

Binning, Andrew

**Working Paper**

## An efficient application of the extended path algorithm in Matlab with examples

New Zealand Treasury Working Paper, No. 22/02

**Provided in Cooperation with:**

The Treasury, New Zealand Government

*Suggested Citation:* Binning, Andrew (2022) : An efficient application of the extended path algorithm in Matlab with examples, New Zealand Treasury Working Paper, No. 22/02, New Zealand Government, The Treasury, Wellington

This Version is available at:

<https://hdl.handle.net/10419/268088>

**Standard-Nutzungsbedingungen:**

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

**Terms of use:**

*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*



<https://creativecommons.org/licenses/by/4.0/>

# An Efficient Application of the Extended Path Algorithm in Matlab With Examples

Andrew Binning

New Zealand Treasury Working Paper 22/02

July 2022

## DISCLAIMER

The views, opinions, findings, and conclusions or recommendations expressed in this Working Paper are strictly those of the author(s). They do not necessarily reflect the views of the New Zealand Treasury or the New Zealand Government. The New Zealand Treasury and the New Zealand Government take no responsibility for any errors or omissions in, or for the correctness of, the information contained in these working papers. The paper is presented not as policy, but with a view to inform and stimulate wider debate.

**NZ TREASURY WORKING  
PAPER 22/02**

An Efficient Application of the Extended Path Algorithm in Matlab  
With Examples

**MONTH/YEAR**

July 2022

**AUTHORS**

Andrew Binning  
New Zealand Treasury  
No. 1 The Terrace  
Wellington  
New Zealand  
Email: andrew.binning@treasury.govt.nz  
Telephone: +64 4 917 6010

**URL**

Treasury website at July 2022:  
<https://www.treasury.govt.nz/publications/wp/wp-22-02>

**ACKNOWLEDGEMENTS**

I'd like to thank Mariano Kulish, Murat Özbilgin and Christie Smith for their useful comments. All remaining errors (if any) are my own.

**NZ TREASURY**

New Zealand Treasury  
PO Box 3724  
Wellington 6008  
NEW ZEALAND  
Email: information@treasury.govt.nz  
Telephone: +64 4 472 2733  
Website: [www.treasury.govt.nz](http://www.treasury.govt.nz)

# Abstract

Recent experience with interest rates hitting the effective lower bound and agents facing binding borrowing constraints has emphasised the importance of understanding the behaviour of an economy in which some variables may be restricted at times. The extended path algorithm is a commonly used and fairly general method for solving dynamic non-linear models with rational expectations. This algorithm can be used for a wide range of cases, including for models with occasionally binding constraints, or for forecasting with models in which some variables must satisfy a certain path. In this paper I propose computational improvements to the algorithm that speed up the calculations via vectorisations of the Jacobian matrix and residual equations. I illustrate the advantages of the method with a number of policy relevant applications: conditional forecasting with both exactly identified and underidentified shocks, occasionally binding constraints on interest rates, anticipated shocks, calendar-based forward guidance, optimal monetary policy with a binding constraint and transition paths.

## JEL CLASSIFICATION

C53  
C61  
C63  
E37  
E47

## KEYWORDS

Extended path algorithm; conditional forecasting;  
occasionally binding constraints; effective lower bound;  
optimal monetary policy; transition paths;

# Executive Summary

Recent experience with interest rates hitting the effective lower bound and agents facing binding borrowing constraints has emphasised the importance of understanding the behaviour of an economy in which some variables may be restricted at times. These non-trivial asymmetries are not easily handled in linear models with standard solution methods. The extended path algorithm is a commonly used and fairly general method for solving large non-linear models with rational (model consistent) expectations that can be used in a wide range of situations. Matlab is a commonly used software for structural macroeconomic modelling. In this paper I describe how Matlab code can be written that speeds up the implementation of the extended path algorithm. I show how the code can be applied to a number of policy relevant applications including conditional forecasting, where a subset of the forecasts are constrained to follow particular paths. This is particularly useful for forecasters who may want to judgementally adjust forecasts from a non-linear rational expectations model. I show how occasionally binding constraints can be implemented in the extended path algorithm in a way that results in significant speed gains. I also show how the algorithm can be adapted to handle optimal monetary policy under commitment with a binding lower bound constraint on nominal interest rates. Finally, I show how the code can be used to compute transitions paths, which can be used to show policy makers how an economy might adjust to permanent changes in government policy.

# Contents

<b>Abstract .....</b>	<b>ii</b>
<b>Executive Summary .....</b>	<b>iv</b>
<b>1 Introduction .....</b>	<b>1</b>
<b>2 A Simple New Keynesian DSGE Model .....</b>	<b>2</b>
<b>3 The Extended Path Algorithm.....</b>	<b>5</b>
3.1 The Deterministic Extended Path Algorithm.....	6
3.2 The Stochastic Extended Path Algorithm .....	8
<b>4 Computing Equation Residuals and Filling the Jacobian Matrix in Parallel.....</b>	<b>8</b>
4.1 A Sparse Banded Matrix Representation of the Jacobian .....	10
<b>5 Applications.....</b>	<b>12</b>
5.1 Conditional Forecasting with Exactly Identified Shocks .....	12
5.2 Conditional Forecasting with Underidentified Shocks.....	15
5.3 Matching Data.....	20
5.4 Occasionally Binding Constraints .....	22
5.4.1 A Lower Bound “Regime” – Model LBR.....	22
5.4.2 Anticipated Monetary Policy Shocks – Model APS.....	24
5.4.3 Comparing the Different Models with Occasionally Binding Constraints .....	25
5.4.4 Comparing the Endogenously Binding Lower Bound Constraint with Calendar- Based Odyssean Forward Guidance .....	29
5.4.5 Comparing the Non-linear Model with the Linearised Model .....	30
5.5 Optimal Monetary Policy Under Commitment with a Binding Lower Bound Constraint .....	31
5.6 Transition Paths .....	37
<b>6 Conclusion .....</b>	<b>39</b>
<b>A The Stochastic Extended Path Algorithm.....</b>	<b>42</b>
<b>B Speed Tests .....</b>	<b>43</b>
<b>C The Linearised Model and Occasionally Binding Constraints .....</b>	<b>45</b>

## List of Tables

Table 1 – Endogenous Model Variables .....	2
Table 2 – Shocks .....	3
Table 3 – Parameters.....	3
Table 4 – Calibrated Parameters .....	4
Table 5 – Estimated Parameters.....	5
Table 6 – Simple New Keynesian Model: Vectorised Code vs. For-Loops.....	43
Table 7 – Medium Sized Policy Model: Vectorised Code vs. For-Loops .....	43
Table 8 – Symbolic Derivatives With OBC’s vs. Quasi-Newton Methods With OBC’s .....	44
Table 9 – Linearised Model With OBC’s vs. Non-Linear Model With OBC’s .....	44

## List of Figures

Figure 1 – Output .....	14
Figure 2 – Model Variables .....	14
Figure 3 – Output .....	18
Figure 4 – Model Variables .....	19

Figure 5 – Fitted Shocks .....	21
Figure 6 – The Effective Lower Bound: Comparing Methods.....	26
Figure 7 – The Nominal Interest Rate vs The Shadow Interest Rate.....	26
Figure 8 – The Effective Lower Bound: Comparing Methods.....	28
Figure 9 – The Effective Lower Bound: Extended Path vs Perfect Foresight .....	30
Figure 10 – The Effective Lower Bound: Non-linear vs Linearised.....	31
Figure 11 – Optimal Policy: Model Variables .....	35
Figure 12 – Optimal Policy: Comparison with Simple Rule .....	36
Figure 13 – A Transition Path: Inflation.....	38
Figure 14 – A Transition Path: Model Variables .....	38
Figure 15 – The Sacrifice Ratio .....	39
Figure 16 – A Diagramatic Representation of the Stochastic Extended Path Algorithm.....	42

# An Efficient Application of the Extended Path Algorithm in Matlab With Examples

## 1. Introduction

Many problems in macroeconomics, like the occasionally binding lower bound constraint on interest rates or occasionally binding collateral constraints on household borrowing, can only be investigated using non-linear models. The extended path algorithm, originally due to Fair & Taylor (1983), is a reliable, flexible and frequently used method for solving large non-linear rational expectations models. Unlike other non-linear rational expectations solution methods the extended path algorithm is less prone to the curse of dimensionality as the size of the problem increases linearly in the number of endogenous variables in proportion to the solution horizon. Versions of it have been built into the TROLL programming language and the widely used Dynare toolbox. The extended path algorithm uses Newton's method to solve a deterministic problem or a sequence of deterministic problems to approximate rational expectations. This involves the calculation, at each iteration, of a large number of model equations, which are stacked into a vector, and a large Jacobian matrix containing the derivatives of this vector. In this paper I develop an efficient method in Matlab for vectorising the model equations and their derivatives which are used to construct a sparse Jacobian matrix. By vectorising this part of the code I save time that would otherwise be required to fill in and update these matrices using for-loops which are generally slower to implement in Matlab, and I avoid using the parallel toolbox. My code exploits the sparsity of the Jacobian matrix and makes use of symbolic derivatives. I calculate symbolic derivatives using Jakub Rysanek's symbolic differentiator, coded in Matlab, which is part of project Dyn:Ammo (Rysanek 2021). To demonstrate the flexibility of my extended path algorithm I apply it to six policy relevant problems faced by empirical economists and practitioners, namely conditional forecasting in non-linear rational expectations models with exactly identified shocks, conditional forecasting in non-linear rational expectations models with underidentified shocks, matching data, solving models with occasionally binding constraints, optimal monetary policy under commitment with a binding lower bound constraint and modelling transition paths.

As highlighted by Waggoner & Zha (1999), forecasters and policy makers are often faced by questions like "how do the forecasts of key macroeconomic variables change when forecasts of the interest rate change?". Conditional forecasting provides a solution to such problems by finding sequences of shocks that match a set of conditions imposed on a subset of the model's forecasts. I show how the extended path algorithm can be adapted to incorporate conditional forecasting in non-linear rational expectations models with both exactly identified (the same number of shocks as conditioning points) and underidentified shocks (more shocks than conditioning points). I demonstrate how the extended path algorithm can easily be used to back out the shocks of a non-linear model that matches historical data, which is both important for understanding the drivers of history, but also for producing counterfactual historical analysis and for bootstrapping exercises.<sup>1</sup>

Recent experience with both the lower bound on interest rates and collateral constraints on borrowing have demonstrated the importance of being able to incorporate occasionally

---

<sup>1</sup> See Andrieu & Hunt (2020) for examples of using the fitted shocks from the extended path algorithm in a bootstrapping exercise.



binding constraints in modelling analysis. The extended path algorithm provides a method for solving non-linear rational expectations models that are subject to occasionally binding constraints. Traditionally this has meant using quasi-Newton methods in the extended path algorithm due to the non-differentiable discontinuities associated with occasionally binding constraints. Quasi-Newton methods, which use an approximate Jacobian matrix, are generally slower to converge requiring more numerical operations than pure Newton methods. I show how occasionally binding constraints can be incorporated in the extended path algorithm using symbolic derivatives, resulting in code that runs more than two times faster than quasi-Newton methods. Optimal policy provides a useful counterfactual of what policy can achieve when set to minimise a policy maker's loss function, and my application of the extended path algorithm allows optimal policy to be solved in the presence of occasionally binding constraints. Policy makers are often concerned with how an economy will adjust to a permanent policy change. These adjustments can be captured by modelling transition paths using the extended path algorithm which I demonstrate in this paper.<sup>2</sup>

The rest of the paper is structured as follows: in Section 2, I describe the simple New Keynesian Dynamic Stochastic General Equilibrium (DSGE) model I use to illustrate the algorithm, and how the model can be re-written in a form that complies with the algorithm. In Section 3 I describe the deterministic and stochastic extended path algorithms. In Section 4 I show how the time paths for the model variables, equations and derivatives can be written in a way to exploit the element-wise operators in Matlab and then translated into a sparse Jacobian matrix. In Section 5 I demonstrate how the algorithm can be used for a number of exercises, including applications to conditional forecasting, matching data and finding shocks, occasionally binding constraints such as the effective lower bound, optimal monetary policy and modelling transition paths. Section 6 concludes.

## 2. A Simple New Keynesian DSGE Model

In this section I present the equilibrium conditions for a simple non-linear New Keynesian DSGE model that I use later to illustrate applications of my extended path algorithm in Matlab. The model is not too dissimilar from the one used in Ascari & Rossi (2012). To simplify the model I assume that firms are subject to quadratic price adjustment costs à la Rotemberg (1982). For the sake of brevity, I only present the model's equilibrium conditions without their derivation. The model consists of a New Keynesian IS curve, a New Keynesian Phillips curve, a Taylor-type monetary policy rule, autoregressive shock processes for both technology and preference shocks, and a white noise monetary policy shock. The (endogenous) model variables are listed in Table 1.

**Table 1 – Endogenous Model Variables**

Variable	Description
$Y_t$	Output
$\pi_t$	Gross inflation rate
$R_t$	Gross nominal interest rate
$A_t$	Technology shock process
$Z_t$	Preference shifter process

The model shocks (exogenous variables) are listed in Table 2.

<sup>2</sup> The algorithm and code presented in this paper was used to solve the stochastic neoclassical growth model used in the 2021 Long-Term Fiscal Statement. See Binning (2021) for details.

**Table 2 – Shocks**

Shock	Description
$\varepsilon_{R,t}$	Monetary policy shock
$\varepsilon_{A,t}$	Technology shock
$\varepsilon_{Z,t}$	Preference shifter shock

The model parameters and their descriptions are listed in Table 3.

**Table 3 – Parameters**

Parameter	Description
$\beta$	Time preference parameter
$\pi$	Steady state inflation rate/target
$\nu$	Elasticity of substitution between differentiated intermediate goods
$\kappa$	Weight on the disutility of working
$\chi$	Weight on habit formation
$\sigma$	Inverse of the intertemporal elasticity of substitution
$\eta$	Inverse of the Frisch elasticity of labour supply
$\phi$	Weight on quadratic price adjustment costs
$\psi$	Weight on price indexation
$\rho_R$	Interest rate smoothing term
$\kappa_\pi$	Weight on inflation in the Taylor-type rule
$\kappa_Y$	Weight on the output gap in the Taylor-type rule
$\rho_A$	Persistence parameter for total factor productivity process
$\rho_Z$	Persistence parameter for preference shifter
$\sigma_R$	Monetary policy shock standard deviation
$\sigma_A$	Technology shock standard deviation
$\sigma_Z$	Preference shock standard deviation

I list the model equations that describe equilibrium in the simple New Keynesian DSGE model, The New Keynesian IS curve:

$$Z_t (Y_t - \chi Y_{t-1})^{-\sigma} = E_t \left\{ \beta \frac{R_t}{\pi_{t+1}} Z_{t+1} (Y_{t+1} - \chi Y_t)^{-\sigma} \right\}, \quad (1)$$

The New Keynesian Phillips curve:

$$\begin{aligned} \left( \frac{\phi}{\nu - 1} \right) \pi_t \left[ \pi_t - \pi_t^\psi \pi^{1-\psi} \right] &= \left( \frac{\nu}{\nu - 1} \right) \kappa (Y_t - \chi Y_{t-1})^\sigma Y_t^\eta A_t^{-(1+\eta)} - 1 + \dots \\ \dots + E_t \left\{ \left( \frac{\phi}{\nu - 1} \right) \beta \left( \frac{Z_{t+1}}{Z_t} \right) \left( \frac{Y_{t+1} - \chi Y_t}{Y_t - \chi Y_{t-1}} \right)^{-\sigma} \left( \frac{Y_{t+1}}{Y_t} \right) \pi_{t+1} \left[ \pi_{t+1} - \pi_t^\psi \pi^{1-\psi} \right] \right\}, \end{aligned} \quad (2)$$

Taylor-type rule:

$$R_t = R_{t-1}^{\rho_r} \left( R \left( \frac{\pi_t}{\pi} \right)^{\kappa_\pi} \left( \frac{Y_t}{Y} \right)^{\kappa_Y} \right)^{1-\rho_r} \exp(\sigma_R \varepsilon_{R,t}), \quad (3)$$

Total factor productivity:

$$\log A_t = \rho_A \log A_{t-1} + \sigma_A \varepsilon_{A,t}, \quad (4)$$

Preference shifter:

$$\log Z_t = \rho_Z \log Z_{t-1} + \sigma_Z \varepsilon_{Z,t}. \quad (5)$$

This results in a system of five equations in five endogenous variables:  $Y_t$ ,  $\pi_t$ ,  $R_t$ ,  $A_t$ ,  $Z_t$ , and three exogenous variables:  $\varepsilon_{R,t}$ ,  $\varepsilon_{A,t}$ ,  $\varepsilon_{Z,t}$ . Note that the Rotemberg adjustment cost

is rebated to households along with firms' profits and debt is in zero net supply, so that in equilibrium output is equal to consumption. The dynamic model implies the following steady state solution:  $A_t = 1$ ,  $Z_t = 1$ ,  $\pi_t = \pi$ ,  $R_t = \frac{\pi_t}{\beta}$ ,  $Y_t = \left[ \left( \frac{v-1}{v} \right) \kappa^{-1} A_t^{1+\eta} (1-\chi)^{-\sigma} \right]^{\frac{1}{\sigma+\eta}}$ ,  $\varepsilon_{R,t} = 0$ ,  $\varepsilon_{A,t} = 0$ ,  $\varepsilon_{Z,t} = 0$ .<sup>3</sup>

Following Schmitt-Grohé & Uribe (2004) and Adjemian & Juillard (2010), among others, the set of equilibrium conditions for a wide range of dynamic non-linear rational expectations models can be written in the general form,

$$E_t \{f(x_{t+1}, x_t, x_{t-1}, \varepsilon_t)\} = 0, \quad (6)$$

where  $x_t$  is a vector of date  $t$  endogenous variables and  $\varepsilon_t$  is a vector of date  $t$  exogenous variables. The simple New Keynesian DSGE model can be recast in this form as follows,

$$E_t \{f(x_{t+1}, x_t, x_{t-1}, \varepsilon_t)\} = \begin{bmatrix} Z_t (Y_t - \chi Y_{t-1})^{-\sigma} - E_t \left\{ \beta \frac{R_t}{\pi_{t+1}} Z_{t+1} (Y_{t+1} - \chi Y_t)^{-\sigma} \right\} \\ \left( \frac{\phi}{v-1} \right) \pi_t \left[ \pi_t - \pi_{t-1}^{\psi} \pi^{1-\psi} \right] - \left( \frac{v}{v-1} \right) \kappa (Y_t - \chi Y_{t-1})^{\sigma} Y_t^{\eta} A_t^{-(1+\eta)} + 1 - \dots \\ \dots - E_t \left\{ \left( \frac{\phi}{v-1} \right) \beta \left( \frac{Z_{t+1}}{Z_t} \right) \left( \frac{Y_{t+1} - \chi Y_t}{Y_t - \chi Y_{t-1}} \right)^{-\sigma} \left( \frac{Y_{t+1}}{Y_t} \right) \pi_{t+1} \left[ \pi_{t+1} - \pi_t^{\psi} \pi^{1-\psi} \right] \right\} \\ R_t - R_{t-1}^{\rho_r} \left( R \left( \frac{\pi_t}{\pi} \right)^{\kappa \pi} \left( \frac{Y_t}{Y} \right)^{\kappa_Y} \right)^{1-\rho_r} \exp(\sigma_R \varepsilon_{R,t}) \\ \log A_t - \rho_A \log A_{t-1} - \sigma_A \varepsilon_{A,t} \\ \log Z_t - \rho_Z \log Z_{t-1} - \sigma_Z \varepsilon_{Z,t} \end{bmatrix} = 0, \quad (7)$$

where  $x_t = [Y_t, \pi_t, R_t, A_t, Z_t]^{\top}$ ,  $\varepsilon_t = [\varepsilon_{R,t}, \varepsilon_{A,t}, \varepsilon_{Z,t}]^{\top}$  and  $\top$  is the vector/matrix transpose operator. I approximate rational expectations with perfect foresight, which leads to,

$$f(E_t \{x_{t+1}\}, x_t, x_{t-1}, v_t) = \begin{bmatrix} Z_t (Y_t - \chi Y_{t-1})^{-\sigma} - \beta \frac{R_t}{\pi_{t+1}} Z_{t+1} (Y_{t+1} - \chi Y_t)^{-\sigma} \\ \left( \frac{\phi}{v-1} \right) \pi_t \left[ \pi_t - \pi_{t-1}^{\psi} \pi^{1-\psi} \right] - \left( \frac{v}{v-1} \right) \kappa (Y_t - \chi Y_{t-1})^{\sigma} Y_t^{\eta} A_t^{-(1+\eta)} + 1 - \dots \\ \dots - \left( \frac{\phi}{v-1} \right) \beta \left( \frac{Z_{t+1}}{Z_t} \right) \left( \frac{Y_{t+1} - \chi Y_t}{Y_t - \chi Y_{t-1}} \right)^{-\sigma} \left( \frac{Y_{t+1}}{Y_t} \right) \pi_{t+1} \left[ \pi_{t+1} - \pi_t^{\psi} \pi^{1-\psi} \right] \\ R_t - R_{t-1}^{\rho_r} \left( R \left( \frac{\pi_t}{\pi} \right)^{\kappa \pi} \left( \frac{Y_t}{Y} \right)^{\kappa_Y} \right)^{1-\rho_r} \exp(\sigma_R \varepsilon_{R,t}) \\ \log A_t - \rho_A \log A_{t-1} - \sigma_A \varepsilon_{A,t} \\ \log Z_t - \rho_Z \log Z_{t-1} - \sigma_Z \varepsilon_{Z,t} \end{bmatrix} = 0. \quad (8)$$

This is now a deterministic problem. Before solving and simulating the model, I need to set parameter values for the model. I used a mixed calibration and estimation strategy. Table 4 lists the parameters that I calibrate and their calibrated values,

**Table 4 – Calibrated Parameters**

$\beta$	$\pi$	$v$	$\kappa$
0.9975	1.005	6	2.6272

where I have set  $\kappa$  to ensure that  $Y_t = 1$  in the steady state (as per footnote 3). The remaining parameters are estimated using Bayesian methods on quarterly US data that spans the 1985Q1 to 2007Q4 period, where data includes demeaned interest rates (the fed

<sup>3</sup> I set  $\kappa = \left( \frac{v-1}{v} \right) (1-\chi)^{-\sigma}$ , so that  $Y_t = 1$  as a normalisation.

funds rate), demeaned (CPI) inflation and the output gap.<sup>4</sup> The output gap is constructed by taking logs of real GDP and removing a trend using the Hodrick Prescott filter with a smoothing parameter of 1600. I report the posterior modes which I use in the solution and simulation of the model in Table 5.

