2013 | 18

Working Paper

Monetary Policy

Solving second and third-order approximations to DSGE models: A recursive Sylvester equation solution

Andrew Binning

NB NORGES BANK

Working papers fra Norges Bank, fra 1992/1 til 2009/2 kan bestilles over e-post:
servicesenter@norges-bank.no

Fra 1999 og senere er publikasjonene tilgjengelige på www.norges-bank.no

Working papers inneholder forskningsarbeider og utredninger som vanligvis ikke har fått sin endelige form. Hensikten er blant annet at forfatteren kan motta kommentarer fra kolleger og andre interesserte. Synspunkter og konklusjoner i arbeidene står for forfatternes regning.

Working papers from Norges Bank, from 1992/1 to 2009/2 can be ordered by e-mail:
servicesenter@norges-bank.no

Working papers from 1999 onwards are available on www.norges-bank.no

Norges Bank's working papers present research projects and reports (not usually in their final form) and are intended inter alia to enable the author to benefit from the comments of colleagues and other interested parties. Views and conclusions expressed in working papers are the responsibility of the authors alone.

ISSN 1502-8143 (online)
ISBN 978-82-7553-771-1 (online)

Solving second and third-order approximations to DSGE models: a recursive Sylvester equation solution

Andrew Binning[1,2]

29 July 2013

Monetary Policy Department, Norges Bank, Oslo, Norway

Abstract

In this paper I derive the matrix chain rules for solving a second and a third-order approximation to a DSGE model that allow the use of a recursive Sylvester equation solution method. In particular I use the solution algorithms of Kamenik (2005) and Martin & Van Loan (2006) to solve the generalised Sylvester equations. Because I use matrix algebra instead of tensor notation to find the system of equations, I am able to provide standalone Matlab routines that make it feasible to solve a medium scale DSGE model in a competitive time. I also provide Fortran code and Matlab/Fortran mex files for my method.

Keywords: Solving dynamic models, Second-order approximation, Third-order approximation, Second-order matrix chain rule, Third-order matrix chain rule, Generalised Sylvester equations

1. Introduction

Solving higher order approximations of DSGE models can be computationally demanding at best. As the size of the model increases, the number of coefficients that need to be solved increases at a greater rate, a feature commonly referred to as the curse of dimensionality. Using simple matrix algebra to find the unknown coefficients can place quite severe limitations on the model's size as memory capacity becomes an issue. The use of generalised Sylvester equations has been suggested by Gomme & Klein (2011) as a more memory efficient approach to solving higher order approximations of DSGE models. In particular they use the Kågström & Poromaa (1996) representation for the generalised Sylvester equations. Kamenik (2005) presents an alternative Sylvester equation representation and solution method

Email address: andrew.binning@norges-bank.no (Andrew Binning)

[1]Any opinions expressed here do not necessarily reflect the views of the management of the Norges Bank.

[2]The author would like to thank Martin Andreasen, Gisle Natvik, Martin Seneca and seminar participants at the Norges Bank for useful comments. All remaining errors are my own.

that exploits the Kronecker product structure of the problem allowing it to be solved recursively. This results in significant performance improvements over existing solution methods (see Kamenik (2005) for a comparison with other methods of solving generalised Sylvester equations). Representing the problem as a system of generalised Sylvester equations is key to developing a fast and efficient solution method. The method for finding the matrices in the generalised Sylvester equations also plays a significant role in the performance of the solution method. It is common to use chain rules written in tensor notation to find these matrices (see Schmitt-Grohe & Uribe (2004), Ruge-Murcia (2010), Andreasen (2011) and Kamenik (2005)), although this is not the most efficient method. In this paper I derive second and third-order matrix chain rules that with a small amount of manipulation, can be written in the generalised Sylvester equation form outlined in Kamenik (2005). These matrix chain rules are easier to code, easier to write out and understand, and fast to implement when combined with a recursive Sylvester equation solution algorithm.

