{21} - B_{22}N]\left([B_{11} - B_{12}N]^{-1}[A_{11} - A_{12}N]x + [B_{11} - B_{12}N]^{-1}[G_1 - A_{12}L]\varepsilon\right)\\ &\qquad = [A_{21} - A_{22}N]x + [G_2 - A_{22}L]\varepsilon\end{aligned}$$

The equation above holds for all x, ε. Collecting terms in x, ε,

$$[B_{21} - B_{22}N][B_{11} - B_{12}N]^{-1}[A_{11} - A_{12}N] = [A_{21} - A_{22}N] \qquad (1)$$

$$[B_{21} - B_{22}N][B_{11} - B_{12}N]^{-1}[G_1 - A_{12}L] = [G_2 - A_{22}L] \qquad (2)$$

The deterministic part (Pullback for N)

Equation 1 only relies on the deterministic part of the model, and is expressed solely in output matrix N. From (1), we derive the pushforward

$$\begin{aligned}
&\partial[A_{21} - A_{22}N] \\
&\quad = \partial[B_{21} - B_{22}N][B_{11} - B_{12}N]^{-1}[A_{11} - A_{12}N] \\
&\qquad - [B_{21} - B_{22}N][B_{11} - B_{12}N]^{-1}\partial[B_{11} - B_{12}N][B_{11} - B_{12}N]^{-1}[A_{11} - A_{12}N] \\
&\qquad + [B_{21} - B_{22}N][B_{11} - B_{12}N]^{-1}\partial[A_{11} - A_{12}N].
\end{aligned}$$

Let

$$\beta_2 := B_{21} - B_{22}N, \quad \beta_1 := B_{11} - B_{12}N, \quad \alpha_1 := A_{11} - A_{12}N.$$

Rearranging the pushforward, we have

$$\begin{aligned}
&(B_{22} - \beta_2\beta_1^{-1}B_{12})\dot{N}C + (\beta_2\beta_1^{-1}A_{12} - A_{22})\dot{N} \\
&\quad = (\dot{B}_{21} - \dot{B}_{22}N)C - \beta_2\beta_1^{-1}(\dot{B}_{11} - \dot{B}_{12}N)C + \beta_2\beta_1^{-1}(\dot{A}_{11} - \dot{A}_{12}N) - (\dot{A}_{21} - \dot{A}_{22}N).
\end{aligned}$$

Without loss of generality, let $x := \mathrm{vec}(X)$. Vectorising, we have

$$\Xi\dot{n} = (C' \otimes I_{n_y})\dot{b}_{21} - ((NC)' \otimes I_{n_y})\dot{b}_{22} - (C' \otimes \beta_2\beta_1^{-1})\dot{b}_{11} + ((NC)' \otimes \beta_2\beta_1^{-1})\dot{b}_{12}$$
$$+ (I_{n_x} \otimes \beta_2\beta_1^{-1})\dot{a_{11}} - (N' \otimes \beta_2\beta_1^{-1})\dot{a_{12}} - \dot{a_{21}} + (N' \otimes I_{n_y})\dot{a_{22}} \tag{3}$$

where

$$\Xi = C' \otimes (B_{22} - \beta_2\beta_1^{-1}B_{12}) + I_{n_x} \otimes (\beta_2\beta_1^{-1}A_{12} - A_{22}).$$

Introduce dummy $-z$ and take inner products w.r.t. both sides

$$\begin{aligned}\langle -z, \Xi\dot{n}\rangle = &\langle -z, (C' \otimes I_{n_y})\dot{b}_{21}\rangle - \langle -z, ((NC)' \otimes I_{n_y})\dot{b}_{22}\rangle \\ &- \langle -z, (C' \otimes \beta_2\beta_1^{-1})\dot{b}_{11}\rangle + \langle -z, ((NC)' \otimes \beta_2\beta_1^{-1})\dot{b}_{12}\rangle \\ &+ \langle -z, (I_{n_x} \otimes \beta_2\beta_1^{-1})\dot{a_{11}}\rangle - \langle -z, (N' \otimes \beta_2\beta_1^{-1})\dot{a_{12}}\rangle \\ &- \langle -z, \dot{a_{21}}\rangle + \langle -z, (N' \otimes I_{n_y})\dot{a_{22}}\rangle\end{aligned}$$