**Table 5 – Estimated Parameters**

$\chi$	$\sigma$	$\eta$	$\phi$	$\psi$	$\rho_R$	$\kappa_\pi$
0.48608	1.7249	0.82397	103.7	0.327	0.84974	1.502

$\kappa_Y$	$\rho_A$	$\rho_Z$	$\sigma_R$	$\sigma_A$	$\sigma_Z$
0.21104	0.3233	0.75172	0.038779	0.022062	0.001198

### 3. The Extended Path Algorithm

In this section I give a brief description of the extended path algorithm. Following Heer & Maussner (2009), I distinguish between the deterministic and the stochastic extended path algorithms. The deterministic extended path algorithm involves solving time paths for the variables where the entire sequence of shocks is revealed to agents in the first simulation period. In all other simulation periods the sequence of shocks is therefore perfectly anticipated. Under the stochastic extended path algorithm agents are surprised in each period by that period's shocks. This is the standard assumption made about unanticipated shocks when solving rational expectations models. This assumption is implemented in the stochastic extended path algorithm by solving a sequence of deterministic problems, one for each time period where the model is subject to shocks, where only shocks for the first simulation period are revealed to agents. The first simulation period corresponds to the solution for that time period, with projections of future model variables updated as shocks are revealed to agents. After solving the deterministic problem at a particular point in time, the model is rolled forward one period and the solution from the first simulation period in the previous iteration becomes the initial condition for the next iteration of the algorithm. This continues for the number of periods that the economy is subject to unexpected shocks. I start by presenting the pseudocode for the deterministic extended path algorithm. First, I list some notation and useful definitions for the deterministic extended path algorithm.

- $x_t$  is the  $n \times 1$  vector of date  $t$  endogenous variables and  $n$  is the number of endogenous variables in the model.
- $x_{1:T} = [x_1^\top, \dots, x_t^\top, \dots, x_T^\top]^\top$  is the  $n.T \times 1$  vector of endogenous variables and  $T$  is the solution horizon (not to be confused with  $\top$ , the matrix/vector transpose operator), which needs to be of sufficient length to ensure the model has converged to its steady state by period  $T + 1$ .
- $x_t^h$  is the  $h$ th iterate of  $x_t$  in Newton's method.
- $x_{1:T}^h = [(x_1^h)^\top, \dots, (x_t^h)^\top, \dots, (x_T^h)^\top]^\top$  is the  $n.T \times 1$  vector of endogenous variables evaluated at the  $h$ th iteration.
- $\varepsilon_t$  is the  $n_e \times 1$  vector of exogenous variables, where  $n_e$  is the number of exogenous variables.
- $\varepsilon_{1:T} = [\varepsilon_1^\top, \dots, \varepsilon_t^\top, \dots, \varepsilon_T^\top]^\top$  is the  $n_e.T \times 1$  vector of exogenous variables.

<sup>4</sup> I estimate the model in Matlab using the RISE toolbox, which can be downloaded from [https://github.com/jmaih/RISE\\_toolbox](https://github.com/jmaih/RISE_toolbox).

- $f(x_{t+1}, x_t, x_{t-1}, \varepsilon_t)$  is the  $n \times 1$  vector of date  $t$  model equations.
- $f(x_{t+1}^h, x_t^h, x_{t-1}^h, \varepsilon_t)$  is the  $n \times 1$  vector of date  $t$  model equations evaluated at the  $h$ th iterate of  $x$ .
- $x_0$  is the  $n \times 1$  vector of initial conditions.
- $x_{1:T}^0$  is the initial guess for the  $n.T \times 1$  vector of endogenous variables.
- $x_{T+1}$  is the  $n \times 1$  vector of terminal conditions. The terminal conditions need to be a steady state that is consistent with the dynamic model.<sup>5</sup>
- $\epsilon$  is the minimum tolerance for the difference between the  $h + 1$ th and  $h$ th iterates of  $x$ , and  $\delta$  is the minimum tolerance for the model equations holding (the equation residuals).

$$F(x_{1:T}^h, \varepsilon_{1:T}) = \begin{bmatrix} f(x_2^h, x_1^h, x_0, \varepsilon_1) \\ f(x_3^h, x_2^h, x_1^h, \varepsilon_2) \\ \vdots \\ f(x_{t+1}^h, x_t^h, x_{t-1}^h, \varepsilon_t) \\ \vdots \\ f(x_{T+1}^h, x_T^h, x_{T-1}^h, \varepsilon_T) \end{bmatrix},$$

is the  $n.T \times 1$  vector of model equations evaluated at the  $h$ th iteration of  $x_{1:T}$ , where  $x_0$  and  $x_{T+1}$  do not have  $h$  superscripts because they are given.

- $J(x_{1:T}^h; \varepsilon_{1:T}) = \frac{\partial F(x_{1:T}^h, \varepsilon_{1:T})}{\partial x_{1:T}^h}$  is the  $n.T \times n.T$  Jacobian matrix evaluated at the  $h$ th iterate of  $x$ .  $J(x_{1:T}^h; \varepsilon_{1:T})$  has a sparse banded representation.<sup>6</sup>

### 3.1 The Deterministic Extended Path Algorithm

The pseudocode for the deterministic extended path algorithm is presented in Algorithm 1.

---

#### Algorithm 1 The Deterministic Extended Path Algorithm

---

```

Set values for:  $x_0, x_{1:T}^0, x_{T+1}$ ,
Set  $h = 0$ 
while  $\|x_{1:T}^{h+1} - x_{1:T}^h\| > \epsilon$  and  $\|F(x_{1:T}^h, \varepsilon_{1:T})\| > \delta$  do
     $x_{1:T}^{h+1} = x_{1:T}^h - J(x_{1:T}^h, \varepsilon_{1:T})^{-1} F(x_{1:T}^h, \varepsilon_{1:T})$ 
     $h = h + 1$ 
end while
Obtain the date  $t$  values for  $x$ :  $x_{1:T} = x_{1:T}^h$ 

```

---

Because all shocks that hit the economy over the simulation/projection period are revealed to agents in the first projection period, and are then perfectly anticipated in all subsequent periods, the deterministic model only needs to be solved once. As a consequence, the time periods in the algorithm will also correspond exactly with the actual time periods. The  $n.T \times 1$  vector  $x_{1:T}$  can be reshaped into a  $T \times n$  matrix.

However, the standard assumption used when solving rational expectations models is to assume that shocks are only revealed to agents in the period that they hit the economy.

<sup>5</sup> For the examples that I cover in this paper, I assume that the terminal condition is a steady state. However in certain circumstances the extended path algorithm can handle unbalanced growth paths in which case the terminal condition need not be a steady state (see Maliar et al. 2020, for example).

<sup>6</sup> A sparse banded matrix has zeros off-diagonal and matrix blocks on-diagonal.

An approximate rational expectations solution can be found by solving a deterministic extended path problem for each period that shocks hit the economy, with the shocks only revealed to agents in the period they hit the economy. Following Heer & Maussner (2009) I refer to this procedure as the stochastic extended path algorithm.

The stochastic extended path algorithm requires the following additional definitions,

- $\tilde{x}_{k|t}$  is the  $n \times 1$  vector of endogenous variables in simulation period  $k$ , for  $k \in 1, \dots, T$ , for the deterministic problem solved in time period  $t$  with the information available at time period  $t$ .<sup>7</sup>
- $\tilde{x}_{k|t}^h$  is the  $h$ th iterate of  $\tilde{x}_{k|t}$  in Newton's method, for the deterministic problem solved in period  $t$ .
- $\tilde{x}_{1:T|t}^h = \left[ (\tilde{x}_{1|t}^h)^\top, \dots, (\tilde{x}_{k|t}^h)^\top, \dots, (\tilde{x}_{T|t}^h)^\top \right]^\top$  is the  $h$ th iterate of the  $n.T \times 1$  vector of endogenous variables for the deterministic problem solved in time period  $t$ .
- $f(\tilde{x}_{k+1|t}^h, \tilde{x}_{k|t}^h, \tilde{x}_{k-1|t}^h, \varepsilon_t)$  is the  $n \times 1$  vector of model equations in period  $k$ , for the deterministic problem solved in time period  $t$ , evaluated at the  $h$ th iterate of  $\tilde{x}$ .
- $\tilde{x}_{0|t}$  is the  $n \times 1$  vector of initial conditions, for the deterministic problem solved in time period  $t$ . For  $t > 1$ , I set  $\tilde{x}_{0|t} = \tilde{x}_{1|t-1}$ , so that the solution in the first simulation period from the deterministic problem solved in the previous time period becomes the initial condition for the current time period.
- $\tilde{x}_{1:T|t}^0$  is the initial guess for the  $n.T \times 1$  vector of endogenous variables, for the deterministic problem solved in time period  $t$ .
- $\tilde{x}_{T+1|t}$  is the  $n \times 1$  vector of terminal conditions, for the deterministic problem solved in time period  $t$ .

$$F(\tilde{x}_{1:T|t}^h, \varepsilon_t) = \begin{bmatrix} f(\tilde{x}_{2|t}^h, \tilde{x}_{1|t}^h, \tilde{x}_{0|t}, \varepsilon_t) \\ f(\tilde{x}_{3|t}^h, \tilde{x}_{2|t}^h, \tilde{x}_{1|t}, 0) \\ \vdots \\ f(\tilde{x}_{k+1|t}^h, \tilde{x}_{k|t}^h, \tilde{x}_{k-1|t}, 0) \\ \vdots \\ f(\tilde{x}_{T+1|t}, \tilde{x}_{T|t}^h, \tilde{x}_{T-1|t}, 0) \end{bmatrix},$$

is the  $n.T \times 1$  vector of model equations evaluated at the  $h$ th iteration, for the deterministic problem solved in time period  $t$ .

- $J(\tilde{x}_{1:T|t}^h, \varepsilon_t) = \frac{\partial F(\tilde{x}_{1:T|t}^h, \varepsilon_t)}{\partial \tilde{x}_{1:T|t}^h}$  is the  $n.T \times n.T$  Jacobian matrix, for the deterministic problem solved in time period  $t$ , evaluated at the  $h$ th iterate of  $\tilde{x}$  (i.e.  $\tilde{x}_{1:T|t}^h$ ).  $J(\tilde{x}_{1:T|t}^h, \varepsilon_t)$  has a sparse banded representation.

<sup>7</sup> I make the distinction between  $k$  which indexes simulation periods and  $t$  which indexes time periods. I retain  $T$  as the length of the simulation horizon. Converting simulation periods to time periods implies simulation period  $k$ , for a simulation constructed at time period  $t$ , is equal to time period  $t + k - 1$ , so that the first simulation period will be the solution for time period  $t$  when constructing the ex-post stochastic time series under rational expectations.

## 3.2 The Stochastic Extended Path Algorithm

The pseudocode for the stochastic extended path algorithm is presented in Algorithm 2, where  $s$  is the number of time periods that the model economy is hit by unexpected shocks.

---

### Algorithm 2 The Stochastic Extended Path Algorithm

---

```

Set initial values for:  $\tilde{x}_{0|1}, \tilde{x}_{1:T|1}^0, \tilde{x}_{T+1|1}$ 
for  $t = 1$  to  $s$  do
  Set  $h = 0$ 
  while  $\|\tilde{x}_{1:T|t}^{h+1} - \tilde{x}_{1:T|t}^h\| > \epsilon$  and  $\|F(\tilde{x}_{1:T|t}^h, \epsilon_t)\| > \delta$  do
     $\tilde{x}_{1:T|t}^{h+1} = \tilde{x}_{1:T|t}^h - J(\tilde{x}_{1:T|t}^h, \epsilon_t)^{-1} F(\tilde{x}_{1:T|t}^h, \epsilon_t)$ 
     $h = h + 1$ 
  end while
  Obtain the date  $t$  value for  $x$ :  $x_t = \tilde{x}_{1|t}^h$ 
  Update the initial condition for the next iteration:  $\tilde{x}_{0|t+1} = \tilde{x}_{1|t}^h$ 
  Update the initial guess for the next iteration:  $\tilde{x}_{1:T|t+1}^0 = \tilde{x}_{1:T|t}^h$ 
end for

```

---

This is illustrated diagrammatically in Figure 16 in Appendix A. The  $n.s \times 1$  vector of endogenous variables is given by,<sup>8</sup>

$$\begin{aligned}
 x_{1:s} &= [ (\tilde{x}_{1|1})^\top, \dots (\tilde{x}_{1|t})^\top, \dots (\tilde{x}_{1|s})^\top ]^\top, \\
 &= [ x_1^\top, \dots x_t^\top, \dots x_s^\top ]^\top.
 \end{aligned} \tag{9}$$

Equation (9) makes it clear that the time path for the endogenous variables, when subjected to a sequence of unexpected shocks, is the sequence of first (simulation) period solutions to a sequence of deterministic problems. The vector,  $x_{1:s}$ , can be reshaped into an  $s \times n$  matrix. The stochastic extended path algorithm can just be thought of as a sequence of deterministic extended path problems, where the model is only shocked in the first simulation period.

To ease up on notation and improve readability I will use notation for the deterministic extended path algorithm in subsequent sections, even though all the examples I investigate use the stochastic extended path algorithm.

## 4. Computing Equation Residuals and Filling the Jacobian Matrix in Parallel

In this section I show how the vector of model equations and the Jacobian matrix used in the extended path algorithm can be vectorised and calculated using elementwise matrix and vector operations. Using elementwise operations results in speed gains when the algorithm is coded in Matlab as the vectors can be calculated and filled in parallel. This avoids certain for-loops in the code, which are generally slower to implement in Matlab. It also avoids the explicit use of the parallel toolbox. In order to describe this algebraically, I introduce notation for the Hadamard product, Hadamard division and Hadamard powers. Specifically,

<sup>8</sup> I ignore the model projections in periods where no shocks are hitting the economy. These can easily be added by appending the entire  $n.T \times 1$  solution for the deterministic problem in time period  $s$  to the end of the stochastic extended path solution.

- Hadamard (elementwise) product:

$$A \odot B = C \text{ such that } a_{ij} \cdot b_{ij} = c_{ij}, \quad (10)$$

where  $A$ ,  $B$  and  $C$  are all  $m \times p$  matrices and  $a_{ij}$ ,  $b_{ij}$  and  $c_{ij}$  are the elements in the  $i$ th row and the  $j$ th column of  $A$ ,  $B$  and  $C$  respectively.

- Hadamard (elementwise) division:

$$A \oslash B = C \text{ such that } a_{ij}/b_{ij} = c_{ij}, \quad (11)$$

where  $A$ ,  $B$  and  $C$  are all  $m \times p$  matrices.

- Hadamard (elementwise) power:

$$A^{\circ(\alpha)} = C \text{ such that } a_{ij}^\alpha = c_{ij}, \quad (12)$$

where  $A$  and  $C$  are both  $m \times p$  matrices.

I use the following notation to represent the  $T \times 1$  vectors for each of the endogenous variables and shocks,

$$Q_{1:T} = \begin{bmatrix} Q_1 \\ \vdots \\ Q_T \end{bmatrix}, \quad Q_{0:T-1} = \begin{bmatrix} Q_0 \\ \vdots \\ Q_{T-1} \end{bmatrix}, \quad Q_{2:T+1} = \begin{bmatrix} Q_2 \\ \vdots \\ Q_{T+1} \end{bmatrix}. \quad (13)$$

In the context of the model presented in Section 2,

$Q_{1:T} \in \{A_{1:T}, Z_{1:T}, Y_{1:T}, R_{1:T}, \pi_{1:T}, \varepsilon_{R,1:T}, \varepsilon_{A,1:T}, \varepsilon_{Z,1:T}\}$  and  $Q_t \in \{A_t, Z_t, Y_t, R_t, \pi_t, \varepsilon_{R,t}, \varepsilon_{A,t}, \varepsilon_{Z,t}\}$ .

The  $n$  date  $t$  model equations can be expressed as,

$$f_t = f(x_{t+1}, x_t, x_{t-1}, \varepsilon_t) = \begin{bmatrix} g_1(x_{t+1}, x_t, x_{t-1}, \varepsilon_t) \\ \vdots \\ g_i(x_{t+1}, x_t, x_{t-1}, \varepsilon_t) \\ \vdots \\ g_n(x_{t+1}, x_t, x_{t-1}, \varepsilon_t) \end{bmatrix}, \quad (14)$$

where  $g_i(x_{t+1}, x_t, x_{t-1}, \varepsilon_t)$  is the  $i$ th model equation. I use  $g_i(x_{t+1}, x_t, x_{t-1}, \varepsilon_t)$  to represent the  $i$ th equation in period  $t$ , because  $f_t$  is used to represent the block of equations in the  $t$ th time period. Also note that  $g_i(x_{t+1}, x_t, x_{t-1}, \varepsilon_t)$  represents both the  $i$ th model equation in period  $t$  and the residual for the same equation in the same period. It then follows that the vector of residuals for the  $i$ th model equation in all  $T$  simulation periods can be written as,

$$g_{i,1:T} = g_i(x_{2:T+1}, x_{1:T}, x_{0:T-1}, \varepsilon_{1:T}) = \begin{bmatrix} g_i(x_2, x_1, x_0, \varepsilon_1) \\ \vdots \\ g_i(x_{t+1}, x_t, x_{t-1}, \varepsilon_t) \\ \vdots \\ g_i(x_{T+1}, x_T, x_{T-1}, \varepsilon_T) \end{bmatrix}. \quad (15)$$

Next I demonstrate how elementwise operations can be used to calculate the vector of equation residuals for each equation in the model using an example. The residual from equation (1), the first model equation in the simple New Keynesian DSGE model, in time period  $t$  is given by,

$$g_1 = Z_t (Y_t - \chi Y_{t-1})^{-\sigma} - \beta \frac{R_t}{\pi_{t+1}} Z_{t+1} (Y_{t+1} - \chi Y_t)^{-\sigma}. \quad (16)$$



Let  $g_{1,1:T}$  represent the equation residuals for the first equation, over all  $T$  simulation periods. Equation (16), the first equation in the model, can then be rewritten in all  $T$  periods using elementwise operations as follows,

$$g_{1,1:T} = Z_{1:T} \odot (Y_{1:T} - \chi \cdot Y_{0:T-1})^{\odot(-\sigma)} - \dots \\ \dots - \beta \cdot (R_{1:T} \odot \pi_{2:T+1}) \odot Z_{2:T+1} \odot (Y_{2:T+1} - \chi \cdot Y_{1:T})^{\odot(-\sigma)}. \quad (17)$$

In a second example I show how the  $T \times 1$  vector of equation residuals for the Taylor-type interest rate rule can be calculated using elementwise operations. The Taylor-type rule (3) is the third equation in the system, the equation residual in period  $t$  is given by,

$$g_3 = R_t - R_{t-1}^{\rho_r} \left( R \left( \frac{\pi_t}{\pi} \right)^{\kappa_\pi} \left( \frac{Y_t}{Y} \right)^{\kappa_Y} \right)^{1-\rho_r} \exp(\sigma_R \varepsilon_{R,t}). \quad (18)$$

Let  $g_{3,1:T}$  represent the equation residuals for the third equation, over all  $T$  simulation periods. Equation (18) can then be rewritten in all  $T$  periods as follows,

$$g_{3,1:T} = R_{1:T} - R_{0:T-1}^{\odot(\rho_r)} \odot \left( R \cdot (\pi_{1:T}/\pi)^{\odot(\kappa_\pi)} (Y_{1:T}/Y)^{\odot(\kappa_Y)} \right)^{\odot(1-\rho_r)} \odot \exp(\sigma_R \cdot \varepsilon_{R,1:T}). \quad (19)$$

It is relatively straight forward to write a parser to convert the model into this format and exploit Matlab's ability to calculate and fill the vectors and matrices in parallel using elementwise operations. The vector of equation residuals is constructed by reordering the elements from each equation vector by time as follows,

$$F(x) = F(x_{1:T}) = \begin{bmatrix} f_1 \\ \vdots \\ f_t \\ \vdots \\ f_T \end{bmatrix} = P \begin{bmatrix} g_{1,1:T} \\ \vdots \\ g_{i,1:T} \\ \vdots \\ g_{n,1:T} \end{bmatrix}, \quad (20)$$

where  $P$  is a permutation matrix,<sup>9</sup> and  $f_t$  is the model equations in period  $t$ , which is the same as equation (8) for the model used in this paper.