Tensor notation has become a popular method for representing the chain rules used in the solution of higher order approximations of DSGE models. Schmitt-Grohe & Uribe (2004) use tensor notation to find the matrices in the solution of a second-order approximation. Ruge-Murcia (2010) and Andreasen (2011) extend this tensor notation representation of the chain rule to solving third-order approximations. Kamenik (2005) uses tensor notation to write out the nth order chain rules consistent with the representation of his generalised Sylvester equations. While popular, there are limitations to using tensor notation, in particular tensor notation is difficult to understand, difficult to code and is slow to implement when using Matlab (see Binning, 2013). An alternative approach to using tensor notation uses matrix chain rules to represent the problem. Gomme & Klein (2011) use the Magnus & Neudecker (1999) definition of a Hessian to find a second-order approximation. Binning (2013) extends the approach of Gomme & Klein (2011) to find a matrix chain rule for third-order approximations. The matrix chain-rules described in these papers can be solved using the generalised Sylvester equation algorithm of Kågström & Poromaa (1996) (as demonstrated in Gomme & Klein, 2011), but they are not consistent with the more efficient solution algorithm of Kamenik (2005). However, the matrix chain rules in Gomme & Klein (2011) and Binning (2013) are not unique.

In this paper I derive a second and a third-order matrix chain rule, that with a small amount of algebra, can be rearranged into the type of generalised Sylvester equations in Kamenik (2005). Then I apply the recursive Sylvester equation solution algorithm of Kamenik (2005) to find the unknown coefficient matrices for the second and third-order approximate solutions. This avoids the use of tensor notation, resulting in a solution procedure that is much easier to write and code, and feasible to implement in Matlab, the resulting code can solve a medium size DSGE model in a competitive time.[3] I also show how to use a similar algorithm by Martin & Van Loan (2006) to solve the system of generalised Sylvester equa-

[3] The equivalent Matlab code using tensor notation would be significantly slower due to the speed with which Matlab implements For loops. Dynare++ uses the Kamenik algorithm and tensor notation to solve nth order approximations but is coded in C++ due to Matlab's limitations.

tions and I compare the performance of both algorithms. In addition to providing Matlab code for my solution method, I also provide Fortran and Matlab/Fortran mex code.[4]

The remainder of the paper is set out as follows; section 2 outlines the general problem and the form the solutions take. In section 3 I present the second and third-order matrix chain rules and in section 4, I give a brief description of the generalised Sylvester equation solution algorithms. Sections 5 and 6 present the matrix chain rules for a second and a third-order approximation of a DSGE model respectively. They also demonstrate the steps required to get these matrices into the appropriate generalised Sylvester equation form. In section 7 I demonstrate the performance of the algorithm using some small and medium sized DSGE models, while section 8 concludes.

2. Preliminaries

Following Schmitt-Grohe & Uribe (2004) a large set of DSGE models can be recast in the following form

$$\mathrm{E}_t\left(f\left(x_{t+1}, y_{t+1}, x_t, y_t\right)\right) = 0, \tag{1}$$

where x_{t+1} is an $nx \times 1$ vector of the date $t+1$ predetermined variables and y_{t+1} is an $ny \times 1$ vector of the date $t+1$ non-predetermined variables, f is a function that maps $\mathbb{R}^{2nx+2ny}$ into $\mathbb{R}^{nx+ny}$, and E_t is the expectations operator conditional on date t information. The total number of variables (and equations) in the model is $n = nx + ny$.

As shown in Schmitt-Grohe & Uribe (2004) a solution to equation (1) takes the form:

$$x_{t+1} = h(x_t, \sigma) + \sigma\varepsilon_{t+1}, \tag{2}$$

$$y_t = g(x_t, \sigma), \tag{3}$$

where $h(\cdot)$ is a policy function that maps x_t into x_{t+1}, σ is the perturbation parameter, ε_{t+1} is an $nx \times 1$ vector of expectation errors and $g(\cdot)$ is a policy function that maps x_t into y_t.