Isolate $\dot{b}, \dot{a}$ partials

$$\begin{aligned}\langle -\Xi' z, \dot{n}\rangle = &\langle -(C' \otimes I_{n_y})' z, \dot{b}_{21}\rangle + \langle ((NC)' \otimes I_{n_y})' z, \dot{b}_{22}\rangle \\ &+ \langle (C' \otimes \beta_2\beta_1^{-1})' z, \dot{b}_{11}\rangle + \langle -((NC)' \otimes \beta_2\beta_1^{-1})' z, \dot{b}_{12}\rangle \\ &+ \langle -(I_{n_x} \otimes \beta_2\beta_1^{-1})' z, \dot{a_{11}}\rangle + \langle (N' \otimes \beta_2\beta_1^{-1})' z, \dot{a_{12}}\rangle \\ &+ \langle z, \dot{a_{21}}\rangle + \langle -(N' \otimes I_{n_y})' z, \dot{a_{22}}\rangle\end{aligned}$$

Solve for the pullbacks

$$\bar{A}_{21} = \texttt{reshape}(-\Xi' \backslash \bar{n}, n_y, n_x)$$

$$\bar{A}_{22} = -\bar{A}_{21} N'$$

$$\bar{A}_{11} = -(\beta_2 \beta_1^{-1})' \bar{A}_{21}$$

$$\bar{A}_{12} = (\beta_2 \beta_1^{-1})' \bar{A}_{21} N'$$

$$\bar{B}_{21} = -\bar{A}_{21} C'$$

$$\bar{B}_{22} = \bar{A}_{21} C' N'$$

$$\bar{B}_{11} = (\beta_2 \beta_1^{-1})' \bar{A}_{21} C'$$

$$\bar{B}_{12} = -(\beta_2 \beta_1^{-1})' \bar{A}_{21} C' N'$$

To complete the derivation, follow the same steps for solution matrices L, C, D and canonical form matrix G.

Appendix B

The Small Scale New Keynesian model of An and Schorfheide (2007)

The small scale New Keynesian model used to compute the benchmarks listed in this paper is expressed in full as follows:

$$
\begin{aligned}
y_t &= \mathbb{E}_t[y_{t+1}] - \frac{1}{\tau}\left(R_t - \mathbb{E}_t[\pi_{t+1}] - \mathbb{E}_t[z_{t+1}]\right) + g_t - \mathbb{E}_t[g_{t+1}] \\
\pi_t &= \beta \mathbb{E}_t[\pi_{t+1}] + \kappa(y_t - g_t) \\
R_t &= \rho_R R_{t-1} + (1-\rho_R)(1+\psi_1)\pi_t + (1-\rho_R)\psi_2(y_t - g_t) + \sigma_R \varepsilon_{Rt} \\
z_t &= \rho_z z_{t-1} + \sigma_z \varepsilon_{zt} \\
g_t &= \rho_g g_{t-1} + \sigma_g \varepsilon_{gt} \\
\Delta y_t &= y_t - y_{t-1}
\end{aligned}
$$

where shock terms ε are i.i.d with mean zero and unit standard deviation. The observable variables are $\Delta y_t, R_t, \pi_t$, and observation errors are i.i.d with standard deviation 0.01. The prior distributions used for benchmarking exercises are as follows:

$1/\tau \sim$	InverseGamma(8,8)	$\kappa \sim$	Uniform(0.0,1.0)
$\psi_1 \sim$	Gamma(4,1/8)	$\psi_2 \sim$	Gamma(4,1/8)
$\rho_R \sim$	Uniform(0.5,0.9)	$\sigma_R \sim$	InverseGamma(4,0.32)
$\rho_g \sim$	Uniform(0.9,0.99)	$\sigma_g \sim$	InverseGamma(4,2.0)
$\rho_z \sim$	Uniform(0.9,0.99)	$\sigma_z \sim$	InverseGamma(4,0.5)