## 4.1 A Sparse Banded Matrix Representation of the Jacobian

Following Boucekkine (1995), the Jacobian matrix for the type of non-linear rational expectations models covered by the extended path algorithm takes the general form,

$$J(x_{1:T}^h) = \begin{bmatrix} B_1^h & C_1^h & 0 & \dots & & & & \\ A_2^h & B_2^h & C_2^h & 0 & \dots & & & \\ 0 & A_3^h & B_3^h & C_3^h & 0 & \dots & & \\ & & \ddots & \ddots & \ddots & & & \\ \dots & 0 & A_t^h & B_t^h & C_t^h & 0 & \dots & \\ & & & \ddots & \ddots & \ddots & & \\ & & \dots & 0 & A_{T-1}^h & B_{T-1}^h & C_{T-1}^h & \\ & & & \dots & 0 & A_T^h & B_T^h \end{bmatrix}, \quad (21)$$

where

$$B_1^h = \frac{\partial f(x_2^h, x_1^h, x_0, \varepsilon_t)}{\partial x_1^h}, \quad C_1^h = \frac{\partial f(x_2^h, x_1^h, x_0, \varepsilon_t)}{\partial x_2^h},$$

<sup>9</sup> I do not use a permutation matrix in the code, instead reordering the equation residuals using an index. Reordering is required to put the equation residuals back in the order they occur in the model and over time.

$$A_t^h = \frac{\partial f(x_{t+1}^h, x_t^h, x_{t-1}^h, \varepsilon_t)}{\partial x_{t-1}^h}, \quad B_t^h = \frac{\partial f(x_{t+1}^h, x_t^h, x_{t-1}^h, \varepsilon_t)}{\partial x_t^h}, \quad C_t^h = \frac{\partial f(x_{t+1}^h, x_t^h, x_{t-1}^h, \varepsilon_t)}{\partial x_{t+1}^h},$$

$$A_T^h = \frac{\partial f(x_{T+1}^h, x_T^h, x_{T-1}^h, \varepsilon_T)}{\partial x_{T-1}^h}, \quad B_T^h = \frac{\partial f(x_{T+1}^h, x_T^h, x_{T-1}^h, \varepsilon_T)}{\partial x_T^h}.$$

The matrix has a sparse banded representation, which I exploit. The sparse Jacobian matrix can be constructed using Matlab's sparse matrix function,

$$J(x_{1:T}^h) = \text{sparse} \left( r, c, \left[ \frac{\partial g_{1,1:T}^\top}{\partial (x_{1:T}^h)^\top}, \dots, \frac{\partial g_{i,1:T}^\top}{\partial (x_{1:T}^h)^\top}, \dots, \frac{\partial g_{n,1:T}^\top}{\partial (x_{1:T}^h)^\top} \right]^\top \right), \quad (22)$$

where  $r$  is a matrix of the row locations for the non-zero elements in the sparse matrix,  $c$  is a matrix of the column locations for the non-zero elements in the sparse matrix and,

$$\frac{\partial g_{i,1:T}}{\partial x_{1:T}^h} = \begin{bmatrix} \frac{\partial g_i(x_2^h, x_1^h, x_0, \varepsilon_1)}{\partial x_0} & \frac{\partial g_i(x_2^h, x_1^h, x_0, \varepsilon_1)}{\partial x_1^h} & \frac{\partial g_i(x_2^h, x_1^h, x_0, \varepsilon_1)}{\partial x_2^h} \\ \vdots & \vdots & \vdots \\ \frac{\partial g_i(x_{t+1}^h, x_t^h, x_{t-1}^h, \varepsilon_t)}{\partial x_{t-1}^h} & \frac{\partial g_i(x_{t+1}^h, x_t^h, x_{t-1}^h, \varepsilon_t)}{\partial x_t^h} & \frac{\partial g_i(x_{t+1}^h, x_t^h, x_{t-1}^h, \varepsilon_t)}{\partial x_{t+1}^h} \\ \vdots & \vdots & \vdots \\ \frac{\partial g_i(x_{T+1}^h, x_T^h, x_{T-1}^h, \varepsilon_T)}{\partial x_{T-1}^h} & \frac{\partial g_i(x_{T+1}^h, x_T^h, x_{T-1}^h, \varepsilon_T)}{\partial x_T^h} & \frac{\partial g_i(x_{T+1}^h, x_T^h, x_{T-1}^h, \varepsilon_T)}{\partial x_{T+1}^h} \end{bmatrix}. \quad (23)$$

The columns of  $\frac{\partial g_{i,1:T}}{\partial x_{1:T}^h}$  can be calculated in parallel using elementwise operations in a similar fashion to the calculation of the equation residuals as demonstrated earlier in this section, which greatly speeds up their calculation. The terms  $\frac{\partial g_i(x_2^h, x_1^h, x_0, \varepsilon_1)}{\partial x_0}$  and  $\frac{\partial g_i(x_{T+1}^h, x_T^h, x_{T-1}^h, \varepsilon_T)}{\partial x_{T+1}^h}$  are ignored as they are not used in the Jacobian matrix in equation (21).<sup>10</sup>

I use the default sparse matrix left division function in Matlab to invert the sparse Jacobian matrix, which uses the UMFPACK routines.<sup>11</sup>

I test the vectorised code against a version of the code where for-loops are used to fill both the vector of model equations (the equation residuals) and the Jacobian matrix. I compare the speed of both algorithms when solving and simulating the simple New Keynesian DSGE model described in Section 2 and a medium sized DSGE with 82 endogenous variables. When solving and simulating the simple New Keynesian DSGE model, the vectorised code is actually 2% slower than the code with for-loops. Closer inspection of the code reveals that while the vectorised code that fills and updates the equation residuals is faster than the respective code that performs the same task using for-loops, there is a speed gain from filling and updating the Jacobian matrix using a for-loop in place of vectorised code. The experiment was carried out using Matlab 2021a and reflects in part improvements Mathworks has made in the performance of for-loops. When the medium sized DSGE model is solved and simulated, the vectorised code is 13% faster than the code using for-loops. While 13% is a modest speed gain, this can add up to hours of savings if the model needs to be solved and simulated a large number of times as part of a Monte Carlo simulation exercise. The results are summarised in Appendix B along with the details of the simulation experiments.

<sup>10</sup> Note that I omit any columns of the matrix in equation (23) where the derivative is equal to zero.

<sup>11</sup> The profiler in Matlab reveals that this is the last remaining significant bottleneck in the code, typically taking up 70% of the computation time. While there are other sparse matrix solvers written by third parties in Matlab, (anecdotally) for most problems it is difficult to beat the performance of the native sparse matrix division operation in Matlab.

## 5. Applications

In this section, I show how to apply my efficient Matlab representation of the extended path algorithm to six problems: conditional forecasting with exactly identified shocks, conditional forecasting with underidentified shocks, matching data, occasionally binding constraints, optimal monetary policy under commitment with a binding lower bound constraint and modelling transition paths.<sup>12</sup>

### 5.1 Conditional Forecasting with Exactly Identified Shocks

In this first application I show how the simple New Keynesian DSGE model, and more generally dynamic non-linear rational expectations models, can be modified to work with the extended path algorithm to produce conditional forecasts using exactly identified shocks. As highlighted by Waggoner & Zha (1999), forecasters and policy makers are often concerned with questions like “how do the forecasts of key macroeconomic variables change when forecasts of the policy rate change?” In order to answer questions like these using a dynamic multivariate model, conditions need to be imposed on the future values of one or more of the model’s endogenous variables before the forecasts are made. The resulting forecasts are known as conditional forecasts.

Conditional forecasts are normally produced by finding sequences and/or combinations of shocks over the forecast period that match the model’s endogenous variables with the conditions imposed on the forecasts. In this example I consider the case where the shocks are exactly identified, that is the number of shocks used to match the conditions is equal to the number of conditions being imposed on the forecasts. In this case the shocks that match forecast conditions will be unique. Conditional forecasting is introduced into the model by endogenising the shocks used to match the conditioning information, for the number of periods the conditions are imposed. This is achieved by adding auxiliary variables to represent the endogenous shocks. Adding endogenous variables to the model also means adding auxiliary equations to the system. These additional equations exogenise the variables to be conditioned on in the conditioning periods. I introduce an exogenous indicator variable that determines when the conditions are imposed.

In this specific example I condition on an output path, by choosing a sequence of preference shifter shocks. The preference shifter shock was chosen primarily due to its proximity to output in the model. The same output path could be obtained by either endogenising monetary policy shocks or productivity shocks, with different consequences for the other endogenous variables in the model.<sup>13</sup> The process for conditioning on an interest rate or inflation track over the forecast period is exactly the same.

I start the augmentation of the model by replacing equation (5) with,

$$\log Z_t = \rho_Z \log Z_{t-1} + \tilde{\varepsilon}_{Z,t}, \quad (24)$$

where  $\tilde{\varepsilon}_{Z,t}$  is an endogenous variable that replaces  $\varepsilon_{Z,t}$  in equation (5). The additional endogenous variable means I need to add the following equation to the system,

$$X_t (Y_t - \mathbb{Y}_t) + (1 - X_t) (\tilde{\varepsilon}_{Z,t} - \varepsilon_{Z,t}) = 0, \quad (25)$$

which takes the form of an occasionally binding constraint, where  $X_t$  is an exogenous indicator variable that takes the value 1 in the periods where I condition on  $Y_t$  and 0 in

<sup>12</sup> Code for the applications can be downloaded from <https://sites.google.com/site/andrewbinningecon/research>.

<sup>13</sup> The set of preference shifter shocks required to match the forecast conditions is unique, as are the set of monetary policy shocks and productivity shocks that would match the same set of conditions on output.

all other periods. When the stochastic extended path algorithm is used and shocks are unanticipated,  $X_t$  will only take the value 1 in the first projection period, taking the value 0 in all other projection periods.  $\mathbb{Y}_t$  captures the conditions I impose on output in period  $t$ . When the constraint binds,  $Y_t$  is no longer endogenous, but forced to match  $\mathbb{Y}_t$ , and I solve for the endogenous shock  $\tilde{\varepsilon}_{Z,t}$  such that  $Y_t = \mathbb{Y}_t$ . In all other periods when the constraint does not bind the endogenous shock equals the exogenous shock, so that  $\tilde{\varepsilon}_{Z,t} = \varepsilon_{Z,t}$ . The vector of endogenous variables now becomes,

$$x_t = [ Y_t, \pi_t, R_t, A_t, Z_t, \tilde{\varepsilon}_{Z,t} ], \quad (26)$$

while the vector of exogenous variables becomes,

$$\varepsilon_t = [ \varepsilon_{R,t}, \varepsilon_{A,t}, \varepsilon_{Z,t}, X_t, \mathbb{Y}_t ]. \quad (27)$$

The system augmented with the endogenous preference shifter shock and the exogenous conditions on output now looks like,

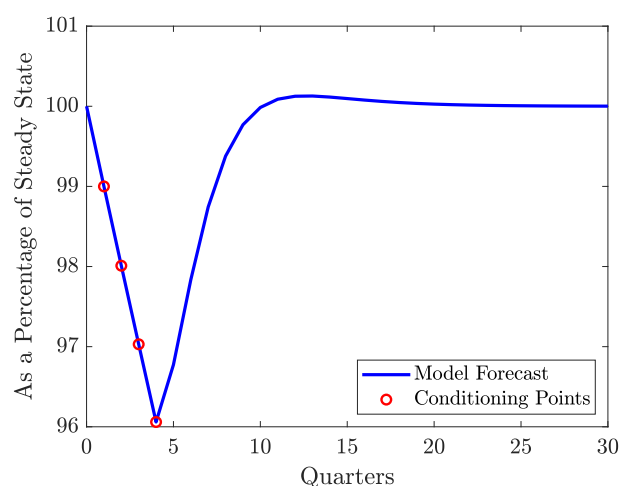
$$f(E_t \{x_{t+1}\}, x_t, x_{t-1}, \varepsilon_t) = \begin{bmatrix} Z_t (Y_t - \chi Y_{t-1})^{-\sigma} - \beta \frac{R_t}{\pi_{t+1}} Z_{t+1} (Y_{t+1} - \chi Y_t)^{-\sigma} \\ \left(\frac{\phi}{v-1}\right) \pi_t \left[ \pi_t - \pi_{t-1}^\psi \pi^{1-\psi} \right] - \left(\frac{v}{v-1}\right) \kappa (Y_t - \chi Y_{t-1})^\sigma Y_t^\eta A_t^{-(1+\eta)} + 1 - \dots \\ \dots - \left(\frac{\phi}{v-1}\right) \beta \left(\frac{Z_{t+1}}{Z_t}\right) \left(\frac{Y_{t+1} - \chi Y_t}{Y_t - \chi Y_{t-1}}\right)^{-\sigma} \left(\frac{Y_{t+1}}{Y_t}\right) \pi_{t+1} \left[ \pi_{t+1} - \pi_t^\psi \pi^{1-\psi} \right] \\ R_t - R_{t-1}^{\rho_r} \left( R \left(\frac{\pi_t}{\pi}\right)^{\kappa_\pi} \left(\frac{Y_t}{Y}\right)^{\kappa_Y} \right)^{1-\rho_r} \exp(\sigma_R \varepsilon_{R,t}) \\ \log A_t - \rho_A \log A_{t-1} - \sigma_A \varepsilon_{A,t} \\ \log Z_t - \rho_Z \log Z_{t-1} - \sigma_Z \tilde{\varepsilon}_{Z,t} \\ X_t (Y_t - \mathbb{Y}_t) + (1 - X_t) (\tilde{\varepsilon}_{Z,t} - \varepsilon_{Z,t}) \end{bmatrix} = 0. \quad (28)$$

To demonstrate the algorithm, I condition on an output path that falls by 1% each quarter for the first 4 quarters of the forecast. Starting from the steady state, where output is equal to 1, I condition on a GDP path equal to  $[0.99, 0.9801, 0.9703, 0.9606]^\top$ , and I let the algorithm find the sequence of preference shocks that matches this path exactly. At the end of the conditioning period, the model is free to transition back to the steady state. The exogenous indicator variable  $X_t$  takes the value 1 in the periods the conditions are imposed and 0 in all other periods. Consistent with rational expectations, I use the stochastic version of the algorithm when imposing conditions on GDP, which means only imposing the conditions one period at a time which also means solving the endogenous shocks one period at a time.<sup>14</sup> As the conditions are imposed for 4 quarters, 4 iterations of the time loop in the stochastic extended path algorithm are required to compute the time path.

The model's conditional forecast for output is plotted alongside the forecast conditions in Figure 1, while a subset of the forecasts for the other variables are plotted in Figure 2.

<sup>14</sup> Because the shocks are solved for period by period, the full sequence of shocks that match the forecast conditions ex post, will not be known to agents ex ante. Imposing the conditions on the first 4 periods, ex ante, by setting  $X_{1:4} = 1$  would make the new GDP path fully anticipated after the initial surprise in the first forecast period, consistent with the deterministic extended path algorithm. While this is not what I do in this example, it is something that can easily be handled with my solution routines.

**Figure 1 – Output**



**Figure 2 – Model Variables**

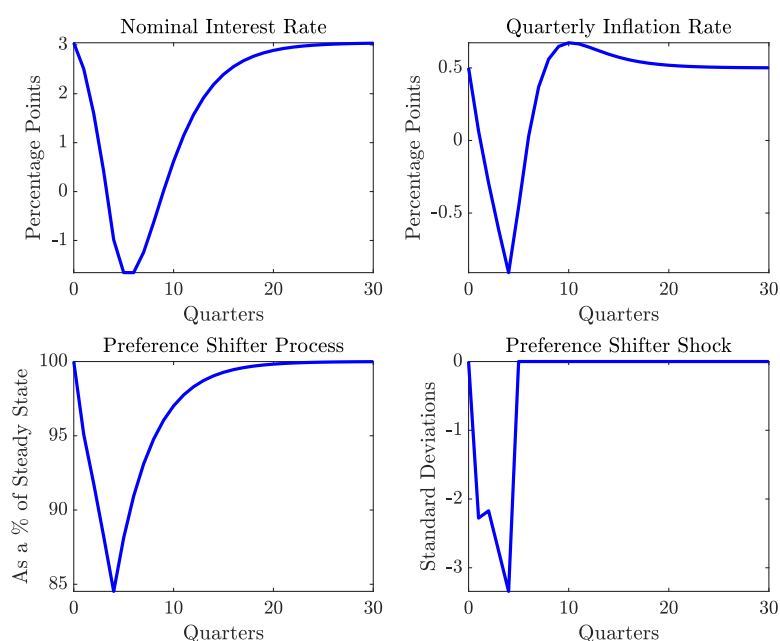


Figure 1 shows that the algorithm is able to match the conditions imposed on output over the first 4 quarters exactly. This is achieved by adding a sequence of preference shifter shocks to the first 4 forecast periods. The sequence of preference shifter shocks required to fit the conditions on the forecast implies weaker demand. Weaker demand is consistent with lower inflation. Following a Taylor-type rule, the central bank cuts interest rates in response to the negative output and inflation gaps. The bottom right panel in Figure 2 plots the sequence of shocks the algorithm finds that match the forecast conditions exactly. Because the shocks are exactly identified, this sequence is unique. Conditional forecasting with exactly identified shocks does not choose the most likely shock sequence (the shock sequence with the smallest variance). There may be other shock types, that could be used to fit this judgement, that have a smaller overall variance. This is illustrated more clearly in the next section where the same forecast conditions are fitted using multiple shock types.

As mentioned, the same output path could have been achieved using monetary policy

shocks or total factor productivity shocks. Using either of these shocks would mean matching the same output path for the first 4 quarters. However, the paths for the other endogenous variables would look very different, as would the path for output in the periods after the conditions are imposed. Hence the choice of shock determines the economic story. Using monetary policy shocks would require adding a sequence of positive shocks to raise interest rates. Higher interest rates would then make households less willing to spend and more willing to save, lowering consumption, output and inflation. Likewise, using productivity shocks would require adding a sequence of negative shocks to lower productivity levels and the level of output. This in turn would lead to higher inflation. The interest rate response would depend on the relative weights on inflation and the output gap (the implied output/inflation trade-off) in the Taylor-type rule.

## 5.2 Conditional Forecasting with Underidentified Shocks

Often it is the case that policy makers and/or practitioners want to condition on a particular path for an endogenous variable, but they do not know which shocks to use, or they would like to use a combination of shocks. This is where conditional forecasting with underidentified shocks comes into play. When the number of shocks used to match the conditioning information exceeds the number of conditions (the shocks are underidentified), there will potentially be an infinite number of shock combinations that match the conditioning information. Choosing the most likely combination of shocks in an econometric sense means choosing the shock combination with the smallest variance. This is a least squares approach and implies adding the smallest amount of judgement to match the conditions on the endogenous variables.<sup>15</sup> This has been handled in the linear case by Doan et al. (1983), Waggoner & Zha (1999), Beneš et al. (2008) and Maih (2010). I demonstrate how this can be handled more generally in the non-linear case using an adapted extended path algorithm. I start by simplifying my previous notation letting  $x = x_{1:T}$ . Note that the shocks that have been “endogenised” to replicate the conditional forecasts, such as  $\tilde{\varepsilon}_{Z,t}$  in equations (26) and (28) are elements of  $x_t$ .

The problem of finding the most likely combination of shocks involves minimising the weighted sum of squares of the model variables,  $q(x) = x^\top Hx$ , subject to the model’s non-linear equations holding in all periods. This can be written more formally as a minimization problem with a non-linear equality constraint, as follows,

$$\begin{aligned} \min_x \quad & x^\top Hx \\ \text{subject to} \quad & \tilde{F}(x) = 0, \end{aligned} \tag{29}$$

where  $\tilde{F}(x)$  is the  $(n.T - z + c) \times 1$  vector of non-linear equality constraints that come from the dynamic non-linear model,  $c$  is the number of conditions imposed on the forecasts and  $z \geq c$  is the number of shocks used to fit the conditional forecast,  $H$  is the  $n.T \times n.T$  Hessian matrix of the function  $q(x)$ . In this context,  $H$  can also be thought of as a diagonal selection matrix for the shocks used to fit the conditional forecasts.  $H$  is defined more explicitly in the example later in this section.

In order to solve this problem, I take a second order approximation of the objective function and a first order approximation of the equality constraint. The second order approximation

<sup>15</sup> In the linear case, the expected value from the full forecast density, in the absence of parameter uncertainty, should be the same as the unconditional forecast when no shocks are added. In the nonlinear case, the expected value from the full forecast density will not necessarily align with the no judgement point forecast. In this case the vector of shocks that produce the point forecast that matches the expected value from the full forecast density are probably the most likely outcome. This could be incorporated into the algorithm, but I do not do that here.

of  $q(x^{h+1})$  around the point  $x^h$  is given by,

$$q(x^{h+1}) = q(x^h + \Delta x^{h+1}) \approx (x^h)^\top H x^h + 2 (\Delta x^{h+1})^\top H x^h + (\Delta x^{h+1})^\top H \Delta x^{h+1}, \quad (30)$$

where  $\Delta x^{h+1} = x^{h+1} - x^h$  and  $H x^h$  is the gradient/Jacobian matrix evaluated at  $x^h$ . The first order approximation of the non-linear equality constraint around the point  $x^h$  is given by,

$$\tilde{F}(x^{h+1}) = \tilde{F}(x^h + \Delta x^{h+1}) \approx \tilde{F}(x^h) + \tilde{J}(x^h) \Delta x^{h+1}, \quad (31)$$

where  $\tilde{J}(x)$  is the  $(n.T - z + c) \times n.T$  matrix of first derivatives. Plugging the Taylor series approximations of the objective function and the linear constraint into (29) gives,

$$\min_{\Delta x^{h+1}} (x^h)^\top H x^h + 2 (\Delta x^{h+1})^\top H x^h + (\Delta x^{h+1})^\top H \Delta x^{h+1} \quad (32)$$

$$\text{subject to} \quad \tilde{F}(x^h) + \tilde{J}(x^h) \Delta x^{h+1} = 0.$$

Note that the minimisation is now in terms of  $\Delta x^{h+1}$ , the change in  $x$  between each iteration, rather than  $x^h$  or  $x^{h+1}$ . I minimise the objective function subject to the equality constraint and obtain the first order condition with respect to  $\Delta x^{h+1}$ ,

$$2H x^h + 2H \Delta x^{h+1} + \tilde{J}(x^h)^\top \lambda = 0, \quad (33)$$

where  $\lambda$  is the  $(n.T - z + c) \times 1$  vector of Lagrange multipliers on the equality constraints. The linear equality constraint implies,

$$\tilde{F}(x^h) + \tilde{J}(x^h) \Delta x^{h+1} = 0. \quad (34)$$

Combining equations (33) and (34) in matrix form gives,

$$\begin{bmatrix} 2H & \tilde{J}(x^h)^\top \\ \tilde{J}(x^h) & 0 \end{bmatrix}_{(n.T-z+c) \times (n.T-z+c)} \begin{bmatrix} \Delta x^{h+1} \\ \lambda \end{bmatrix} = \begin{bmatrix} -2H x^h \\ -\tilde{F}(x^h) \end{bmatrix}. \quad (35)$$

This can be re-written in the form of Newton's method as follows,

$$\begin{bmatrix} x^{h+1} \\ \lambda \end{bmatrix} = \begin{bmatrix} x^h \\ 0 \end{bmatrix}_{(n.T-z+c) \times 1} - \begin{bmatrix} 2H & \tilde{J}(x^h)^\top \\ \tilde{J}(x^h) & 0 \end{bmatrix}_{(n.T-z+c) \times (n.T-z+c)}^{-1} \begin{bmatrix} 2H x^h \\ \tilde{F}(x^h) \end{bmatrix}. \quad (36)$$