Typically the functions $h(\cdot)$ and $g(\cdot)$ are unknown, and in general they are non-linear and do not have exact analytical forms. Because an exact solution does not exist an approximate solution must be found. A common approximation strategy involves finding the Taylor series expansion of the policy functions around the non-stochastic steady state. This usually involves taking a first-order approximation of the policy functions. The resulting linear/log-linear solution will be adequate for many problems. However taking a first-order approximation introduces certainty equivalence into the solution which may be inappropriate

[4]Dynare/Dynare++ is the main alternative for solving third-order approximations of medium sized DSGE models. However Dynare/Dynare++ package the routines in such a way that it makes it difficult to combine them with other Matlab code. For example it would require some knowledge to integrate the Dynare/Dynare++ solution routines into an external estimation procedure in an efficient way. The routines I present in this paper are standalone, meaning they do not rely on other toolboxes to run and are therefore easy to combine with existing Matlab code and/or programs, they have similar performance to Dynare/Dynare++, and are therefore a natural choice for practitioners developing procedures for estimating non-linear DSGE models.

when studying the effects of risk, or when performing welfare analysis. There may also be important asymmetries in the model that would be lost if only a first-order approximation of the model were taken (see Kim & Ruge-Murcia, 2011). Solving a second-order approximation introduces a constant correction for the effects of risk, while taking a third-order approximation introduces a time varying risk term and an additional intercept correction for the effect of skewed shocks. The increased computational demands, even with the smallest of models, combined with only modest improvements in accuracy mean fourth and higher order approximations are not commonly implemented. As will be explained in more detail in this section, solving a second-order approximation requires the solution to the first-order approximation, and solving a third-order approximation requires the solutions to both the first and second-order approximations.

I follow such a strategy and obtain the second-order approximation of the policy functions (equations (2) and (3))[5]

$$x_{t+1} = h_x x_t + \tfrac{1}{2}\sigma^2 h_{\sigma\sigma} + \tfrac{1}{2} h_{xx}\left(x_t \otimes x_t\right) + \sigma \varepsilon_{t+1}, \tag{4}$$

$$y_t = g_x x_t + \tfrac{1}{2}\sigma^2 g_{\sigma\sigma} + \tfrac{1}{2} g_{xx}\left(x_t \otimes x_t\right). \tag{5}$$

The coefficient matrices for the first order terms g_x and h_x are given by

$$\underset{nx\times nx}{h_x} = \begin{bmatrix} \frac{\partial h^1}{\partial x_{1,t}} & \cdots & \frac{\partial h^1}{\partial x_{i,t}} & \cdots & \frac{\partial h^1}{\partial x_{nx,t}} \\ \vdots & & \vdots & & \vdots \\ \frac{\partial h^q}{\partial x_{1,t}} & \cdots & \frac{\partial h^q}{\partial x_{i,t}} & \cdots & \frac{\partial h^q}{\partial x_{nx,t}} \\ \vdots & & \vdots & & \vdots \\ \frac{\partial h^{nx}}{\partial x_{1,t}} & \cdots & \frac{\partial h^{nx}}{\partial x_{i,t}} & \cdots & \frac{\partial h^{nx}}{\partial x_{nx,t}} \end{bmatrix}, \quad \underset{ny\times nx}{g_x} = \begin{bmatrix} \frac{\partial g^1}{\partial x_{1,t}} & \cdots & \frac{\partial g^1}{\partial x_{i,t}} & \cdots & \frac{\partial g^1}{\partial x_{nx,t}} \\ \vdots & & \vdots & & \vdots \\ \frac{\partial g^r}{\partial x_{1,t}} & \cdots & \frac{\partial g^r}{\partial x_{i,t}} & \cdots & \frac{\partial g^r}{\partial x_{nx,t}} \\ \vdots & & \vdots & & \vdots \\ \frac{\partial g^{ny}}{\partial x_{1,t}} & \cdots & \frac{\partial g^{ny}}{\partial x_{i,t}} & \cdots & \frac{\partial g^{ny}}{\partial x_{nx,t}} \end{bmatrix},$$

where $h^q = h^q(x_t, \sigma)$ is the policy function for the qth predetermined variable for $q = 1, \cdots, nx$ and $g^r = g^r(x_t, \sigma)$ is the policy function for the rth non-predetermined variable for $r = 1, \cdots, ny$. The matrices g_x and h_x can be found using the algorithm described in Klein (2000). The remaining terms in equations (4) and (5): g_{xx}, h_{xx}, $g_{\sigma\sigma}$ and $h_{\sigma\sigma}$, are the second derivatives of the policy functions and are defined as follows:

$$\underset{nx\times nx^2}{h_{xx}} = \begin{bmatrix} h_{x,x_1} & \cdots & h_{x,x_j} & \cdots & h_{x,x_{nx}} \end{bmatrix}, \quad \underset{ny\times nx^2}{g_{xx}} = \begin{bmatrix} g_{x,x_1} & \cdots & g_{x,x_j} & \cdots & g_{x,x_{nx}} \end{bmatrix},$$

$$\underset{nx\times 1}{h_{\sigma\sigma}} = \begin{bmatrix} \frac{\partial^2 h^1}{\partial \sigma^2} \\ \vdots \\ \frac{\partial^2 h^q}{\partial \sigma^2} \\ \vdots \\ \frac{\partial^2 h^{nx}}{\partial \sigma^2} \end{bmatrix}, \quad \underset{ny\times 1}{g_{\sigma\sigma}} = \begin{bmatrix} \frac{\partial^2 g^1}{\partial \sigma^2} \\ \vdots \\ \frac{\partial^2 g^r}{\partial \sigma^2} \\ \vdots \\ \frac{\partial^2 g^{ny}}{\partial \sigma^2} \end{bmatrix},$$

[5] Schmitt-Grohe & Uribe (2004) show that $g_{\sigma x} = h_{\sigma x} = 0$.

where

$$
\underset{nx\times nx}{h_{x,x_j}} = \begin{bmatrix} \frac{\partial^2 h^1}{\partial x_{1,t}\partial x_{j,t}} & \cdots & \frac{\partial^2 h^1}{\partial x_{i,t}\partial x_{j,t}} & \cdots & \frac{\partial^2 h^1}{\partial x_{nx,t}\partial x_{j,t}} \\ \vdots & & \vdots & & \vdots \\ \frac{\partial^2 h^q}{\partial x_{1,t}\partial x_{j,t}} & \cdots & \frac{\partial^2 h^q}{\partial x_{i,t}\partial x_{j,t}} & \cdots & \frac{\partial^2 h^q}{\partial x_{nx,t}\partial x_{j,t}} \\ \vdots & & \vdots & & \vdots \\ \frac{\partial^2 h^{nx}}{\partial x_{1,t}\partial x_{j,t}} & \cdots & \frac{\partial^2 h^{nx}}{\partial x_{i,t}\partial x_{j,t}} & \cdots & \frac{\partial^2 h^{nx}}{\partial x_{nx,t}\partial x_{j,t}} \end{bmatrix},
$$

$$
\underset{ny\times nx}{g_{x,x_j}} = \begin{bmatrix} \frac{\partial^2 g^1}{\partial x_{1,t}\partial x_{j,t}} & \cdots & \frac{\partial^2 g^1}{\partial x_{i,t}\partial x_{j,t}} & \cdots & \frac{\partial^2 g^1}{\partial x_{nx,t}\partial x_{j,t}} \\ \vdots & & \vdots & & \vdots \\ \frac{\partial^2 g^r}{\partial x_{1,t}\partial x_{j,t}} & \cdots & \frac{\partial^2 g^r}{\partial x_{i,t}\partial x_{j,t}} & \cdots & \frac{\partial^2 g^r}{\partial x_{nx,t}\partial x_{j,t}} \\ \vdots & & \vdots & & \vdots \\ \frac{\partial^2 g^{nx}}{\partial x_{1,t}\partial x_{j,t}} & \cdots & \frac{\partial^2 g^{nx}}{\partial x_{i,t}\partial x_{j,t}} & \cdots & \frac{\partial^2 g^{nx}}{\partial x_{nx,t}\partial x_{j,t}} \end{bmatrix}.
$$

The matrices g_{xx} and h_{xx} are the coefficient matrices for the quadratic terms, while $g_{\sigma\sigma}$ and $h_{\sigma\sigma}$ are the intercept corrections due to the presence of risk.