I demonstrate the algorithm with an example using the simple New Keynesian DSGE model. I augment the system of equations with exogenous forecast conditions on output,  $\mathbb{Y}_t$ , and three endogenous shocks,  $\tilde{\varepsilon}_{Z,t}$ ,  $\tilde{\varepsilon}_{A,t}$  and  $\tilde{\varepsilon}_{R,t}$ . That is, I condition on an output track, but let the algorithm find the combination of preference shocks, technology shocks and interest rate shocks that have the smallest combined variance. The new augmented system looks like,

$$f(E_t \{x_{t+1}\}, x_t, x_{t-1}, \varepsilon_t) = \begin{bmatrix} Z_t (Y_t - \chi Y_{t-1})^{-\sigma} - \beta \frac{R_t}{\pi_{t+1}} Z_{t+1} (Y_{t+1} - \chi Y_t)^{-\sigma} \\ \left(\frac{\phi}{v-1}\right) \pi_t \left[\pi_t - \pi_{t-1}^{\psi} \pi^{1-\psi}\right] - \left(\frac{v}{v-1}\right) \kappa (Y_t - \chi Y_{t-1})^\sigma Y_t^\eta A_t^{-(1+\eta)} + 1 - \dots \\ \dots - \left(\frac{\phi}{v-1}\right) \beta \left(\frac{Z_{t+1}}{Z_t}\right) \left(\frac{Y_{t+1} - \chi Y_t}{Y_t - \chi Y_{t-1}}\right)^{-\sigma} \left(\frac{Y_{t+1}}{Y_t}\right) \pi_{t+1} \left[\pi_{t+1} - \pi_t^{\psi} \pi^{1-\psi}\right] \\ R_t - R_{t-1}^{\rho_r} \left(R \left(\frac{\pi_t}{\pi}\right)^{\kappa_\pi} \left(\frac{Y_t}{Y}\right)^{\kappa_Y}\right)^{1-\rho_r} \exp(\sigma_R \tilde{\varepsilon}_{R,t}) \\ \log A_t - \rho_A \log A_{t-1} - \sigma_A \tilde{\varepsilon}_{A,t} \\ \log Z_t - \rho_Z \log Z_{t-1} - \sigma_Z \tilde{\varepsilon}_{Z,t} \\ X_t (Y_t - \mathbb{Y}_t) + (1 - X_t) (\tilde{\varepsilon}_{Z,t} - \varepsilon_{Z,t}) \\ (1 - X_t) (\tilde{\varepsilon}_{A,t} - \varepsilon_{A,t}) \\ (1 - X_t) (\tilde{\varepsilon}_{R,t} - \varepsilon_{R,t}) \end{bmatrix} = 0, \quad (37)$$



where again  $X_t$  is an indicator variable that takes the value 1 when the forecast conditions are imposed, so that  $Y_t = \mathbb{Y}_t$  and takes the value 0 when the forecast conditions do not hold so that the endogenous shocks equal the exogenous shocks,  $\tilde{\varepsilon}_{Z,t} = \varepsilon_{Z,t}$ ,  $\tilde{\varepsilon}_{A,t} = \varepsilon_{A,t}$  and  $\tilde{\varepsilon}_{R,t} = \varepsilon_{R,t}$ . The vector of endogenous variables now becomes,

$$x_t = [Y_t, \pi_t, R_t, A_t, Z_t, \tilde{\varepsilon}_{Z,t}, \tilde{\varepsilon}_{A,t}, \tilde{\varepsilon}_{R,t}]^\top, \quad (38)$$

and the vector of exogenous variables becomes,

$$\varepsilon_t = [\varepsilon_{R,t}, \varepsilon_{A,t}, \varepsilon_{Z,t}, X_t, \mathbb{Y}_t]^\top. \quad (39)$$

The Hessian matrix is zero everywhere except for the diagonal elements that correspond to the endogenous shocks in the periods the conditions are imposed. Just as I did in the first example with exactly identified shocks, I assume that the conditions are imposed period by period, consistent with unexpected shocks under rational expectations (i.e. I am using the stochastic extended path algorithm). This means in this example, ones are added in the 6th, 7th and 8th diagonal elements as the endogenous shocks are now the 6th, 7th and 8th endogenous variables,

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}. \quad (40)$$

The stacked vector of model equations now looks like,

$$\underset{(n.T-z+c) \times 1}{\tilde{F}(x)} = \begin{bmatrix} f_1 \\ \vdots \\ f_t \\ \vdots \\ f_T \end{bmatrix}. \quad (41)$$

In the first projection period the forecasting condition is imposed, which means  $X_1 = 1$ . In all subsequent projection periods  $X_t = 0$ . This means  $f_1$  is given by,

$$\underset{(n-z+c) \times 1}{f_1} = \begin{bmatrix} Z_1 (Y_1 - \chi Y_0)^{-\sigma} - \beta \frac{R_1}{\pi_2} Z_2 (Y_2 - \chi Y_1)^{-\sigma} \\ \left(\frac{\phi}{v-1}\right) \pi_t \left[\pi_1 - \pi_0^\psi \pi^{1-\psi}\right] - \left(\frac{v}{v-1}\right) \kappa (Y_1 - \chi Y_0)^\sigma Y_1^\eta A_1^{-(1+\eta)} + 1 - \dots \\ \dots - \left(\frac{\phi}{v-1}\right) \beta \left(\frac{Z_2}{Z_1}\right) \left(\frac{Y_2 - \chi Y_1}{Y_1 - \chi Y_0}\right)^{-\sigma} \left(\frac{Y_2}{Y_1}\right) \pi_2 \left[\pi_2 - \pi_1^\psi \pi^{1-\psi}\right] \\ R_1 - R_0^{\rho_r} \left(R \left(\frac{\pi_1}{\pi}\right)^{\kappa_\pi} \left(\frac{Y_1}{Y}\right)^{\kappa_Y}\right)^{1-\rho_r} \exp(\sigma_R \tilde{\varepsilon}_{R,1}) \\ \log A_1 - \rho_A \log A_0 - \sigma_A \tilde{\varepsilon}_{A,1} \\ \log Z_1 - \rho_Z \log Z_0 - \sigma_Z \tilde{\varepsilon}_{Z,1} \\ Y_1 - \mathbb{Y}_1 \end{bmatrix} \quad (42)$$

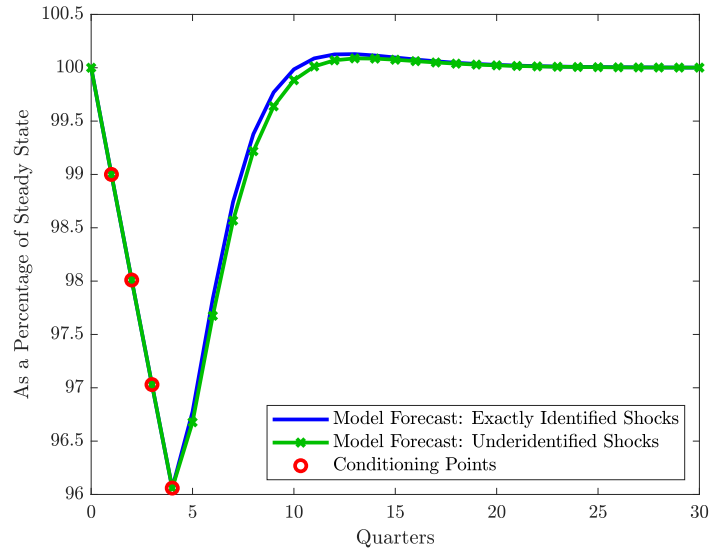


and  $f_t$  is given by,

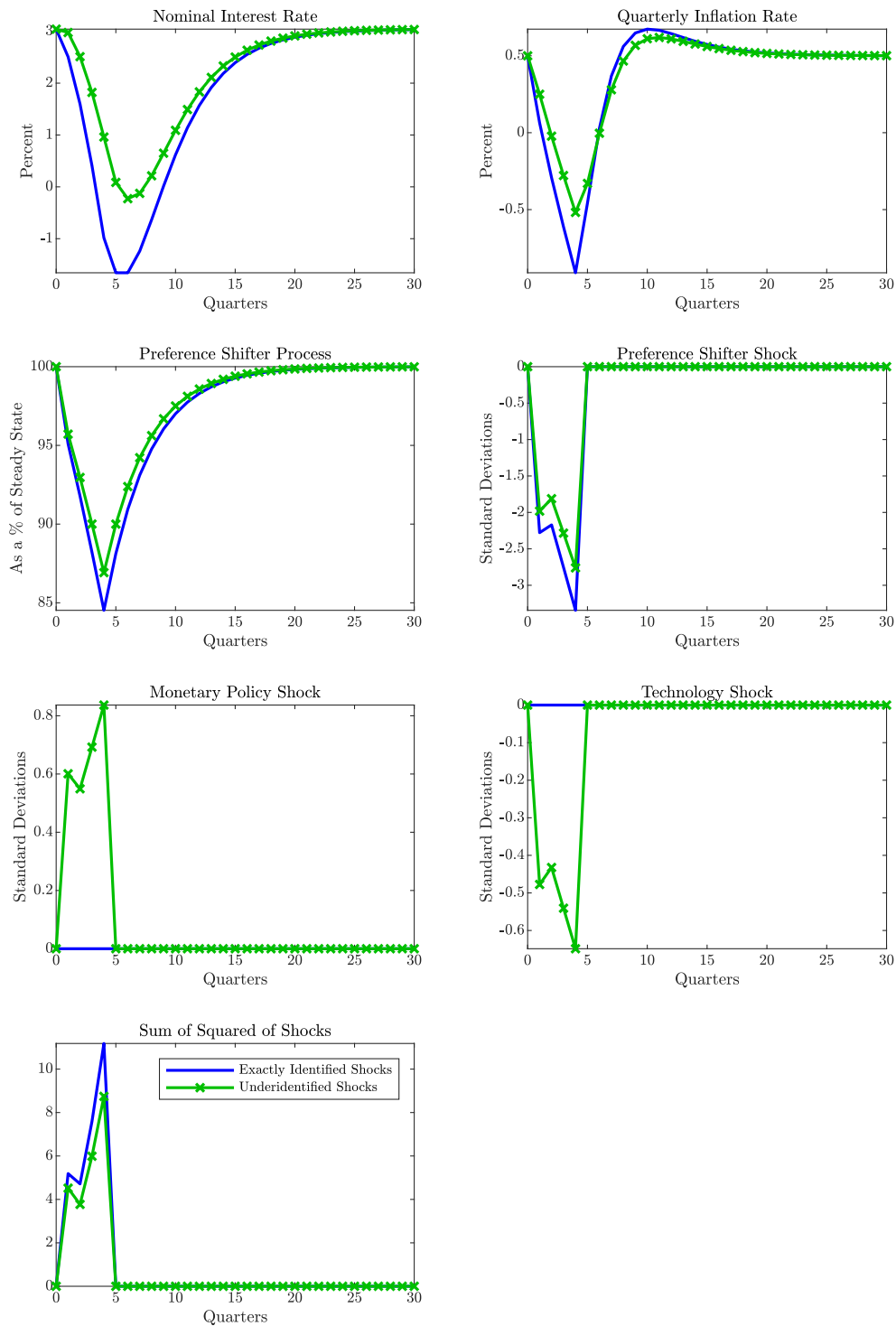
$$f_t = \begin{bmatrix} Z_t (Y_t - \chi Y_{t-1})^{-\sigma} - \beta \frac{R_t}{\pi_{t+1}} Z_{t+1} (Y_{t+1} - \chi Y_t)^{-\sigma} \\ \left( \frac{\phi}{v-1} \right) \pi_t \left[ \pi_t - \pi_{t-1}^\psi \pi^{1-\psi} \right] - \left( \frac{v}{v-1} \right) \kappa (Y_t - \chi Y_{t-1})^\sigma Y_t^\eta A_t^{-(1+\eta)} + 1 - \dots \\ \dots - \left( \frac{\phi}{v-1} \right) \beta \left( \frac{Z_{t+1}}{Z_t} \right) \left( \frac{Y_{t+1} - \chi Y_t}{Y_t - \chi Y_{t-1}} \right)^{-\sigma} \left( \frac{Y_{t+1}}{Y_t} \right) \pi_{t+1} \left[ \pi_{t+1} - \pi_t^\psi \pi^{1-\psi} \right] \\ R_t - R_{t-1}^{\rho_r} \left( R \left( \frac{\pi_t}{\pi} \right)^{\kappa_\pi} \left( \frac{Y_t}{Y} \right)^{\kappa_Y} \right)^{1-\rho_r} \exp(\sigma_R \tilde{\varepsilon}_{R,t}) \\ \log A_t - \rho_A \log A_{t-1} - \sigma_A \tilde{\varepsilon}_{A,t} \\ \log Z_t - \rho_Z \log Z_{t-1} - \sigma_Z \tilde{\varepsilon}_{Z,t} \\ \tilde{\varepsilon}_{Z,t} - \varepsilon_{Z,t} \\ \tilde{\varepsilon}_{A,t} - \varepsilon_{A,t} \\ \tilde{\varepsilon}_{R,t} - \varepsilon_{R,t} \end{bmatrix}. \quad (43)$$

For comparison I use the same forecast conditions on output from the previous example, this time I allow the algorithm to choose the combination of shocks with the smallest combined variance that satisfies the forecast conditions. The results for output using underidentified shocks are plotted next to the results using exactly identified shocks in Figure 3. The results for the other variables and the shocks for both methods are presented in Figure 4.

**Figure 3 – Output**



**Figure 4 – Model Variables**



The algorithm matches the forecast conditions exactly in the periods they are imposed. Rather than using a single shock the algorithm now uses a combination of the three model shocks to match the forecast conditions. These shocks are chosen to have the lowest combined variance. As a consequence the sequence of preference shocks required to fit the GDP profile is smaller in absolute value compared with the case where only the preference shifter shock is used to fit the forecast conditions. Positive monetary policy shocks are now added to reduce the fall in nominal interest rates consistent with the lower GDP track, along with a sequence of negative technology shocks.

The variance of the sequence of preference shifter shocks, when shocks are underidentified, is smaller than the variance of the shocks in the first exercise, when the preference shifter shock is exactly identified. This is illustrated in the bottom left pane of Figure 4 by the sum of squared monetary policy, preference shifter and technology shocks used at each point in time. When the forecasting conditions are imposed using only preference shifter shocks, the sum of squared shocks peaks at nearly 11 in the fourth period. When a mixture of shocks with the smallest combined variance is used, the sum of squared shocks peaks at about 9 in the fourth quarter and remains below the profile of the squared sum of shocks for the entire forecast horizon.

While I investigated conditional forecasting with underidentified and unanticipated shocks in the section, the algorithm can easily be modified to allow for anticipated shocks as is done in the linear case in Beneš et al. (2008) and Maih (2010). This would be equivalent to setting up and solving the model using the deterministic extended path algorithm. Extending the algorithm to incorporate anticipated shocks would require adding 1s in the locations of future endogenous shocks in the indicator variable  $X_t$  and the selection matrix  $H$ . The model equations would then take the same form as  $f_1$  in the periods that the shocks are anticipated.

### 5.3 Matching Data

In this example, I demonstrate how conditional forecasting with exactly identified shocks can be used to match the simple New Keynesian DSGE model with historical data. Matching the data is conceptually the same as the conditional forecasting procedure that has already been discussed. The procedure calculates historical shocks that are consistent with both the data and the model. I compare the fitted shocks from the extended path procedure with the fitted shocks from the linearised version of the model calculated using the Kalman filter. I match three observable variables from the model – the output gap, the inflation gap and the interest rate gap – to their counterparts in the data. Because I am matching three variables I need to endogenise at least three shocks to avoid a stochastic singularity. I use the algorithm with exactly identified shocks so I use exactly three shock types.<sup>16</sup> This means the vector of endogenous variables is now,

$$x_t = [ Y_t, \pi_t, R_t, A_t, Z_t, \tilde{\varepsilon}_{Z,t}, \tilde{\varepsilon}_{R,t}, \tilde{\varepsilon}_{A,t} ], \quad (44)$$

where  $\tilde{\varepsilon}_{Z,t}$ ,  $\tilde{\varepsilon}_{R,t}$  and  $\tilde{\varepsilon}_{A,t}$  are the endogenised preference shock, the endogenised monetary policy shock and the endogenised technology shock, respectively. I also need to exogenise the observed variables. The vector of exogenous variables now looks like,

$$\varepsilon_t = [ \varepsilon_{Z,t}, \varepsilon_{R,t}, \varepsilon_{A,t}, X_t, \hat{Y}_t, \hat{R}_t, \hat{\Pi}_t ], \quad (45)$$

where  $\hat{Y}_t = \log(\frac{Y_t}{Y})$  is the output gap,  $\hat{R}_t = R_t - R$  is the interest rate gap, and  $\hat{\Pi}_t = \Pi_t - \Pi$  is the inflation gap, all calculated from historical data. Again,  $X_t$  functions as an indicator variable that takes the value 1 in the periods the data is matched and 0 in all other periods.

<sup>16</sup> If more shocks than observable variables are used, I can use the extended path algorithm with underidentified shocks to find the fitted shocks over history.

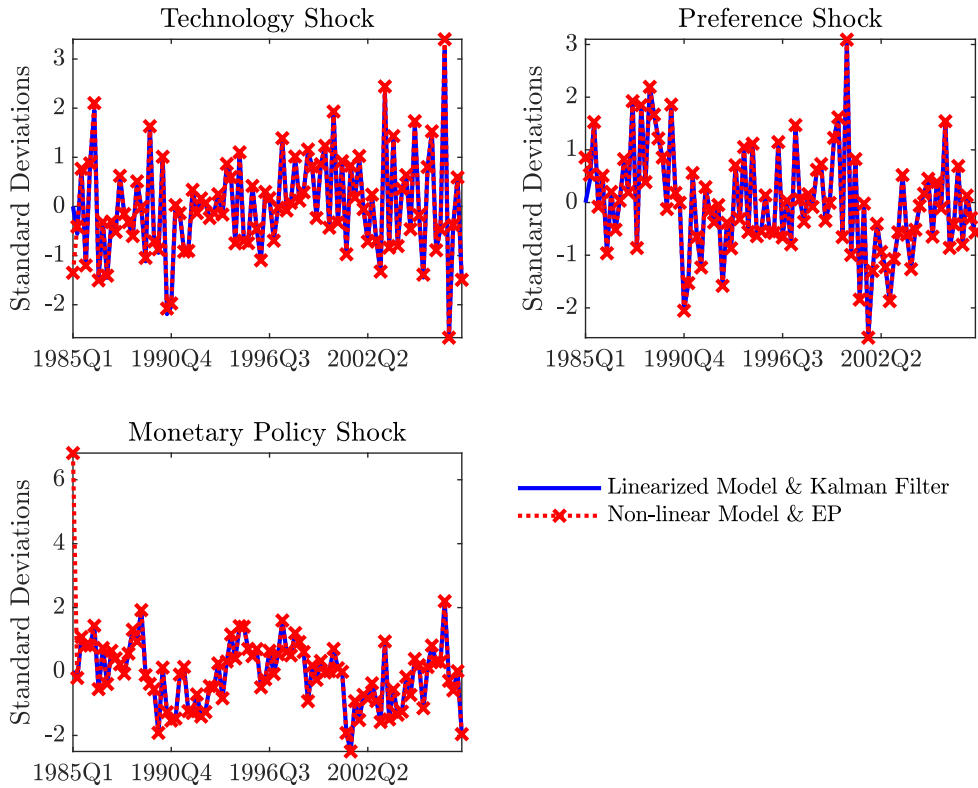
The augmented model now looks like,

$$f(E_t\{x_{t+1}\}, x_t, x_{t-1}, \varepsilon_t) = \begin{bmatrix} Z_t(Y_t - \chi Y_{t-1})^{-\sigma} - \beta \frac{R_t}{\pi_{t+1}} Z_{t+1}(Y_{t+1} - \chi Y_t)^{-\sigma} \\ \left(\frac{\phi}{v-1}\right) \pi_t \left[\pi_t - \pi_{t-1}^{1-\psi}\right] - \left(\frac{v}{v-1}\right) \kappa (Y_t - \chi Y_{t-1})^\sigma Y_t^\eta A_t^{-(1+\eta)} + 1 - \dots \\ \dots - \left(\frac{\phi}{v-1}\right) \beta \left(\frac{Z_{t+1}}{Z_t}\right) \left(\frac{Y_{t+1} - \chi Y_t}{Y_t - \chi Y_{t-1}}\right)^{-\sigma} \left(\frac{Y_{t+1}}{Y_t}\right) \pi_{t+1} \left[\pi_{t+1} - \pi_t^\psi \pi^{1-\psi}\right] \\ R_t - R_{t-1}^{\rho_r} \left(R \left(\frac{\pi_t}{\pi}\right)^{\kappa_\pi} \left(\frac{Y_t}{Y}\right)^{\kappa_Y}\right)^{1-\rho_r} \exp(\sigma_R \tilde{\varepsilon}_{R,t}) \\ \log A_t - \rho_A \log A_{t-1} - \sigma_A \tilde{\varepsilon}_{A,t} \\ \log Z_t - \rho_Z \log Z_{t-1} - \sigma_Z \tilde{\varepsilon}_{Z,t} \\ X_t \left(\log\left(\frac{Y_t}{Y}\right) - \hat{Y}_t\right) + (1 - X_t) (\tilde{\varepsilon}_{Z,t} - \varepsilon_{Z,t}) \\ X_t \left(R_t - R - \hat{R}_t\right) + (1 - X_t) (\tilde{\varepsilon}_{R,t} - \varepsilon_{R,t}) \\ X_t \left(\pi_t - \pi - \hat{\Pi}_t\right) + (1 - X_t) (\tilde{\varepsilon}_{A,t} - \varepsilon_{A,t}) \end{bmatrix} = 0, \quad (46)$$

where I match  $\log\left(\frac{Y_t}{Y}\right)$  with  $\hat{Y}_t$ ,  $R_t - R$  with  $\hat{R}_t$  and  $\pi_t - \pi$  with  $\hat{\Pi}_t$ . To capture rational expectations, I match data in the first period of each iteration of the extended path algorithm, so that shocks are only seen by agents in the period they hit the economy.

I use the same data the model was estimated on (described in Section 2) in this exercise. I plot the shocks calculated in the data matching exercise using the extended path algorithm against the fitted shocks from the linearised model, calculated using the Kalman filter, in Figure 5 below.<sup>17</sup>

**Figure 5 – Fitted Shocks**



The shocks fitted using the extended path and the Kalman filter are almost identical. This highlights two aspects worth mentioning. First, the simple New Keynesian DSGE

<sup>17</sup> The model is linearised, solved, Kalman filtered and smoothed using the RISE toolbox.

model, although non-linear, has dynamics that are close to linear for the shocks and parameterisation considered. Second, the extended path data fitting exercise is legitimate, and a good approximation of accepted methodologies like the Kalman filter, if the effects of uncertainty are not a primary concern.<sup>18</sup>

## 5.4 Occasionally Binding Constraints

In this section I show how occasionally binding constraints can be introduced into the model and handled using the extended path algorithm with symbolic derivatives. More specifically I model an occasionally binding lower bound constraint on interest rates. I demonstrate two equivalent approaches for modelling an endogenously binding lower bound and one approach for modelling an exogenously determined lower bound (otherwise known as calendar-based Odyssean forward guidance). A more detailed discussion of how the different methods of imposing the lower bound relate to the different types of forward guidance is given in Section 5.4.4. I compare the solution for the full non-linear model, with the results from a version of the algorithm where the lower bound constraint is imposed on the linearised model. This version of the algorithm may have advantages over the full non-linear model, especially where the model has dynamics close to linear in the absence of the constraint, the constraint is the main non-linearity and the model is large, so that computing the full non-linear solution is time consuming.

I start by explaining two different approaches for modelling an endogenously binding lower bound on interest rates. In the first method, the lower bound is characterised by a lower bound regime where interest rates are held constant at the lower bound for as long as the constraint binds. In the second approach, anticipated shocks are added to make the lower bound bind. In both cases, it is economic conditions that determine when the lower bound binds, making entry and exit to the lower bound endogenous.

### 5.4.1 A Lower Bound “Regime” – Model LBR

In order to allow the effective lower bound on interest rates to bind, I introduce the shadow interest rate,  $R_t^*$ , which is set according to the Taylor-type rule,<sup>19</sup>

$$R_t^* = (R_{t-1}^*)^{\rho_r} \left( R \left( \frac{\pi_t}{\pi} \right)^{\kappa_\pi} \left( \frac{Y_t}{Y} \right)^{\kappa_Y} \right)^{1-\rho_r} \exp(\sigma_R \varepsilon_{R,t}). \quad (47)$$

I introduce an additional equation that determines when the lower bound does and does not bind, which takes the form,

$$R_t - R_{ELB} = X_t (R_t^* - R_{ELB}), \quad (48)$$

which is equivalent to  $R_t = X_t R_t^* + (1 - X_t) R_{ELB}$ , where  $X_t$  is an endogenous indicator variable, such that,

$$X_t = \begin{cases} 0 & \text{if } R_t^* \leq R_{ELB}, \\ 1 & \text{if } R_t^* > R_{ELB}. \end{cases} \quad (49)$$

This set up is similar to the way conditions were imposed on the forecasts in the conditional forecasting examples. The key difference being that the indicator,  $X_t$ , which determines when the constraint does and does not bind, is now endogenous.

<sup>18</sup> This is because the extended path algorithm assumes certainty equivalence so that agents do not take into account shock volatility when making decisions.

<sup>19</sup> I set the smoothing term in the shadow rate to the lag of the shadow rate. This could be set to the actual interest rate if preferred. Hills & Nakata (2018) discuss the consequences of this choice with regard to the size of fiscal multipliers at the effective lower bound.

Substituting equations (47) and (48) into (8) gives the augmented system,

$$f(E_t \{x_{t+1}\}, x_t, x_{t-1}, \varepsilon_t) = \begin{bmatrix} Z_t (Y_t - \chi Y_{t-1})^{-\sigma} - \beta \frac{R_t}{\pi_{t+1}} Z_{t+1} (Y_{t+1} - \chi Y_t)^{-\sigma} \\ \left(\frac{\phi}{v-1}\right) \pi_t \left[ \pi_t - \pi_{t-1}^\psi \pi^{1-\psi} \right] - \left(\frac{v}{v-1}\right) \kappa (Y_t - \chi Y_{t-1})^\sigma Y_t^\eta A_t^{-(1+\eta)} + 1 - \dots \\ \dots - \left(\frac{\phi}{v-1}\right) \beta \left(\frac{Z_{t+1}}{Z_t}\right) \left(\frac{Y_{t+1} - \chi Y_t}{Y_t - \chi Y_{t-1}}\right)^{-\sigma} \left(\frac{Y_{t+1}}{Y_t}\right) \pi_{t+1} \left[ \pi_{t+1} - \pi_t^\psi \pi^{1-\psi} \right] \\ R_t^* - (R_{t-1}^*)^{\rho_r} \left( R \left(\frac{\pi_t}{\pi}\right)^{\kappa_\pi} \left(\frac{Y_t}{Y}\right)^{\kappa_Y} \right)^{1-\rho_r} \exp(\sigma_R \varepsilon_{R,t}) \\ \log A_t - \rho_A \log A_{t-1} - \sigma_A \varepsilon_{A,t} \\ \log Z_t - \rho_Z \log Z_{t-1} - \sigma_Z \varepsilon_{Z,t} \\ R_t - R_{ELB} - X_t (R_t^* - R_{ELB}) \end{bmatrix} = 0. \quad (50)$$

Note that I leave  $X_t$  in the model at this point, treating it as exogenous until the symbolic (analytical) derivatives have been calculated. After the symbolic derivatives have been calculated, I substitute  $R_t^* > R_{ELB}$  for  $X_t$  in the model equations and the Jacobian matrix.  $R_t^* > R_{ELB}$  then takes the value 1 in Matlab when true and 0 otherwise, ensuring that in the case of the Jacobian matrix the symbolic derivatives are either zeroed out or not, as appropriate.<sup>20</sup>

This sequence of events reflects the fact that the indicator function cannot be differentiated, because it is an inequality. Moreover, derivatives of the endogenous switching mechanism are not required to solve the model. It is the switches themselves that are important and these can be added to the model and the Jacobian once the analytical Jacobian matrix has been calculated. Because derivatives of the switching mechanism are not required to solve the model, symbolic derivatives can be used in place of numerical derivatives to solve models with occasionally binding constraints. This has significant computational advantages over quasi-Newton methods.<sup>21</sup>

Unlike popular piece-wise linear solution methods like OccBin, which require a guess and check verification on the number of periods the constraint binds, the extended path algorithm is able to solve for both the forward paths of the variables and the duration of the constraint binding in the same step. This is because the duration of the constraint binding is a function of the solution paths and each Newton iteration is an update of both the solution paths and the duration of the constraint binding. These iterations continue until the equation residuals are sufficiently small and the duration of the constraint binding is found.

To demonstrate the computational advantages of this solution procedure I run a simple simulation experiment using vectorised code implemented using symbolic derivatives for the simple New Keynesian model with a binding lower bound constraint on interest rates described in this section. I compare the results against the same model and sequence of shocks solved and simulated using vectorised code and a sparse implementation of Broyden's method (a quasi-Newton method). The vectorised code implemented using symbolic derivatives is 63% faster than the sparse implementation of Broyden's method. This is a more significant speed gain and could save hours in computation when the model needs to be solved and simulated a large number of times as part of a Monte Carlo simulation exercise. A full description of the simulation experiment and the results can be found in Appendix B.

<sup>20</sup> In vectorised form,  $X_{1:T}$  in  $R_{1:T} - R_{ELB} - X_{1:T} \odot (R_{1:T}^* - R_{ELB})$  is replaced with  $R_{1:T}^* > R_{ELB}$ .

<sup>21</sup> The endogenous switching mechanism plays the same role if the model is solved using quasi-Newton methods. The quasi-Newton method does not take derivatives with respect to the switching mechanism itself, but the parts of the equation that have been switched on or switched off. This is because the indicator function either takes the value 1 or 0 and is invariant to small changes in the endogenous variables used to calculate the numerical derivatives.

### 5.4.2 Anticipated Monetary Policy Shocks – Model APS

The lower bound on interest rates, and occasionally binding constraints in general, can also be imposed using anticipated shocks, just as Laséen & Svensson (2011), Holden & Paetz (2012) and de Groot et al. (2021) have done. In this example I continue using  $R_t^*$  as the shadow interest rate which is set according to (47). I introduce an additional equation that maps the shadow rate to the actual interest rate as follows,

$$R_t = R_t^* \exp(\tilde{\varepsilon}_{R,t}), \quad (51)$$

where  $\tilde{\varepsilon}_{R,t}$  is an endogenously determined anticipated monetary policy shock. In the LBR model the lower bound is introduced as a constant interest rate that is expected to bind for as long as economic conditions are expected to support a negative shadow interest rate, as determined by the Taylor-type rule in equation (47). The same effect is achieved in the APS model by introducing anticipated monetary policy shocks that prevent the interest rate from violating the lower bound constraint. As long as the shadow rate of interest is expected to be below the lower bound, then anticipated monetary policy shocks will be required to enforce the lower bound constraint. And while these monetary policy shocks are anticipated so that they are known for a number of periods before they actually hit the economy, the sequence of shocks will still be a surprise to agents in the first period that the lower bound binds or is expected to bind. Further unanticipated shocks that hit the economy over the lower bound period and change agents' expectations about how long the shadow rate will be negative will also change the sequence of anticipated shocks required to enforce the lower bound constraint. This is illustrated graphically in Figure 8 in the Section 5.4.3. The constraint on interest rates is introduced by adding the equation,

$$X_t \tilde{\varepsilon}_{R,t} + (1 - X_t)(R_t - R_{ELB}) = 0, \quad (52)$$

in place of (48), so that anticipated monetary policy shocks are added to the model when the lower bound constraint is expected to bind. As in the previous example,  $X_t$  is an endogenous indicator variable, such that,

$$X_t = \begin{cases} 0 & \text{if } R_t^* \leq R_{ELB}, \\ 1 & \text{if } R_t^* > R_{ELB}. \end{cases} \quad (53)$$

Equation (52) can also be thought of as a complementary slackness condition in a constrained optimisation problem, because it implies that,

$$\tilde{\varepsilon}_{R,t}(R_t - R_{ELB}) = 0, \quad (54)$$

in which case the anticipated monetary policy shock can be interpreted as the Lagrange multiplier on the lower bound constraint, as has been pointed out by Lindé et al. (2016).

The new augmented system of equations is given by,

$$f(E_t\{x_{t+1}\}, x_t, x_{t-1}, \varepsilon_t) = \begin{bmatrix} Z_t(Y_t - \chi Y_{t-1})^{-\sigma} - \beta \frac{R_t}{\pi_{t+1}} Z_{t+1}(Y_{t+1} - \chi Y_t)^{-\sigma} \\ \left(\frac{\phi}{v-1}\right) \pi_t \left[ \pi_t - \pi_{t-1}^{\psi} \pi^{1-\psi} \right] - \left(\frac{v}{v-1}\right) \kappa (Y_t - \chi Y_{t-1})^{\sigma} Y_t^{\eta} A_t^{-(1+\eta)} + 1 - \dots \\ \dots - \left(\frac{\phi}{v-1}\right) \beta \left(\frac{Z_{t+1}}{Z_t}\right) \left(\frac{Y_{t+1} - \chi Y_t}{Y_t - \chi Y_{t-1}}\right)^{-\sigma} \left(\frac{Y_{t+1}}{Y_t}\right) \pi_{t+1} \left[ \pi_{t+1} - \pi_t^{\psi} \pi^{1-\psi} \right] \\ R_t^* - (R_{t-1}^*)^{\rho_r} \left( R \left(\frac{\pi_t}{\pi}\right)^{\kappa_{\pi}} \left(\frac{Y_t}{Y}\right)^{\kappa_Y} \right)^{1-\rho_r} \exp(\sigma_R \varepsilon_{R,t}) \\ \log A_t - \rho_A \log A_{t-1} - \sigma_A \varepsilon_{A,t} \\ \log Z_t - \rho_Z \log Z_{t-1} - \sigma_Z \varepsilon_{Z,t} \\ R_t - R_t^* \exp(\tilde{\varepsilon}_{R,t}) \\ X_t \tilde{\varepsilon}_{R,t} + (1 - X_t)(R_t - R_{ELB}) \end{bmatrix} = 0. \quad (55)$$



As in the previous model with the lower bound constraint,  $X_t$  is treated as exogenous until after the symbolic derivatives have been calculated, when it is replaced with  $R_t^* > R_{ELB}$ . This is because  $X_t$  is an indicator function that is not differentiable, and the derivatives of the indicator function with respect to the model variables are not required to solve the model when using Newton's method. As a consequence symbolic derivatives can be used to solve the model with occasionally binding constraints, improving the overall performance of the procedure. Once the derivatives have been taken, all instances of  $X_t$  in the vector of model equations and the Jacobian matrix are replaced with  $R_t^* > R_{ELB}$ , which takes the value 1 when true in Matlab and 0 otherwise.

### 5.4.3 Comparing the Different Models with Occasionally Binding Constraints

In this section I compare the results from the the model where the lower bound constraint is imposed using anticipated monetary policy shocks (the APS model) and the model where the lower bound is introduced as a separate regime (the LBR model). I show the equivalence of the methods algebraically and through a scenario, with both methods delivering the same result.

It is relatively straight forward to show algebraically that the LBR model is equivalent to the APS model. I start by listing the complementary slackness conditions from both models and the mapping from the shadow rate to the nominal interest rate in the APS model.

The complementary slackness condition from the LBR model is given by,

$$R_t - R_{ELB} - X_t (R_t^* - R_{ELB}) = 0. \quad (56)$$

The complementary slackness condition from the APS model is given by,

$$X_t \tilde{\varepsilon}_{R,t} + (1 - X_t) (R_t - R_{ELB}) = 0, \quad (57)$$

where the shadow rate is mapped to the actual nominal interest rate according to,

$$R_t - R_t^* \exp(\tilde{\varepsilon}_{R,t}) = 0. \quad (58)$$

When  $R_t^* \leq R_{ELB}$  then  $X_t = 0$  according to equation (53). In the model with a lower bound regime, equation (56) becomes,

$$R_t = R_{ELB}, \quad (59)$$

and in the model with anticipated monetary policy shocks, equation (57) becomes,

$$R_t = R_{ELB}, \quad (60)$$

so that both models give the same result for the same value of the indicator function, which is common to both models. This implies,

$$\tilde{\varepsilon}_{R,t} = \log\left(\frac{R_{ELB}}{R_t^*}\right), \quad (61)$$

in the model where the lower bound is imposed using anticipated monetary policy shocks. When  $R_t^* \geq R_{ELB}$  then  $X_t = 1$  according to equation (53). In the model with a lower bound regime, equation (56) becomes,

$$R_t = R_t^*, \quad (62)$$

and in the second model with anticipated shocks, equation (57) becomes,

$$\tilde{\varepsilon}_{R,t} = 0, \quad (63)$$



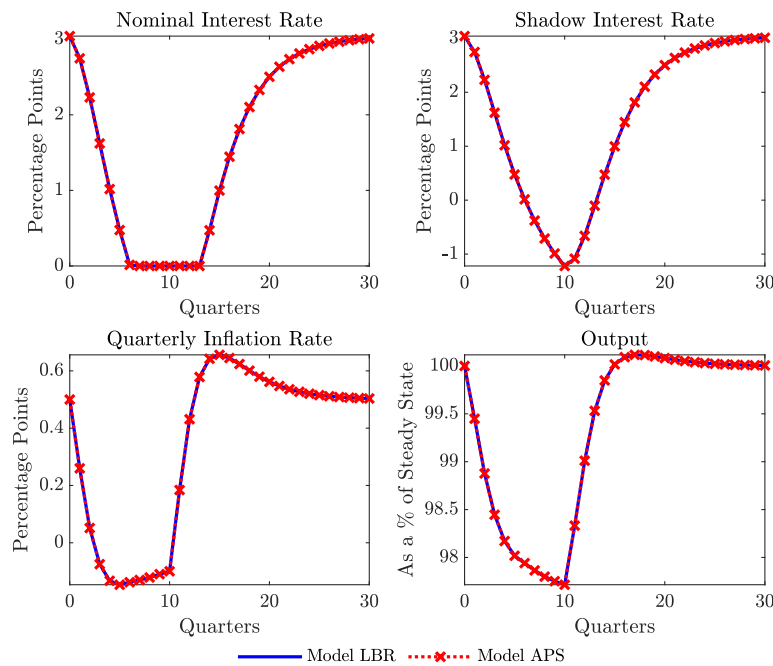
which implies that equation (58) becomes,

$$R_t = R_t^*, \quad (64)$$

so that both models give the same result when the constraint does and does not bind, making them equivalent.

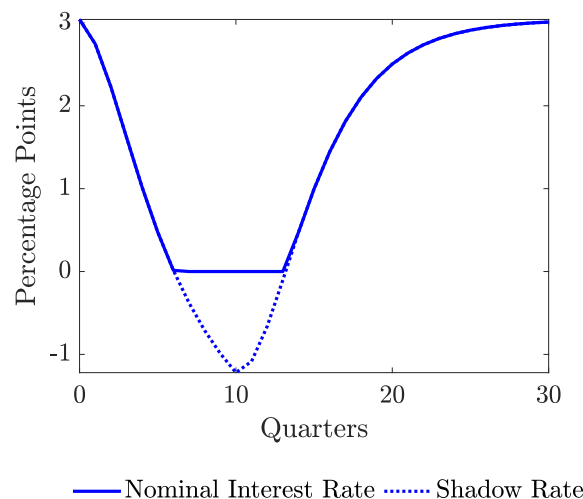
Furthermore, I demonstrate that both models give the same result through a scenario. Using the same sequence of shocks I perturb both model economies for 10 consecutive quarters with negative preference shocks. These are large enough to take the model economy to the lower bound. I plot the results in Figure (6) below.

**Figure 6 – The Effective Lower Bound: Comparing Methods**



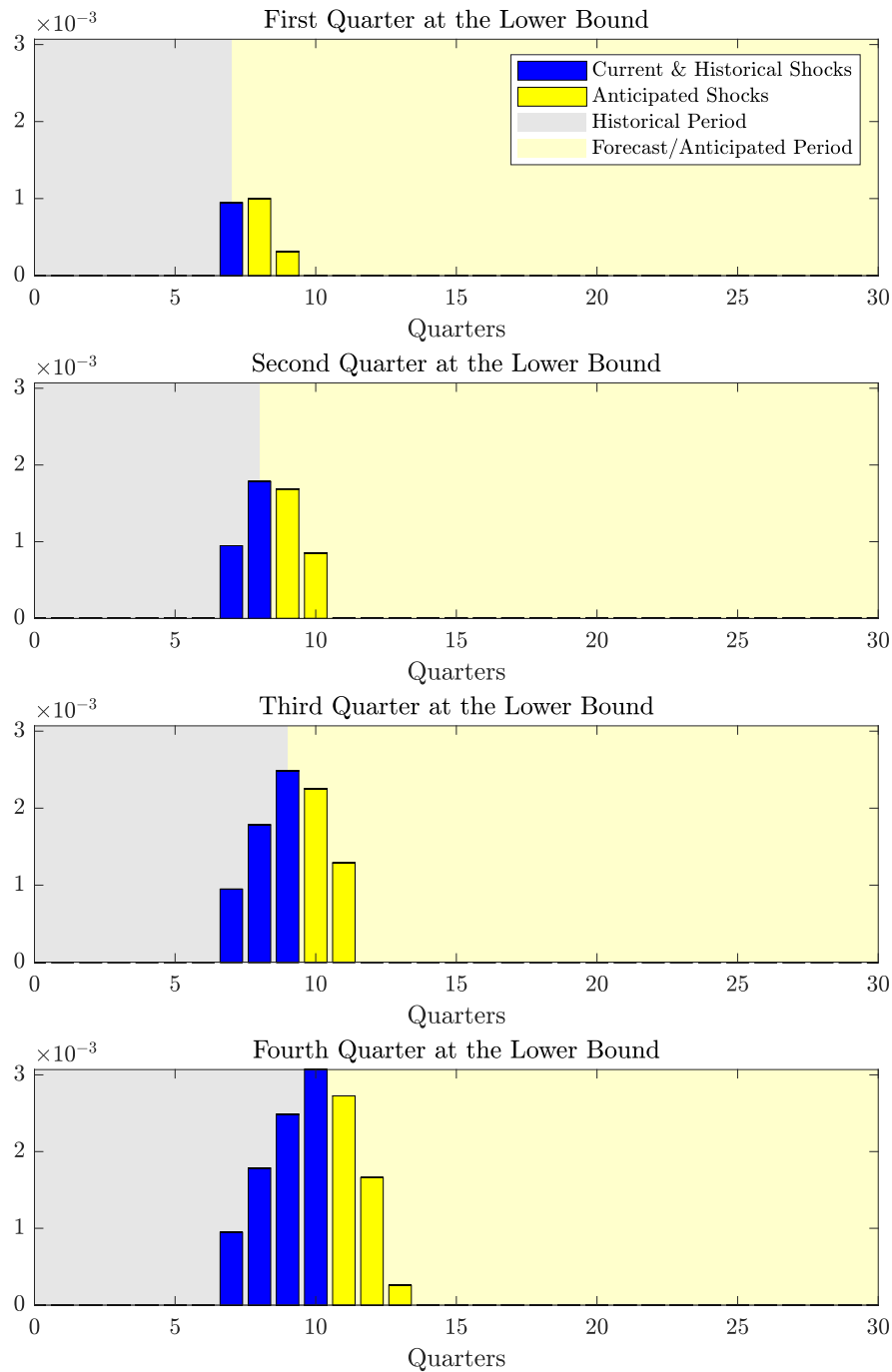
As expected, both methods give the same result. I plot the shadow rate against the nominal interest rate in Figure (7).

**Figure 7 – The Nominal Interest Rate vs The Shadow Interest Rate**



This figure, together with the top two panels of Figure 6, illustrates that the lower bound constraint in both models only binds while the shadow rate is negative and that the switch back to monetary policy determined by a Taylor-type rule occurs when the shadow rate becomes positive. I also show how the anticipated monetary policy shocks change through the scenario as each unanticipated preference shock is revealed to agents, according to the standard rational expectations assumption. Each unanticipated preference shock that hits the economy over the lower bound period changes the shadow rate, altering the expected duration of the lower bound so that agents update their expectations of the monetary policy shocks required to enforce the lower bound constraint. This is illustrated in Figure 8 below.

**Figure 8 – The Effective Lower Bound: Comparing Methods**



The first panel of Figure 8 shows the actual and anticipated monetary policy shocks required to prevent interest rates going below the effective lower bound. The economy hits the lower bound in the seventh simulation period, where the blue bar represents the size of the actual monetary policy shock in that period and the yellow bars represent anticipated shocks. Because there is persistence in both the model economy and the preference shock process, the economy does not immediately revert to a state where the shadow rate is greater than the effective lower bound. In fact the shadow rate is expected to remain below the lower bound for a further two periods. As only one negative preference shock has hit the economy in the first period that the lower bound binds, and no further preference shocks are expected to hit the economy at this point, the actual and expected monetary policy shocks required to prevent interest rates going below the lower bound are

quite small. Each successive panel reveals an additional negative preference shock hitting the economy, extending the period agents believe they will be at the lower bound and deepening the recession. With each additional shock and the worsening recession, larger and larger monetary policy shocks (actual and expected) are required to both stop current and expected interest rates from violating the lower bound constraint. Using monetary policy shocks is essentially a modelling device to impose the effective lower bound. Figure 8 makes it clear that expectations and the corresponding expected monetary policy shocks are continually being updated as unexpected shocks hit the economy.