Similarly I obtain a third-order approximation to the policy functions (equations (2) and (3))[6]

$$
\begin{aligned}
x_{t+1} &= h_x x_t + \tfrac{1}{2}\sigma^2 h_{\sigma\sigma} + \tfrac{1}{2} h_{xx}\left(x_t \otimes x_t\right) + \tfrac{1}{6}\sigma^2 h_{\sigma\sigma\sigma} + \cdots \\
&\qquad \cdots + \tfrac{3}{6}\sigma^2 h_{\sigma\sigma x} x_t + \tfrac{1}{6} h_{xxx}\left(x_t \otimes x_t \otimes x_t\right) + \sigma\varepsilon_{t+1}, \qquad (6) \\
y_t &= g_x x_t + \tfrac{1}{2}\sigma^2 g_{\sigma\sigma} + \tfrac{1}{2} g_{xx}\left(x_t \otimes x_t\right) + \tfrac{1}{6}\sigma^2 g_{\sigma\sigma\sigma} + \tfrac{3}{6}\sigma^2 g_{\sigma\sigma x} x_t + \tfrac{1}{6} g_{xxx}\left(x_t \otimes x_t \otimes x_t\right). \qquad (7)
\end{aligned}
$$

The same first and second-order terms that appeared in equations (4) and (5) also appear in the third-order solution, but now there are some additional third-order terms: g_{xxx}, h_{xxx}, $g_{\sigma\sigma x}$, $h_{\sigma\sigma x}$, $g_{\sigma\sigma\sigma}$ and $h_{\sigma\sigma\sigma}$, these are defined as follows:

$$
\underset{nx\times nx^3}{h_{xxx}} = \left[\; h_{x,x,x_1}, \;\; \cdots, \;\; h_{x,x,x_k}, \;\; \cdots, \;\; h_{x,x,x_{nx}} \;\right], \qquad \underset{ny\times nx^3}{g_{xxx}} = \left[\; g_{x,x,x_1}, \;\; \cdots, \;\; g_{x,x,x_k}, \;\; \cdots, \;\; g^{nx}_{x,x,x_{nx}} \;\right],
$$

$$
\underset{nx\times nx}{h_{\sigma\sigma x}} = \begin{bmatrix} \frac{\partial^3 h^1}{\partial\sigma^2\partial x_{1,t}} & \cdots & \frac{\partial^3 h^1}{\partial\sigma^2\partial x_{i,t}} & \cdots & \frac{\partial^3 h^1}{\partial\sigma^2\partial x_{nx,t}} \\ \vdots & & \vdots & & \vdots \\ \frac{\partial^3 h^j}{\partial\sigma^2\partial x_{1,t}} & \cdots & \frac{\partial^3 h^j}{\partial\sigma^2\partial x_{i,t}} & \cdots & \frac{\partial^3 h^j}{\partial\sigma^2\partial x_{nx,t}} \\ \vdots & & \vdots & & \vdots \\ \frac{\partial^3 h^{nx}}{\partial\sigma^2\partial x_{1,t}} & \cdots & \frac{\partial^3 h^{nx}}{\partial\sigma^2\partial x_{i,t}} & \cdots & \frac{\partial^3 h^{nx}}{\partial\sigma^2\partial x_{nx,t}} \end{bmatrix}, \quad \underset{ny\times nx}{g_{\sigma\sigma x}} = \begin{bmatrix} \frac{\partial^3 g^1}{\partial\sigma^2\partial x_{1,t}} & \cdots & \frac{\partial^3 g^1}{\partial\sigma^2\partial x_{i,t}} & \cdots & \frac{\partial^3 g^1}{\partial\sigma^2\partial x_{nx,t}} \\ \vdots & & \vdots & & \vdots \\ \frac{\partial^3 g^j}{\partial\sigma^2\partial x_{1,t}} & \cdots & \frac{\partial^3 g^j}{\partial\sigma^2\partial x_{i,t}} & \cdots & \frac{\partial^3 g^j}{\partial\sigma^2\partial x_{nx,t}} \\ \vdots & & \vdots & & \vdots \\ \frac{\partial^3 g^{nx}}{\partial\sigma^2\partial x_{1,t}} & \cdots & \frac{\partial^3 g^{nx}}{\partial\sigma^2\partial x_{i,t}} & \cdots & \frac{\partial^3 g^{nx}}{\partial\sigma^2\partial x_{nx,t}} \end{bmatrix},
$$