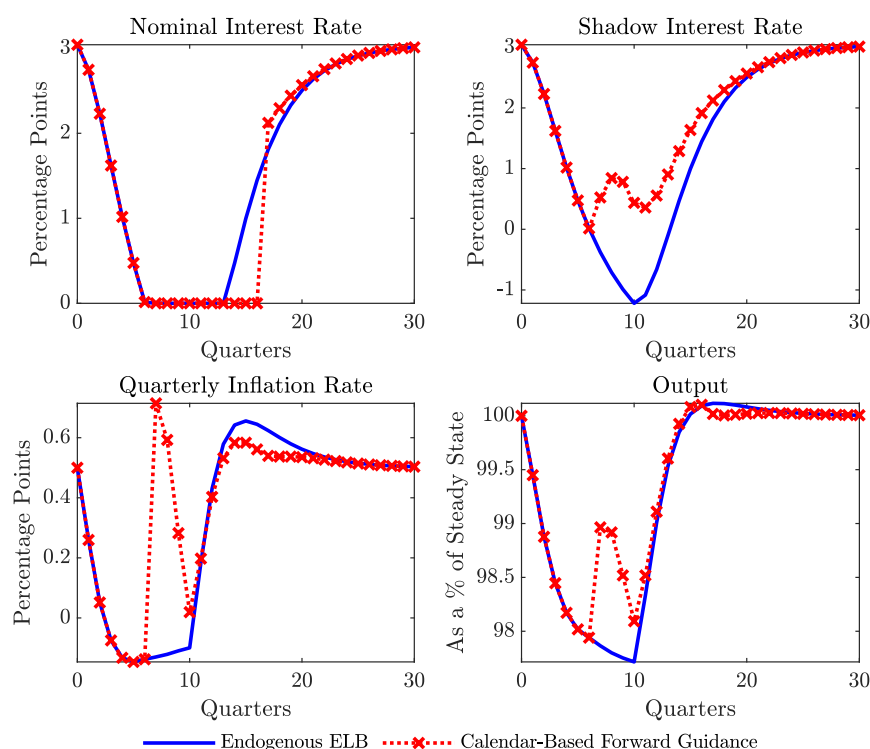
#### 5.4.4 Comparing the Endogenously Binding Lower Bound Constraint with Calendar-Based Odyssean Forward Guidance

The previous section illustrated the equivalence between two different methods for endogenously imposing the lower bound constraint on interest rates. In both cases the lower bound was only binding for as long as the shadow rate – determined by a Taylor-type rule and hence a function of economic conditions – was below the lower bound. However, forward guidance has become a new weapon in central banks' inflation fighting arsenal. Campbell et al. (2012) make the distinction between Delphic forward guidance, which is the announcement and publication of conditional or endogenous interest rate tracks dependent on economic conditions, and Odyssean forward guidance, where central banks publicly commit to an interest rate track that deviates from their usual reaction function. Jones et al. (2020) focus on what they call calendar-based Odyssean forward guidance, where the central bank commits to holding interest rates at the lower bound for a fixed period of time without explicit regard to current and future economic conditions. In a modelling context this is equivalent to imposing the lower bound on interest rates using either exogenous anticipated monetary policy shocks or the lower bound regime for a fixed period of time, so that the expected exit from the lower bound is not a function of the shadow interest rate or economic conditions. Calendar-based Odyssean forward guidance will have different economic implications as the central bank commits to keeping interest rates at the lower bound for a set period of time. When at the lower bound, agents' expectations about the duration of the lower bound binding will not be updated as economic shocks alter economic conditions.

I compare the results from calendar-based Odyssean forward guidance with an endogenously binding lower bound constraint through a scenario. I impose the lower bound in both models as a lower bound regime, although the same results could be achieved using anticipated monetary policy shocks. I repeat the same exercise in Section 5.4.3, where for 10 consecutive quarters the economy is hit by a sequence of negative preference shocks that take the economy to the lower bound in the seventh simulation period.<sup>22</sup> This time I assume that when the economy hits the lower bound, the central bank announces that they will keep interest rates at the lower bound for two and a half years, which is believed by agents. This commitment to keep interest rates at the lower bound is independent of economic conditions or subsequent shocks, which represents calendar-based Odyssean forward guidance. This is achieved by setting  $X_t = 1$  for 10 quarters, once the shadow rate goes below the lower bound, so that agents know once they are at the lower bound they will be there for at least 10 quarters. This scenario is plotted against the results from the model with an endogenously binding lower bound constraint in Figure 9 below.

<sup>22</sup> In both scenarios I assume that agents' expectations about future interest rates are subject to the endogenous lower bound constraint before the lower bound binds (that is expected entry and exit into the lower bound are determined by expected economic conditions, which determine when the lower bound is expected to bind). For the simulation using calendar-based Odyssean forward guidance, I take the first period the lower bound actually binds (the seventh simulation period) as initial conditions for the simulation. I then splice the first 7 periods of the simulation with the endogenous lower bound (where the constraint does not bind for the first 6 periods) to the second simulation where the lower bound is exogenously imposed to create the scenario with calendar-based Odyssean forward guidance.

**Figure 9 – The Effective Lower Bound: Extended Path vs Perfect Foresight**



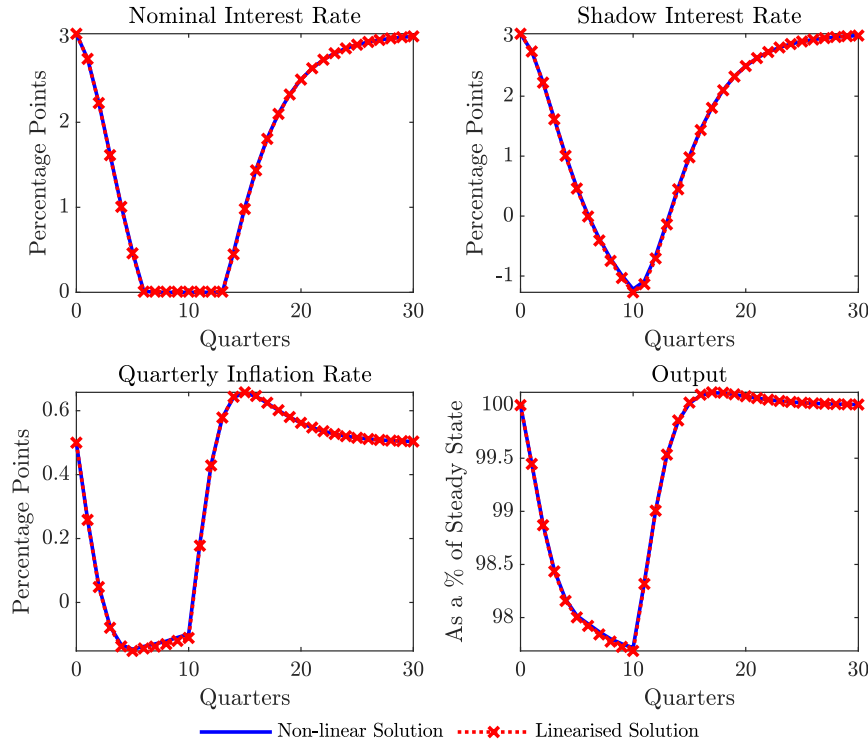
The announcement and commitment to keeping interest rates at the lower bound for 10 quarters as soon as the lower bound binds is quite powerful in this simple model.<sup>23</sup> At the onset of the recession agents are unaware how long the recession will last. Committing to keep rates at the lower bound for two and a half years generates additional inflation, raising inflation expectations, lowering the real interest rate, which provides some support to output, relative to the endogenous case, where the expected duration of the lower bound is gradually extended or updated as the extent of the recession is revealed. This illustrates, at least in a rational expectations modelling context, the power of forward guidance. However, it leaves unanswered the extent of the commitment required to optimise the return to steady state.

#### 5.4.5 Comparing the Non-linear Model with the Linearised Model

In the previous sections the occasionally binding constraint was imposed on the non-linear model, leading to a fully non-linear solution. As highlighted by Aruoba et al. (2006), many dynamic non-linear models (in the absence of occasionally binding constraints) display behaviour that is close to a linear law of motion for plausible parameterisations and shocks. In many cases it is the introduction and imposition of occasionally binding constraints that are the main source of non-linearity and asymmetry in otherwise standard dynamic non-linear models. When combined with larger models that take longer to solve and simulate, it may prove satisfactory to apply the occasionally binding constraint to the linearised version of the model. I develop a version of my algorithm that first linearises the non-linear model automatically, avoiding mistakes from pencil and paper annotations, before applying the constraint. Appendix C provides the updated equations for this version of the algorithm. I repeat the same scenario from Section 5.4.3, this time comparing the output from the linearised model with the constraint applied to the full non-linear solution. The results are plotted in Figure 10.

<sup>23</sup> This may be due to the forward guidance puzzle (see Giannoni et al. 2015, for example).

**Figure 10 – The Effective Lower Bound: Non-linear vs Linearised**



As was established in Section 5.3, this parameterisation of the simple New Keynesian model results in behaviour that is close to a linear law of motion. Figure 10 shows that the results from the linearised solution with the lower bound constraint imposed are almost identical to the results from the full non-linear solution. Given the reduced computational burden, it may be beneficial to use the linearised model with the constraint imposed, over the full non-linear solution, when the dynamic non-linear model is close to linear in behaviour.

I run some simulation experiments to illustrate the potential speed gains of solving and simulating the linearised model with occasionally binding constraints in place of the full non-linear solution. The linearised model with occasionally binding constraints is 72% faster to solve and simulate than the same model with the same shocks solved and simulated using the full non-linear model with the vectorised code. The full details of the simulation experiments and results are reported in Appendix B.

## 5.5 Optimal Monetary Policy Under Commitment with a Binding Lower Bound Constraint

In this section, I show how optimal monetary policy under commitment can be implemented in the extended path algorithm, with a binding lower bound on nominal interest rates. For the sake of simplicity, I assume that the social planner sets monetary policy to minimise the following ad hoc loss function,

$$L_0 = E_0 \left\{ \sum_{t=0}^{\infty} \beta^t \left[ (\pi_t - \pi)^2 + \omega_Y \left( \frac{Y_t - Y}{Y} \right)^2 + \omega_{\Delta R} \Delta R_t^2 \right] \right\}, \quad (65)$$

where  $L_0$  is the loss function in period 0,  $\Delta R_t = R_t - R_{t-1}$  is the change in the nominal interest rate and  $\omega_Y$  and  $\omega_{\Delta R}$  are the relative weights attached to the deviations of output

from steady state and changes in the nominal interest.<sup>24</sup> The social planner minimises the ad hoc loss function by choosing allocations of the model variables,  $Y_t$ ,  $\pi_t$ ,  $R_t$ ,  $\Delta R_t$  and  $R_t^*$ , subject to the model's equilibrium conditions and the lower bound constraint on interest rates. I continue to use  $R_t^*$  to represent the shadow interest rate. This can be written more formally as,

$$\min_{Y_t, \pi_t, R_t, \Delta R_t, R_t^*} E_0 \left\{ \sum_{t=0}^{\infty} \beta^t \left[ (\pi_t - \pi)^2 + \omega_Y \left( \frac{Y_t - Y}{Y} \right)^2 + \omega_{\Delta R} (R_t - R_{t-1})^2 \right] \right\}, \quad (66)$$

$$\text{subject to} \quad (67)$$

$$Z_t (Y_t - \chi Y_{t-1})^{-\sigma} - E_t \left\{ \beta \frac{R_t}{\pi_{t+1}} Z_{t+1} (Y_{t+1} - \chi Y_t)^{-\sigma} \right\} = 0, \quad (68)$$

$$\left( \frac{\phi}{v-1} \right) \pi_t \left[ \pi_t - \pi_{t-1}^\psi \pi^{1-\psi} \right] - \left( \frac{v}{v-1} \right) \kappa (Y_t - \chi Y_{t-1})^\sigma Y_t^\eta A_t^{-(1+\eta)} + 1 - \dots \quad (69)$$

$$\dots - E_t \left\{ \left( \frac{\phi}{v-1} \right) \beta \left( \frac{Z_{t+1}}{Z_t} \right) \left( \frac{Y_{t+1} - \chi Y_t}{Y_t - \chi Y_{t-1}} \right)^{-\sigma} \left( \frac{Y_{t+1}}{Y_t} \right) \pi_{t+1} \left[ \pi_{t+1} - \pi_t^\psi \pi^{1-\psi} \right] \right\} = 0, \quad (70)$$

$$\log A_t - \rho_A \log A_{t-1} - \sigma_A \varepsilon_{A,t} = 0, \quad (71)$$

$$\log Z_t - \rho_Z \log Z_{t-1} - \sigma_Z \varepsilon_{Z,t} = 0, \quad (72)$$

$$\Delta R_t - (R_t - R_{t-1}) = 0, \quad (73)$$

$$R_t \geq R_{ELB}. \quad (74)$$

Note the absence of the Taylor-type interest rate rule from the model's equilibrium conditions and the social planner's constraints. This is because interest rates are now being set to optimise the welfare function. This can be rewritten more generally as,

$$\min_x [x - \bar{x}]^\top H [x - \bar{x}], \quad (75)$$

$$\text{subject to } F(x) = 0,$$

where  $\bar{x}$  is the  $n.T \times 1$  vector of steady states for the vector  $n.T \times 1$  vector of endogenous variables  $x$ . This more general form allows the model to be solved using the extended path algorithm. This setup is very similar to the conditional forecasting exercise with underidentified shocks presented in Section 5.2. To improve readability I use  $x^h$  to represent  $x_{1:T}^h$ .<sup>25</sup> Here  $F(x^h)$  contains the model equations for the  $T$  simulation periods,

$$F(x^h) = \begin{bmatrix} f(E_t \{x_2^h\}, x_1^h, x_0, \varepsilon_1) \\ \vdots \\ f(E_t \{x_{t+1}^h\}, x_t^h, x_{t-1}^h, \varepsilon_t) \\ \vdots \\ f(E_t \{x_{T+1}^h\}, x_T^h, x_{T-1}^h, \varepsilon_T) \end{bmatrix}, \quad (76)$$

<sup>24</sup> The algorithm I describe should work equally well with a micro-founded loss function.

<sup>25</sup> I continue to use notation that is consistent with the deterministic extended path algorithm, even though my example uses the stochastic extended path algorithm.

where  $f(E_t \{x_{t+1}^h\}, x_t^h, x_{t-1}^h, \varepsilon_t)$  is the model equations in simulation period  $t$ , such that,

$$f(E_t \{x_{t+1}\}, x_t, x_{t-1}, \varepsilon_t) = \begin{bmatrix} Z_t (Y_t - \chi Y_{t-1})^{-\sigma} - \beta \frac{R_t}{\pi_{t+1}} Z_{t+1} (Y_{t+1} - \chi Y_t)^{-\sigma} \\ \left(\frac{\phi}{v-1}\right) \pi_t \left[\pi_t - \pi_{t-1}^\psi \pi^{1-\psi}\right] - \left(\frac{v}{v-1}\right) \kappa (Y_t - \chi Y_{t-1})^\sigma Y_t^\eta A_t^{-(1+\eta)} + 1 - \dots \\ \dots - \left(\frac{\phi}{v-1}\right) \beta \left(\frac{Z_{t+1}}{Z_t}\right) \left(\frac{Y_{t+1} - \chi Y_t}{Y_t - \chi Y_{t-1}}\right)^{-\sigma} \left(\frac{Y_{t+1}}{Y_t}\right) \pi_{t+1} \left[\pi_{t+1} - \pi_t^\psi \pi^{1-\psi}\right] \\ \log A_t - \rho_A \log A_{t-1} - \sigma_A \varepsilon_{A,t} \\ \log Z_t - \rho_Z \log Z_{t-1} - \sigma_Z \varepsilon_{Z,t} \\ \Delta R_t - R_t + R_{t-1} \\ R_t - R_{ELB} - X_t (R_t^* - R_{ELB}) \end{bmatrix} = 0, \quad (77)$$

where I substitute  $R_t - R_{ELB} - X_t (R_t^* - R_{ELB})$  for  $R_t \geq R_{ELB}$ . As before,  $x_t$  is the vector of date  $t$  endogenous variables, which now look like,

$$x_t = [Y_t \quad \pi_t \quad R_t \quad A_t \quad Z_t \quad \Delta R_t \quad R_t^*], \quad (78)$$

$\varepsilon_t$  is the vector of date  $t$  exogenous variables,

$$\varepsilon_t = [\varepsilon_{Z,t} \quad \varepsilon_{A,t}], \quad (79)$$

and  $X_t$  is an indicator variable that enforces the lower bound constraint on interest rates,

$$X_t = \begin{cases} 0 & \text{if } R_t^* \leq R_{ELB}, \\ 1 & \text{if } R_t^* > R_{ELB}. \end{cases} \quad (80)$$

As with the endogenous lower bound examples from the previous sections,  $X_t$  is only replaced in the equations after the symbolic derivatives in the Jacobian has been calculated. The  $H$  matrix in the objective function can be expressed as,

$$H = \begin{bmatrix} \mathcal{H} & 0 & \dots & \dots & 0 \\ 0 & \ddots & 0 & & \vdots \\ \vdots & 0 & \beta^{t-1} \mathcal{H} & 0 & \vdots \\ \vdots & & 0 & \ddots & 0 \\ 0 & \dots & \dots & 0 & \beta^{T-1} \mathcal{H} \end{bmatrix}, \quad (81)$$

where,

$$\mathcal{H} = \begin{bmatrix} \frac{\omega_Y}{Y^2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \omega_{\Delta R} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (82)$$

Note the denominator of  $\frac{\omega_Y}{Y^2}$  is the steady state level of output squared to account for the squared output gap in the loss function. Taking a second order approximation of the objective function gives,

$$[x^{h+1} - \bar{x}]^\top H [x^{h+1} - \bar{x}] = [x^h + \Delta x^{h+1} - \bar{x}]^\top H [x^h + \Delta x^{h+1} - \bar{x}] \quad (83)$$

$$\approx [x^h - \bar{x}]^\top H [x^h - \bar{x}] + \dots \quad (84)$$

$$\dots + 2(\Delta x^{h+1})^\top H [x^h - \bar{x}] + (\Delta x^h)^\top H \Delta x^{h+1} \quad (85)$$



Taking a first order approximation of the non-linear equality constraints gives,

$$F(x^{h+1}) = F(x^h + \Delta x^{h+1}) \approx F(x^h) + J(x^h) \Delta x^{h+1}, \quad (86)$$

where the Jacobian matrix,  $J(x^h)$ , is  $n_{eq} \times n$ , and  $n_{eq} < n$  is the number of model equations, which is less than  $n$  the number of model variables. The Jacobian matrix is no longer a square matrix under optimal policy, as there are fewer model equations than model variables, given that the monetary policy reaction no longer features in the constraints. Replacing the objective function with (85) and the equality constraints with (86) gives,

$$\min_{\Delta x^{h+1}} \quad [x^h - \bar{x}]^\top H [x^h - \bar{x}] + 2(\Delta x^{h+1})^\top H [x^h - \bar{x}] + (\Delta x^h)^\top H \Delta x^{h+1} \quad (87)$$

$$\text{subject to} \quad F(x^h + \Delta x^{h+1}) \approx F(x^h) + J(x^h) \Delta x^{h+1} \quad (88)$$

This leads to the following first order condition,<sup>26</sup>

$$2H [x^h - \bar{x}] + 2H \Delta x^{h+1} + J(x^h)^\top \lambda = 0, \quad (89)$$

where  $\lambda$  is the  $n_{eq} \cdot T \times 1$  vector of Lagrange multipliers on the equality constraints. Combining the first order condition of the minimisation problem with the linearised equality constraint results in,

$$\begin{bmatrix} 2H & J(x^h)^\top \\ J(x^h) & 0 \end{bmatrix}_{n_{eq} \cdot T \times n_{eq} \cdot T} \begin{bmatrix} \Delta x^{h+1} \\ \lambda \end{bmatrix} = \begin{bmatrix} -2H(x^h - \bar{x}) \\ -F(x^h) \end{bmatrix}. \quad (90)$$

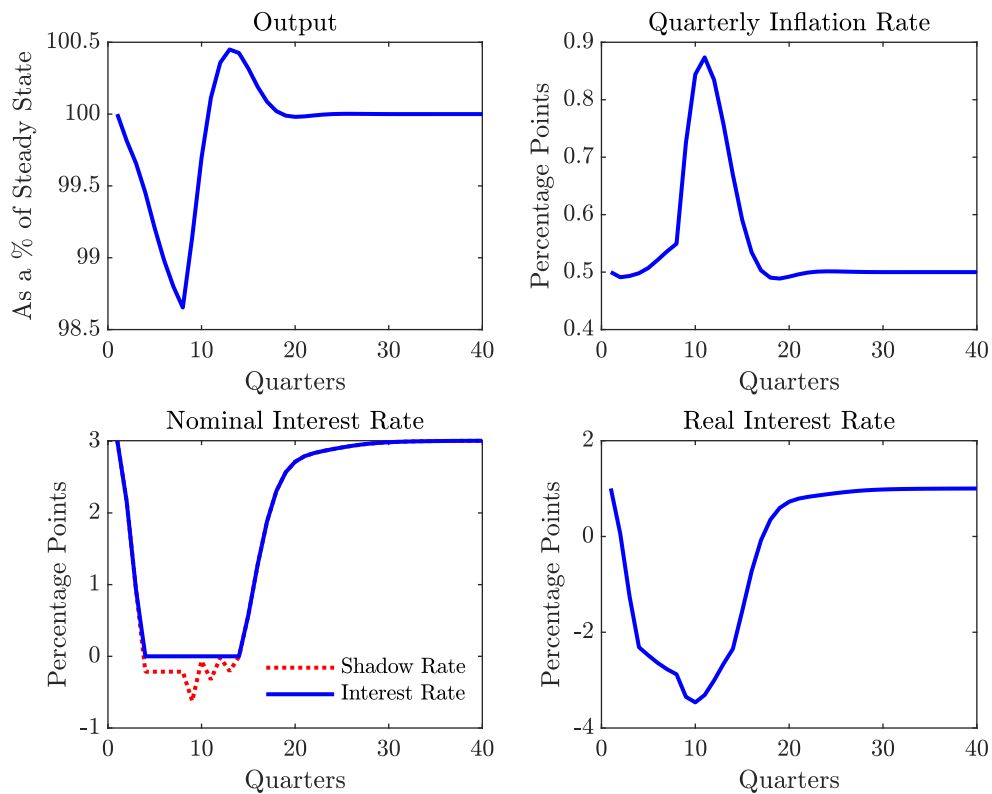
Since  $\Delta x^{h+1} = x^{h+1} - x^h$ , this can be rewritten in the form of Newton's method as follows,

$$\begin{bmatrix} x^{h+1} \\ \lambda \end{bmatrix} = \begin{bmatrix} x^h \\ 0 \end{bmatrix}_{n_{eq} \cdot T \times 1} - \begin{bmatrix} 2H & J(x^h)^\top \\ J(x^h) & 0 \end{bmatrix}_{n_{eq} \cdot T \times n_{eq} \cdot T}^{-1} \begin{bmatrix} 2H(x^h - \bar{x}) \\ F(x^h) \end{bmatrix}. \quad (91)$$

I demonstrate the algorithm with a simple example. I set  $\omega_Y$  and  $\omega_{\Delta R}$  both to 0.5. I perturb the economy with a sequence of negative unexpected preference shocks for seven quarters, until interest rates reach the effective lower bound. The (shadow) interest rate is set optimally to try and minimise the ad hoc loss function in (65). I solve the model using the stochastic extended path algorithm, so that shocks are unexpected, consistent with the usual rational expectations assumptions. The results from the simulation are presented in Figure 11 below.