[6] Andreasen (2011) shows that $g_{xx\sigma} = h_{xx\sigma} = 0$.

$$\underset{nx\times 1}{h_{\sigma\sigma\sigma}} = \begin{bmatrix} \frac{\partial^3 h^1}{\partial \sigma^3} \\ \vdots \\ \frac{\partial^3 h^j}{\partial \sigma^3} \\ \vdots \\ \frac{\partial^3 h^{nx}}{\partial \sigma^3} \end{bmatrix}, \quad \underset{ny\times 1}{g_{\sigma\sigma\sigma}} = \begin{bmatrix} \frac{\partial^3 g^1}{\partial \sigma^3} \\ \vdots \\ \frac{\partial^3 g^j}{\partial \sigma^3} \\ \vdots \\ \frac{\partial^3 g^{ny}}{\partial \sigma^3} \end{bmatrix},$$

where

$$\underset{nx\times nx^3}{h_{x,x,x_k}} = \begin{bmatrix} h_{x,x_1,x_k} & \cdots & h_{x,x_j,x_k} & \cdots & h_{x,x_{nx},x_k} \end{bmatrix}, \quad \underset{ny\times nx^3}{g_{x,x,x_k}} = \begin{bmatrix} g_{x,x_1,x_k} & \cdots & g_{x,x_j,x_k} & \cdots & g_{x,x_{nx},x_k} \end{bmatrix},$$

and

$$\underset{nx\times nx}{h_{x,x_j,x_k}} = \begin{bmatrix} \frac{\partial^3 h^1}{\partial x_{1,t}\partial x_{j,t}\partial x_{k,t}} & \cdots & \frac{\partial^3 h^1}{\partial x_{i,t}\partial x_{j,t}\partial x_{k,t}} & \cdots & \frac{\partial^3 h^1}{\partial x_{nx,t}\partial x_{j,t}\partial x_{k,t}} \\ \vdots & & \vdots & & \vdots \\ \frac{\partial^3 h^q}{\partial x_{1,t}\partial x_{j,t}\partial x_{k,t}} & \cdots & \frac{\partial^3 h^q}{\partial x_{i,t}\partial x_{j,t}\partial x_{k,t}} & \cdots & \frac{\partial^3 h^q}{\partial x_{nx,t}\partial x_{k,t}\partial x_{l,t}} \\ \vdots & & \vdots & & \vdots \\ \frac{\partial^3 h^{nx}}{\partial x_{1,t}\partial x_{j,t}\partial x_{k,t}} & \cdots & \frac{\partial^3 h^{nx}}{\partial x_{i,t}\partial x_{j,t}\partial x_{k,t}} & \cdots & \frac{\partial^3 h^{nx}}{\partial x_{nx,t}\partial x_{j,t}\partial x_{k,t}} \end{bmatrix},$$