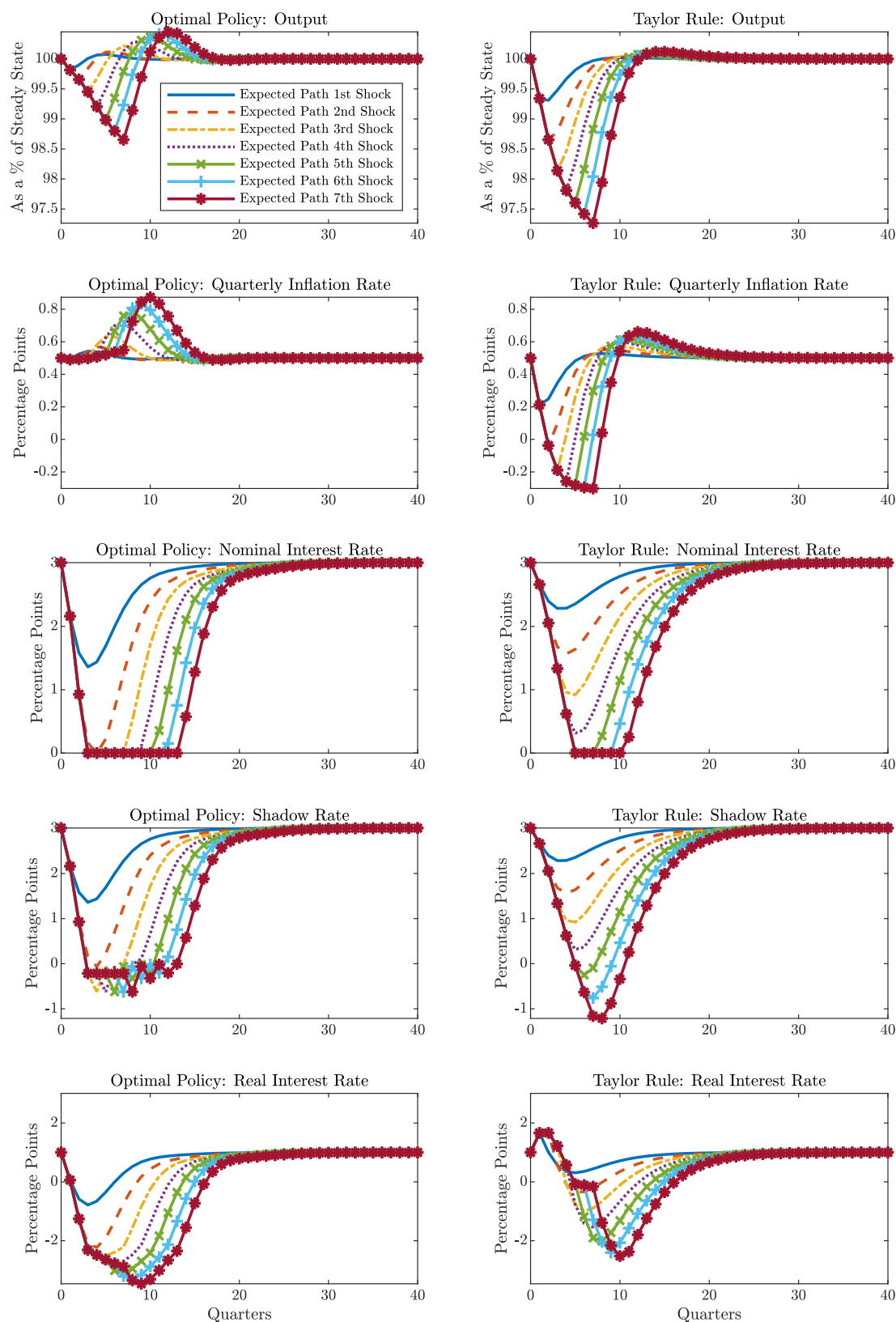
<sup>26</sup> This is almost identical to equation (33).

**Figure 11 – Optimal Policy: Model Variables**



The sequence of negative preference shocks introduced in this scenario lowers output by more than 1.25% of steady state GDP. Interest rates are cut drastically, until they reach the lower bound. Inflation increases due to the relatively large and fast fall in the nominal interest rate. This is because optimal monetary policy requires the central bank to cut rates by an amount that is sufficiently large to raise inflation expectations. Lower nominal interest rates and higher inflation expectations lead to a large fall in the real interest rate which provides support to inflation and output. To better understand how this happens period by period as this scenario unfolds, I plot the simulations after each additional preference shock is added for the key model variables in Figure 12. For comparison I plot the model responses after each additional shock is added when policy follows a simple Taylor-type rule with a binding lower bound constraint, using the parameterisation described in Section 2.

**Figure 12 – Optimal Policy: Comparison with Simple Rule**



Note: The label “Expected Path n-th Shock” represents the expected paths of the variables after the shocks have hit in the n-th simulation period. Expectations are updated as more shocks hit the economy over the simulation period.

From Figure 12 it is clear to see that the monetary authority cuts interest rates quite drastically with each successive shock when policy is set optimally, and that interest rates are expected to fall further in the future, in response to the negative preference shocks when they first start hitting the economy. When policy is set according to a Taylor-type rule, the interest rate response is more sluggish initially, as the nominal interest rate falls by less than the fall in inflation expectations leading to an increase in the real interest rate, which is a movement in the wrong direction. The early more aggressive response when policy is set optimally leads to a smaller overall fall in output and an increase inflation and smaller cuts in the policy rate as further shocks hit the economy. In fact the larger initial response when policy is set optimally leads to an increase in expected inflation and an overshoot in actual inflation. This increase in inflation and inflation expectations causes the real interest rate to fall by more, further stimulating the economy. This scenario illustrates that an initial aggressive policy response as prescribed by optimal policy can prevent larger policy responses later, or as the old adage goes, a stitch in time saves nine.

## 5.6 Transition Paths

I demonstrate how the code can be used to produce a deterministic transition path, representing a transition from an initial steady state to a terminal steady state. Although I don't demonstrate it with this example, it is also possible to produced stochastic transition paths by adding shocks in addition to changes in parameters. In the context of the simple New Keynesian DSGE model, I show how to model a permanent lowering of the inflation target.<sup>27</sup> This requires introducing a time varying inflation target into the model. I assume that it follows an autoregressive process of the form,

$$\log \pi_t^* = \rho_\pi \log \pi_{t-1}^* + (1 - \rho_\pi) \log \pi. \quad (92)$$

I also replace all instances of the steady state inflation rate with the time varying inflation target so that the augmented system of equations becomes,

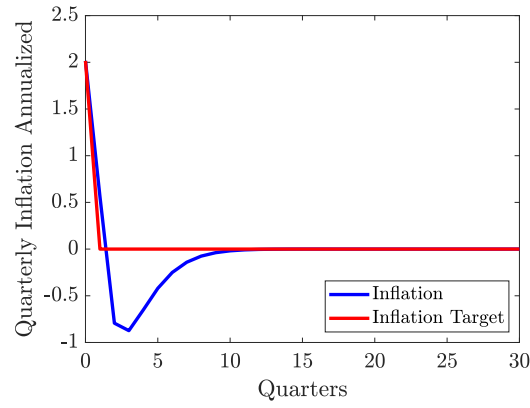
$$f(E_t \{x_{t+1}\}, x_t, x_{t-1}, \varepsilon_t) = \begin{bmatrix} Z_t (Y_t - \chi Y_{t-1})^{-\sigma} - \beta \frac{R_t}{\pi_{t+1}} Z_{t+1} (Y_{t+1} - \chi Y_t)^{-\sigma} \\ \left(\frac{\phi}{v-1}\right) \pi_t \left[ \pi_t - \pi_{t-1}^\psi (\pi_{t-1}^*)^{1-\psi} \right] - \left(\frac{v}{v-1}\right) \kappa (Y_t - \chi Y_{t-1})^\sigma Y_t^\eta A_t^{-(1+\eta)} + 1 - \dots \\ \dots - \left(\frac{\phi}{v-1}\right) \beta \left(\frac{Z_{t+1}}{Z_t}\right) \left(\frac{Y_{t+1} - \chi Y_t}{Y_t - \chi Y_{t-1}}\right)^{-\sigma} \left(\frac{Y_{t+1}}{Y_t}\right) \pi_{t+1} \left[ \pi_{t+1} - \pi_t^\psi (\pi_t^*)^{1-\psi} \right] \\ R_t - R_{t-1}^{\rho_r} \left( R \left( \frac{\pi_t}{\pi_t^*} \right)^{\kappa_\pi} \left( \frac{Y_t}{Y} \right)^{\kappa_Y} \right)^{1-\rho_r} \exp(\sigma_R \varepsilon_{R,t}) \\ \log A_t - \rho_A \log A_{t-1} - \sigma_A \varepsilon_{A,t} \\ \log Z_t - \rho_Z \log Z_{t-1} - \sigma_Z \varepsilon_{Z,t} \\ \log \pi_t^* - \rho_\pi \log \pi_{t-1}^* - (1 - \rho_\pi) \log \pi \end{bmatrix} = 0. \quad (93)$$

To demonstrate, I run a simulation where the central bank cuts the inflation target from 2% to 0%. In the baseline simulation I assume that  $\rho_\pi = 0$ . The initial condition in the simulation,  $x_0$ , is the model steady state solved with a 2% inflation target, and the terminal condition,  $x_{T+1}$ , is the model steady state solved with 0% inflation target.

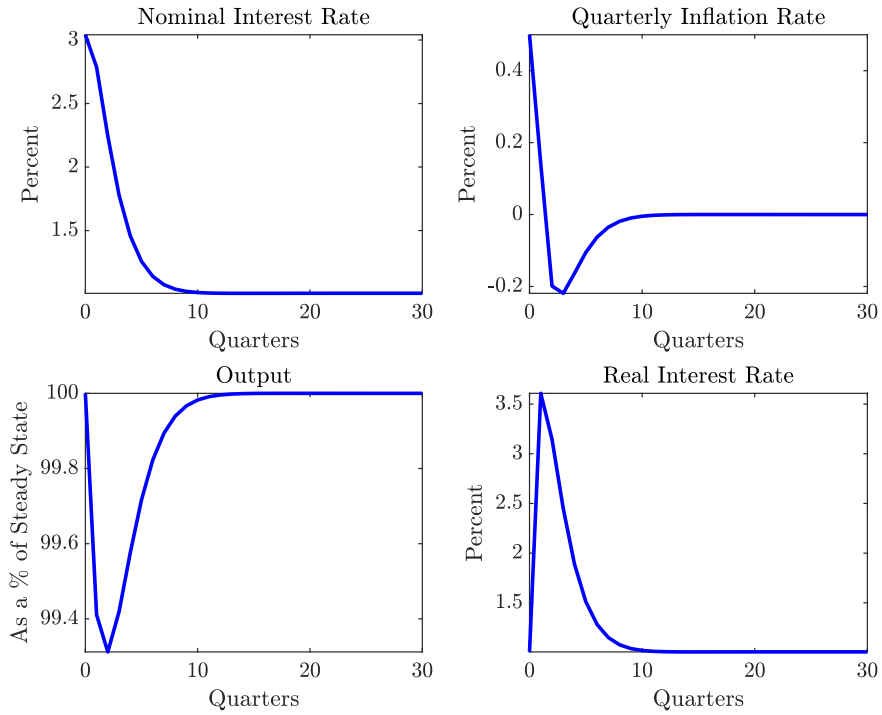
I plot inflation against the inflation target in Figure 13 and a subset of key variables in Figure 14.

<sup>27</sup> The same general methodology and Matlab code was used to produce the projections and supporting analysis for the 2021 Long-Term Fiscal Statement with the stochastic neoclassical growth model. In that case the transition represented permanent demographic changes and policy developments. See Binning (2021) for details.

**Figure 13 – A Transition Path: Inflation**



**Figure 14 – A Transition Path: Model Variables**



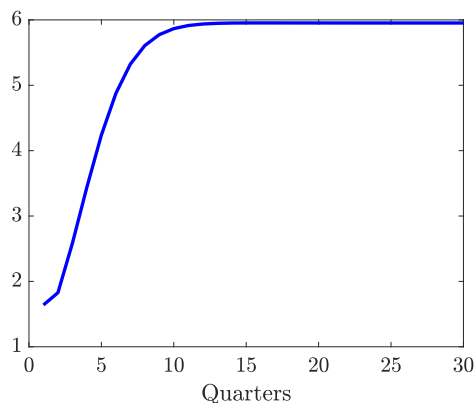
The central bank announces the change in the inflation target and it is credibly believed by agents in the model. The real interest rate needs to increase to lower aggregate demand and reduce inflation. This is initially achieved by reducing the nominal interest rate at a slower rate than inflation falls. The increase in the real interest rate and the reduced aggregate demand cause real GDP to fall, making the disinflation costly.<sup>28</sup> To determine how costly the transition to a lower inflation target is, I calculate the trajectory of the sacrifice ratio as follows,

$$SR_t = -\frac{\sum_{k=0}^t (Y_k - Y_0)/Y_0}{\pi_0 - \pi_t}, \quad (94)$$

<sup>28</sup> Disinflation periods are often costly in terms of lost GDP as real interest rates are held high until inflation has reached its new target. Former Fed chairman Paul Volcker's efforts to reduce inflation in the early 1980s are widely believed to have contributed to the recession of 1981-82 in the US (see Goodfriend & King 2005, for example).

where  $SR_t$  is the date  $t$  sacrifice ratio. The numerator represents the cumulated output losses up to date  $t$ , as a percentage of the steady state and the denominator represents the change in inflation up to date  $t$ . The sacrifice ratio is plotted against time in Figure (15).

**Figure 15 – The Sacrifice Ratio**



The cumulative sacrifice ratio grows with time, peaking at just under 6 after 10 quarters. This is in the neighbourhood of estimates by Wascher & Andersen (1999) for the US in the 1985-1998 period.

## 6. Conclusion

Recent experience with interest rates hitting the effective lower bound and households facing binding borrowing constraints has emphasised the importance of non-linear solution methods and the ability to model occasionally binding constraints. The extended path algorithm is a flexible, reliable and commonly used method for solving large non-linear rational expectations models. In this paper I show how to code an efficient representation of the extended path algorithm in Matlab, making use of vectorisation, sparsity and symbolic derivatives, reducing the time taken to calculate the Jacobian and fill in the equation residuals. I apply my algorithm to a number of policy relevant problems and illustrate some of the speed gains. I show how the algorithm can be used to handle conditional forecasting in non-linear rational expectations models with both exactly identified shocks and overidentified shocks, which is particularly useful for forecasters. I show how occasionally binding constraints can be implemented using symbolic derivatives, which has speed advantages over slower quasi-Newton methods. I show how calendar-based Odyssean forward guidance can be implemented using the algorithm. I present a version of the algorithm where the constraints are applied to the linearised model. This has particular advantages for larger models, which may take longer to solve using the full non-linear solution, where the main non-linearity may be the occasionally binding constraint. I also show how the algorithm can handle optimal monetary policy under commitment with a binding lower bound constraint on nominal interest rates. This provides an important counterfactual for policy makers. Finally I show how the algorithm can be used to construct deterministic transition paths which can be used to show policy makers how an economy might adjust following a permanent change in policy.

## References

- Adjemian, S. & Juillard, M. (2010). Dealing with ZLB in DSGE Models an Application to the Japanese Economy. ESRI Discussion paper series 258, Economic and Social Research Institute (ESRI). URL <https://ideas.repec.org/p/esj/esridp/258.html>.
- Andrle, M. & Hunt, M. B. L. (2020). Model-Based Globally-Consistent Risk Assessment. IMF Working Papers 2020/064, International Monetary Fund. URL <https://ideas.repec.org/p/imf/imfwpa/2020-064.html>.
- Aruoba, S. B., Fernandez-Villaverde, J., & Rubio-Ramirez, J. F. (2006). Comparing Solution Methods for Dynamic Equilibrium Economies. *Journal of Economic Dynamics and Control*, 30(12), 2477–2508. URL <https://ideas.repec.org/a/eee/dyncon/v30y2006i12p2477-2508.html>.
- Ascari, G. & Rossi, L. (2012). Trend Inflation and Firms Price-Setting: Rotemberg Versus Calvo. *Economic Journal*, 122(563), 1115–1141. URL <https://EconPapers.repec.org/RePEc:ecj:econjl:v:122:y:2012:i:563:p:1115-1141>.
- Beneš, J., Binning, A., & Lees, K. (2008). Incorporating Judgement with DSGE Models. Reserve Bank of New Zealand Discussion Paper Series DP2008/10, Reserve Bank of New Zealand. URL <https://ideas.repec.org/p/nzb/nzbdps/2008-10.html>.
- Binning, A. (2021). Shocks and Scenarios Analysis Using a Stochastic Neoclassical Growth Model. Background Paper for the 2021 Statement on the Long-term Fiscal Position, The New Zealand Treasury. URL [https://www.treasury.govt.nz/sites/default/files/2021-09/lfts\\_bp-shocks-scenarios.pdf](https://www.treasury.govt.nz/sites/default/files/2021-09/lfts_bp-shocks-scenarios.pdf).
- Boucekkine, R. (1995). An Alternative Methodology for Solving Nonlinear Forward-Looking Models. *Journal of Economic Dynamics and Control*, 19(4), 711–734. URL <https://EconPapers.repec.org/RePEc:eee:dyncon:v:19:y:1995:i:4:p:711-734>.
- Campbell, J. R., Evans, C. L., Fisher, J. D., & Justiniano, A. (2012). Macroeconomic Effects of Federal Reserve Forward Guidance. *Brookings Papers on Economic Activity*, 43(1 (Spring)), 1–80. URL <https://ideas.repec.org/a/bin/bpeajo/v43y2012i2012-01p1-80.html>.
- de Groot, O., Mazelis, F., Motto, R., & Ristiniemi, A. (2021). A Toolkit for Computing Constrained Optimal Policy Projections (COPPs). Working Paper Series 2555, European Central Bank. URL <https://ideas.repec.org/p/ecb/ecbwps/20212555.html>.
- Doan, T., Litterman, R., & Sims, C. (1983). Forecasting and Conditional Projection Using Realistic Prior Distributions. NBER Working Papers 1202, National Bureau of Economic Research, Inc. URL <https://EconPapers.repec.org/RePEc:nbr:nberwo:1202>.
- Fair, R. & Taylor, J. (1983). Solution and Maximum Likelihood Estimation of Dynamic Nonlinear Rational Expectations Models. *Econometrica*, 51(4), 1169–85. URL <https://EconPapers.repec.org/RePEc:ecm:emetrp:v:51:y:1983:i:4:p:1169-85>.
- Giannoni, M., Patterson, C., & Del Negro, M. (2015). The Forward Guidance Puzzle. 2015 Meeting Papers 1529, Society for Economic Dynamics. URL <https://EconPapers.repec.org/RePEc:red:sed015:1529>.
- Goodfriend, M. & King, R. G. (2005). The Incredible Volcker Disinflation. *Journal of Monetary Economics*, 52(5), 981–1015. URL <https://ideas.repec.org/a/eee/moneco/v52y2005i5p981-1015.html>.
- Heer, B. & Maussner, A. (2009). *Dynamic General Equilibrium Modeling: Computational Methods and Applications*. Springer Berlin Heidelberg. URL <https://books.google.co.nz/books?id=ZdlEAAAAQBAJ>.

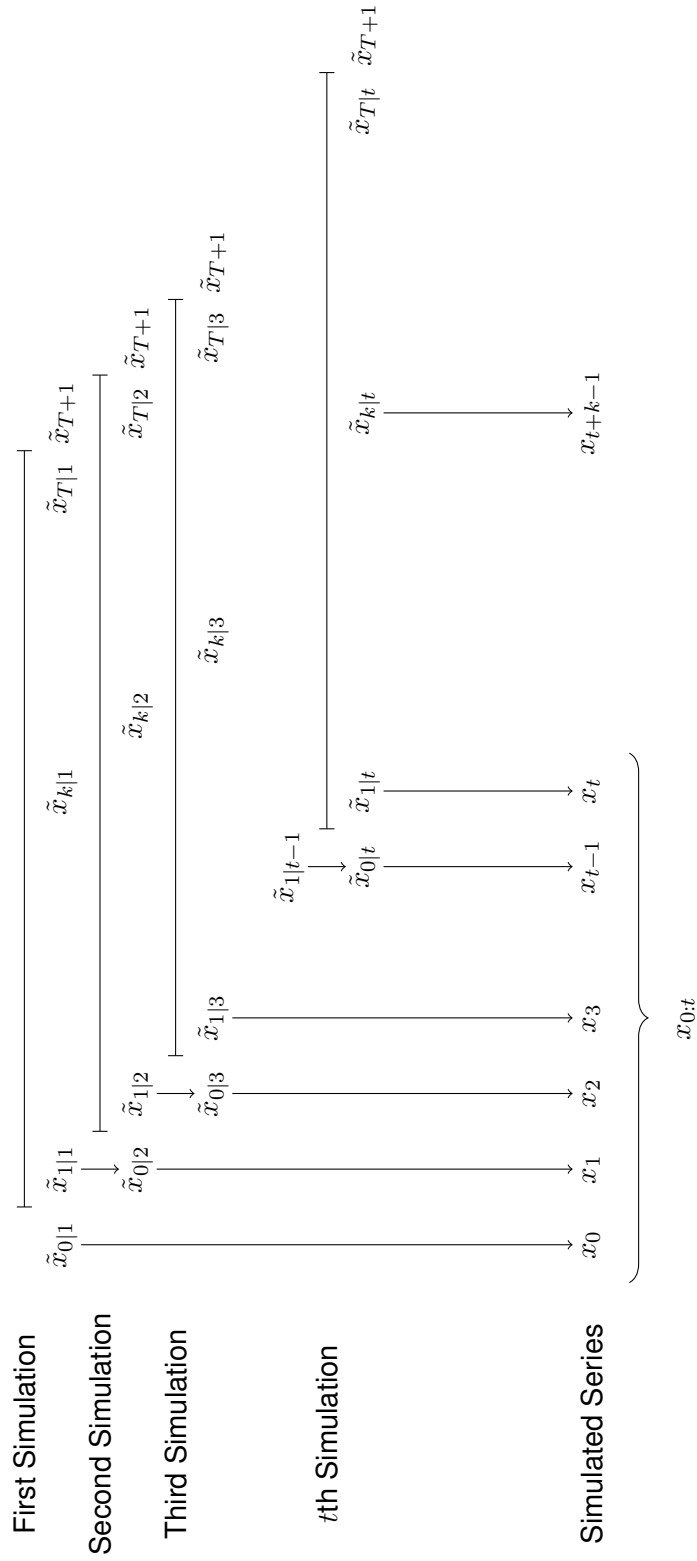


- Hills, T. S. & Nakata, T. (2018). Fiscal Multipliers at the Zero Lower Bound: The Role of Policy Inertia. *Journal of Money, Credit and Banking*, 50(1), 155–172. URL <https://ideas.repec.org/a/wly/jmoncb/v50y2018i1p155-172.html>.
- Holden, T. & Paetz, M. (2012). Efficient Simulation of DSGE Models with Inequality Constraints. School of Economics Discussion Papers 1612, School of Economics, University of Surrey. URL <https://EconPapers.repec.org/RePEc:sur:surrec:1612>.
- Jones, C., Kulish, M., & Rees, D. (2020). International Spillovers of Forward Guidance Shocks. BIS Working Papers 870, Bank for International Settlements. URL <https://ideas.repec.org/p/bis/biswps/870.html>.
- Laséen, S. & Svensson, L. E. (2011). Anticipated Alternative Policy Rate Paths in Policy Simulations. *International Journal of Central Banking*, 7(3), 1–35. URL <https://ideas.repec.org/a/ijc/ijcjou/y2011q3a1.html>.
- Lindé, J., Smets, F., & Wouters, R. (2016). Challenges for Central Banks' Macro Models. CEPR Discussion Papers 11405, C.E.P.R. Discussion Papers. URL <https://ideas.repec.org/p/cpr/ceprdp/11405.html>.
- Maih, J. (2010). Conditional Forecasts in DSGE Models. Working Paper 2010/07, Norges Bank. URL [https://ideas.repec.org/p/bno/worpaper/2010\\_07.html](https://ideas.repec.org/p/bno/worpaper/2010_07.html).
- Maliar, L., Maliar, S., Taylor, J. B., & Tsener, I. (2020). A Tractable Framework for Analyzing a Class of Nonstationary Markov Models. *Quantitative Economics*, 11(4), 1289–1323. URL <https://ideas.repec.org/a/wly/quante/v11y2020i4p1289-1323.html>.
- Rotemberg, J. (1982). Monopolistic Price Adjustment and Aggregate Output. *Review of Economic Studies*, 49(4), 517–531. URL <https://EconPapers.repec.org/RePEc:oup:restud:v:49:y:1982:i:4:p:517-531>.
- Rysanek, J. (2021). Symbolic Derivatives. URL <https://www.mathworks.com/matlabcentral/fileexchange/59438-symbolic-derivatives>.
- Schmitt-Grohé, S. & Uribe, M. (2004). Solving Dynamic General Equilibrium Models Using a Second-Order Approximation to the Policy Function. *Journal of Economic Dynamics and Control*, 28(4), 755–775. URL <https://ideas.repec.org/a/eee/dyncon/v28y2004i4p755-775.html>.
- Waggoner, D. F. & Zha, T. (1999). Conditional Forecasts In Dynamic Multivariate Models. *The Review of Economics and Statistics*, 81(4), 639–651. URL <https://ideas.repec.org/a/tpr/restat/v81y1999i4p639-651.html>.
- Wascher, W. & Andersen, P. S. (1999). Sacrifice Ratios and the Conduct of Monetary Policy in Conditions of Low Inflation. BIS Working Papers 82, Bank for International Settlements. URL <https://ideas.repec.org/p/bis/biswps/82.html>.



## A. The Stochastic Extended Path Algorithm

**Figure 16 — A Diagrammatic Representation of the Stochastic Extended Path Algorithm**



## B. Speed Tests

In this appendix I carry out some simple speed tests to highlight the efficiency gains from different features of the code used to produce the examples in this paper. In particular I compare the speed gains from the vectorised code against code that uses for-loops, the speed gains from using symbolic derivatives in place of quasi-Newton methods to solve models with occasionally binding constraints, and the speed gains from using the linearised model with occasionally binding constraints applied as opposed to the full nonlinear model and solution. All the speed tests are carried out using Matlab 2021a on a desktop computer with a 2.60 GHz Intel i5 processor and 8GB RAM. All times are reported in seconds.

I test the vectorised code against the code that uses for-loops to fill in and update the equation residuals and Jacobian matrices using two models, the simple New Keynesian DSGE model outlined in Section 2 and a medium sized DSGE model with 82 endogenous variables. The simple New Keynesian model is stochastically simulated for 2000 periods, using the same sequence of shocks for both solution methods. I set the solution horizon,  $T$ , to 200 periods, which is sufficient to ensure the model reaches steady state after the model is hit by a battery of shocks. I run each solution method 10 times and report the median time in seconds for each method in Table 6 below.

**Table 6 – Simple New Keynesian Model: Vectorised Code vs. For-Loops**

Model and Solution Method	Time	Relative Gain
Non-linear, vectorised code	53.84s	
Non-linear, for-loops	52.96s	1.63%

In this example the vectorised code is actually slower than the code that uses for-loops. Closer inspection reveals that while the vectorised code that fills in and updates the equation residuals is faster than the code that uses for-loops for the same task, the vectorised code is slower at filling in and updating the Jacobian matrix. This in part reflects the size of the model and the improvements that Mathworks has made in the performance of for-loops in recent years. When the same experiment is performed using Matlab 2015b, the vectorised code comfortably beats the code using for-loops.

To see how the model size affects the results, I stochastically simulate a medium sized DSGE model for 100 periods, using the same sequence of shocks for both solution codes. I set the solution horizon,  $T$ , to 1000 periods, to ensure the model reaches steady state after a battery of shocks hit the model. The medium sized DSGE requires a longer solution horizon than the simple New Keynesian DSGE model because it includes government debt, private debt and physical capital, all of which can be quite persistent. I run each code 10 times and report the median time in seconds for each code in Table 7 below.

**Table 7 – Medium Sized Policy Model: Vectorised Code vs. For-Loops**

Model and Solution Method	Time	Relative Gain
Non-linear, vectorised code	94.81s	12.90%
Non-linear, for-loops	108.86s	

When the vectorised code is used to solve a medium sized DSGE model it is 12.90% faster than the code that uses for-loops.

To investigate the benefits of using symbolic derivatives with occasionally binding constraints, I stochastically simulate the simple New Keynesian model for 1000 periods with a binding ELB constraint. I repeat the experiment with the same sequence of shocks, solving the model using a sparse application of Broyden's method, which is a numerical approach for calculating and updating the Jacobian matrix. I switch off the monetary policy shock and double the standard deviation of the technology and preference shocks to ensure the model hits the lower bound in a sufficient number of periods. I set the solution horizon,  $T$ , to 200 periods and I run each code 10 times and report the median time in seconds for each code in Table 8 below.

**Table 8 – Symbolic Derivatives With OBC's vs. Quasi-Newton Methods With OBC's**

Model and Solution Method	Time	Relative Gain
Non-linear model, OBC's, symbolic derivatives	32.54s	63.43%
Non-linear model, OBC's, Broyden's method	88.98s	

The simulated model spends 196 periods out of 1000 simulation periods at the lower bound. The code implemented with symbolic derivatives is 63.43% faster than the sparse implementation of Broyden's method.

I compare the linearised model solved with occasionally binding constraints against the non-linear model with occasionally binding constraints solved using the vectorised code with symbolic derivatives. I stochastically simulate the simple New Keynesian model for 1000 periods with a binding ELB constraint, using the same sequence of shocks for both experiments. As in the previous simulation experiment, I switch off the monetary policy shock and increase the standard deviations of the productivity and preference shocks to ensure that the model hits the lower bound in a reasonable number of periods. I set the solution horizon,  $T$ , to 200 periods and I run each code 10 times and report the median time in seconds for each code in Table 9 below.

**Table 9 – Linearised Model With OBC's vs. Non-Linear Model With OBC's**

Model and Solution Method	Time	Relative Gain
Linearised, OBC's, symbolic derivatives	9.24s	71.56%
Non-linear, OBC's, symbolic derivatives	32.54s	

The linearised code with a binding lower bound constraint is 71.56% faster than the full nonlinear model solved using the vectorised code with symbolic derivatives.

## C. The Linearised Model and Occasionally Binding Constraints

In this appendix I describe how occasionally binding constraints can be applied to the linearised model and solved using the extended path algorithm. The dynamics of many non-linear rational expectations models are close to linear in the absence of occasionally binding constraints for plausible parameterisations and shocks. In such models it is often the introduction of occasionally binding constraints that is the main source of asymmetry and non-linearity. Applying the occasionally binding constraint to the linearised model speeds up the algorithm, as in general fewer iterations are needed to solve the model. I use Jakub Rysanek's symbolic differentiator procedures in Matlab to write code that linearises the non-linear model and then applies the occasionally binding constraints (see Rysanek 2021). By automating the process I save time and avoid the possibility of mistakes that arise from linearising the model by hand.

I demonstrate the algorithm using the non-linear model in equation (50) from Section 5.4.5,

$$f(E_t \{x_{t+1}\}, x_t, x_{t-1}, \varepsilon_t) = \begin{bmatrix} Z_t (Y_t - \chi Y_{t-1})^{-\sigma} - \beta \frac{R_t}{\pi_{t+1}} Z_{t+1} (Y_{t+1} - \chi Y_t)^{-\sigma} \\ \left(\frac{\phi}{v-1}\right) \pi_t \left[\pi_t - \pi_{t-1}^\psi \pi^{1-\psi}\right] - \left(\frac{v}{v-1}\right) \kappa (Y_t - \chi Y_{t-1})^\sigma Y_t^\eta A_t^{-(1+\eta)} + 1 - \dots \\ \dots - \left(\frac{\phi}{v-1}\right) \beta \left(\frac{Z_{t+1}}{Z_t}\right) \left(\frac{Y_{t+1} - \chi Y_t}{Y_t - \chi Y_{t-1}}\right)^{-\sigma} \left(\frac{Y_{t+1}}{Y_t}\right) \pi_{t+1} \left[\pi_{t+1} - \pi_t^\psi \pi^{1-\psi}\right] \\ R_t^* - (R_{t-1}^*)^{\rho_r} \left(R \left(\frac{\pi_t}{\pi}\right)^{\kappa_\pi} \left(\frac{Y_t}{Y}\right)^{\kappa_Y}\right)^{1-\rho_r} \exp(\sigma_R \varepsilon_{R,t}) \\ \log A_t - \rho_A \log A_{t-1} - \sigma_A \varepsilon_{A,t} \\ \log Z_t - \rho_Z \log Z_{t-1} - \sigma_Z \varepsilon_{Z,t} \\ R_t - R_{ELB} - X_t (R_t^* - R_{ELB,t}) \end{bmatrix} = 0. \quad (95)$$

The vector of endogenous variables for this model is given by,

$$x_t = [Y_t, \pi_t, R_t, A_t, Z_t, R_t^*], \quad (96)$$

and the vector of unexpected exogenous variables is given by,

$$\varepsilon_t = [\varepsilon_{Z,t}, \varepsilon_{R,t}, \varepsilon_{A,t}]. \quad (97)$$

I introduce an additional vector of anticipated exogenous variables,

$$\Upsilon_t = [X_t, R_{ELB,t}], \quad (98)$$

where  $X_t$  is an indicator variable and  $R_{ELB,t}$  is the value of the interest rate at the effective lower bound. The indicator variable,  $X_t$ , will eventually be replaced by a function of endogenous variables when the lower bound regime is endogenously determined, but for the moment I treat  $X_t$  as exogenous. The time paths for the model variables can be described by the full set of model equations for each time period, as before,

$$F(x_{1:T}) = \begin{bmatrix} f(x_2, x_1, x_0, \varepsilon_t) \\ \vdots \\ f(x_{t+1}, x_t, x_{t-1}, \varepsilon_t) \\ \vdots \\ f(x_{T+1}, x_T, x_{T-1}, \varepsilon_t) \end{bmatrix}. \quad (99)$$

The non-linear model can be approximated by the linear system as follows,

$$F(x_{1:T}) \approx \mathcal{F}(\hat{x}_{1:T}) = \mathcal{J}_x(\hat{x}_{1:T}) \hat{x}_{0:T+1} + \mathcal{J}_\varepsilon(\hat{x}_{1:T}) \varepsilon_{1:T} + \mathcal{J}_\Upsilon(\hat{x}_{1:T}) \Upsilon_{1:T}, \quad (100)$$

where,

$$\mathcal{J}_x(\hat{x}_{1:T}) = \frac{\partial F(x_{0:T+1}; \varepsilon_t, \Upsilon_t)}{\partial x_{0:T+1}} \bigg|_{x_{0:T+1}=x, \varepsilon_{1:T}=0, \Upsilon_{1:T}=\Upsilon(\hat{x}_{1:T})}, \quad (101)$$

is the Jacobian matrix for the endogenous variables, evaluated at the steady state,

$$\mathcal{J}_\varepsilon(\hat{x}_{1:T}) = \frac{\partial F(x_{0:T+1}; \varepsilon_t, \Upsilon_t)}{\partial \varepsilon_{1:T}} \bigg|_{x_{0:T+1}=x, \varepsilon_{1:T}=0, \Upsilon_{1:T}=\Upsilon(\hat{x}_{1:T})}, \quad (102)$$

is the Jacobian matrix for the unanticipated exogenous variables, evaluated at the steady state, and

$$\mathcal{J}_\Upsilon(\hat{x}_{1:T}) = \frac{\partial F(x_{0:T+1}; \varepsilon_t, \Upsilon_t)}{\partial \Upsilon_{1:T}} \bigg|_{x_{0:T+1}=x, \varepsilon_{1:T}=0, \Upsilon_{1:T}=\Upsilon(\hat{x}_{1:T})}, \quad (103)$$

is the Jacobian of the anticipated exogenous variables, also evaluated at the steady state. Note that  $\Upsilon_{1:T} = \Upsilon(\hat{x}_{1:T})$ , indicating that after the derivatives have been taken, the indicator variables,  $X_t$ , are replaced with the inequalities,<sup>29</sup>

$$X_t = \begin{cases} 0 & \text{if } R_t^* \leq R_{ELB} \\ 1 & \text{if } R_t^* > R_{ELB} \end{cases}. \quad (104)$$

Following Boucekkine (1995), I write the Jacobian matrix for the endogenous variables as follows,

$$\mathcal{J}_x(\hat{x}_{1:T}) = \begin{bmatrix} \mathcal{A}_1 & \mathcal{B}_1 & \mathcal{C}_1 & 0 & \dots & & \dots & 0 \\ 0 & \mathcal{A}_2 & \mathcal{B}_2 & \mathcal{C}_2 & 0 & \dots & & \vdots \\ \vdots & 0 & \mathcal{A}_3 & \mathcal{B}_3 & \mathcal{C}_3 & 0 & \dots & \\ & & & \ddots & \ddots & \ddots & & \\ & & \dots & 0 & \mathcal{A}_t & \mathcal{B}_t & \mathcal{C}_t & 0 & \dots \\ & & & & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \dots & 0 & \mathcal{A}_{T-1} & \mathcal{B}_{T-1} & \mathcal{C}_{T-1} & 0 \\ 0 & \dots & & \dots & 0 & \mathcal{A}_T & \mathcal{B}_T & \mathcal{C}_T \end{bmatrix}, \quad (105)$$

where,

$$\begin{aligned} \mathcal{A}_t &= \frac{\partial f(x_{t+1}, x_t, x_{t-1}, \varepsilon_t, \Upsilon_t)}{\partial x_{t-1}} \bigg|_{x_{t+1}, x_t, x_{t-1}=x, \varepsilon_t=0, \Upsilon_t=\Upsilon(\hat{x}_t)}, \\ \mathcal{B}_t &= \frac{\partial f(x_{t+1}, x_t, x_{t-1}, \varepsilon_t, \Upsilon_t)}{\partial x_t} \bigg|_{x_{t+1}, x_t, x_{t-1}=x, \varepsilon_t=0, \Upsilon_t=\Upsilon(\hat{x}_t)}, \\ \mathcal{C}_t &= \frac{\partial f(x_{t+1}, x_t, x_{t-1}, \varepsilon_t, \Upsilon_t)}{\partial x_{t+1}} \bigg|_{x_{t+1}, x_t, x_{t-1}=x, \varepsilon_t=0, \Upsilon_t=\Upsilon(\hat{x}_t)}. \end{aligned}$$

The Jacobian matrix for the unanticipated exogenous variables has the form,

$$\mathcal{J}_\varepsilon(\hat{x}_{1:T}) = \begin{bmatrix} \mathcal{D}_1 & 0 & \dots & \dots & 0 \\ 0 & \mathcal{D}_2 & 0 & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ \vdots & \dots & 0 & \mathcal{D}_t & 0 \\ \vdots & & & \ddots & 0 \\ 0 & \dots & \dots & 0 & \mathcal{D}_T \end{bmatrix}, \quad (106)$$

<sup>29</sup> For the same reasons outlined in Section 5.4.1, derivatives of the indicator function do not need to be taken to solve the model with occasionally binding constraints.

where,

$$\mathcal{D}_t = \frac{\partial f(x_{t+1}, x_t, x_{t-1}, \varepsilon_t, \Upsilon_t)}{\partial \varepsilon_t} \Big|_{x_{t+1}, x_t, x_{t-1}=x, \varepsilon_t=0, \Upsilon_t=\Upsilon(\hat{x}_t)}. \quad (107)$$

The Jacobian matrix for the anticipated exogenous variables has the form,

$$\mathcal{J}_\Upsilon(\hat{x}_{1:T}) = \begin{bmatrix} \mathcal{G}_1 & 0 & \dots & \dots & 0 \\ 0 & \mathcal{G}_2 & 0 & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ \vdots & \dots & 0 & \mathcal{G}_t & 0 \\ \vdots & & & \ddots & 0 \\ 0 & \dots & \dots & 0 & \mathcal{G}_T \end{bmatrix}, \quad (108)$$

where,

$$\mathcal{G}_t = \frac{\partial f(x_{t+1}, x_t, x_{t-1}, \varepsilon_t, \Upsilon_t)}{\partial \Upsilon_t} \Big|_{x_{t+1}, x_t, x_{t-1}=x, \varepsilon_t=0, \Upsilon_t=\Upsilon(\hat{x}_t)}. \quad (109)$$

Equation (100) can then be rewritten as,

$$\begin{aligned} \mathcal{F}(\hat{x}_{1:T}) = & \begin{bmatrix} \mathcal{A}_1 & \mathcal{B}_1 & \mathcal{C}_1 & 0 & \dots & \dots & 0 \\ 0 & \mathcal{A}_2 & \mathcal{B}_2 & \mathcal{C}_2 & 0 & \dots & \vdots \\ \vdots & 0 & \mathcal{A}_3 & \mathcal{B}_3 & \mathcal{C}_3 & 0 & \dots \\ & & & \ddots & \ddots & \ddots & \\ & & \dots & 0 & \mathcal{A}_t & \mathcal{B}_t & \mathcal{C}_t & 0 & \dots \\ & & & & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \dots & 0 & \mathcal{A}_{T-1} & \mathcal{B}_{T-1} & \mathcal{C}_{T-1} & 0 \\ 0 & \dots & & \dots & 0 & \mathcal{A}_T & \mathcal{B}_T & \mathcal{C}_T \end{bmatrix} \begin{bmatrix} \hat{x}_0 \\ \hat{x}_1 \\ \vdots \\ \hat{x}_T \\ \hat{x}_{T+1} \end{bmatrix} + \dots \\ & \dots + \begin{bmatrix} \mathcal{D}_1 & 0 & \dots & \dots & 0 \\ 0 & \mathcal{D}_2 & 0 & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ \vdots & \dots & 0 & \mathcal{D}_t & 0 \\ \vdots & & & \ddots & 0 \\ 0 & \dots & \dots & 0 & \mathcal{D}_T \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} \mathcal{G}_1 & 0 & \dots & \dots & 0 \\ 0 & \mathcal{G}_2 & 0 & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ \vdots & \dots & 0 & \mathcal{G}_t & 0 \\ \vdots & & & \ddots & 0 \\ 0 & \dots & \dots & 0 & \mathcal{G}_T \end{bmatrix} \begin{bmatrix} \Upsilon_1 \\ \Upsilon_2 \\ \vdots \\ \Upsilon_t \\ \vdots \\ \Upsilon_T \end{bmatrix}. \quad (110) \end{aligned}$$

The Newton step in the extended path algorithm can then be replaced with,

$$\hat{x}_{1:T}^{h+1} = \hat{x}_{1:T}^h - J(\hat{x}_{1:T}^h)^{-1} \mathcal{F}(\hat{x}_{1:T}^h), \quad (111)$$

or substituting for  $\mathcal{F}(\hat{x}_{1:T}^h)$ ,

$$\hat{x}_{1:T}^{h+1} = \hat{x}_{1:T}^h - J(\hat{x}_{1:T}^h)^{-1} \left[ \mathcal{J}_x(\hat{x}_{1:T}) \hat{x}_{0:T+1}^h + \mathcal{J}_\varepsilon(\hat{x}_{1:T}) \varepsilon_{1:T} + \mathcal{J}_\Upsilon(\hat{x}_{1:T}) \Upsilon_{1:T} \right], \quad (112)$$

where  $\hat{x}_0$  and  $\hat{x}_{T+1}$  in  $\hat{x}_{0:T+1}^h$  are given.