$$\underset{ny\times nx}{g_{x,x_j,x_k}} = \begin{bmatrix} \frac{\partial^3 g^1}{\partial x_{1,t}\partial x_{j,t}\partial x_{k,t}} & \cdots & \frac{\partial^3 g^1}{\partial x_{i,t}\partial x_{j,t}\partial x_{k,t}} & \cdots & \frac{\partial^3 g^1}{\partial x_{nx,t}\partial x_{j,t}\partial x_{k,t}} \\ \vdots & & \vdots & & \vdots \\ \frac{\partial^3 g^r}{\partial x_{1,t}\partial x_{j,t}\partial x_{k,t}} & \cdots & \frac{\partial^3 g^r}{\partial x_{i,t}\partial x_{j,t}\partial x_{k,t}} & \cdots & \frac{\partial^3 g^r}{\partial x_{nx,t}\partial x_{j,t}\partial x_{k,t}} \\ \vdots & & \vdots & & \vdots \\ \frac{\partial^3 g^{nx}}{\partial x_{1,t}\partial x_{j,t}\partial x_{k,t}} & \cdots & \frac{\partial^3 g^{nx}}{\partial x_{i,t}\partial x_{j,t}\partial x_{k,t}} & \cdots & \frac{\partial^3 g^{nx}}{\partial x_{nx,t}\partial x_{j,t}\partial x_{k,t}} \end{bmatrix}.$$

The matrices g_{xxx} and h_{xxx} are the coefficient matrices on the cubic terms, while $g_{\sigma\sigma x}$ and $h_{\sigma\sigma x}$ capture time varying risk. The terms $g_{\sigma\sigma\sigma}$ and $h_{\sigma\sigma\sigma}$ are intercept corrections that are non-zero if the shocks come from a skewed distribution.

Finding the coefficient matrices in a second or a third-order Taylor series approximation around the non-stochastic steady state is complicated by the fact that the policy functions are unknown. However the implicit function theorem can be used to find chain rules involving the unknown derivatives of the policy function. Solutions to lower orders of approximation are required to solve higher orders of approximation; for example the first-order approximation is required to solve a second-order approximation, and both the first and second-order approximations are required to solve a third-order approximation. The steps for finding a second and a third-order approximation are outlined below:

i) First the policy functions in (2) and (3) are substituted into equation (1) to get

$$\mathrm{E}_t f\left(h(x_t,\sigma) + \sigma\varepsilon_{t+1},\ g(h(x_t,\sigma) + \sigma\varepsilon_{t+1},\sigma),\ x_t,\ g(x_t,\sigma)\right) = 0. \tag{8}$$

ii) To find a first-order approximation, differentiate equation (8) with respect to all the elements in x_t. The resulting chain rule is a quadratic in terms of the unknown coefficient matrices g_x and h_x so a solution must be found using a method like the one described in Klein (2000). This requires the gradient matrix to the function f, which can be easily found.

iii) To find the second-order approximation, differentiate equation (8) twice with respect to all combinations of the elements in x_t. This results in a second-order chain rule. The gradient matrix and the Hessian of the function f can easily be found, and the solution to the first order approximation was found in step ii), so all that remains are the unknown coefficients g_{xx} and h_{xx}. These can be found as the solution to a system of linear equations. Similar steps can be used to find $g_{\sigma\sigma}$ and $h_{\sigma\sigma}$.

iv) To find the third-order approximation to the policy functions, differentiate equation (8) three times with respect to all combinations of the elements in x_t. The resulting chain rule is linear in the unknown coefficients g_{xxx} and h_{xxx}. The gradient matrix, the Hessian and the matrix of third derivatives for the function f are easily found, and the gradient matrix and the Hessian of the policy functions were found in steps ii) and iii). The third-order terms can be found as the solution to a system of linear equations. A similar set of steps can be taken to find $g_{\sigma\sigma x}$, $h_{\sigma\sigma x}$, $g_{\sigma\sigma\sigma}$ and $h_{\sigma\sigma\sigma}$.

Typically the chain rules are represented using tensor notation (see Schmitt-Grohe & Uribe (2004), Ruge-Murcia (2010) and Andreasen (2011) for examples). As discussed by Binning (2013) there are drawbacks to using tensor notation, in particular tensor notation is difficult to write down, difficult to code up and slow to implement when using Matlab. The method for solving the system of linear equations also plays a key role in the efficiency of the solution algorithm. Rearranging the chain rules into a system of generalised Sylvester equations is more efficient than using standard matrix algebra. In particular Kamenik (2005) presents a representation of the generalised Sylvester equations with a convenient Kronecker product structure and an extremely efficient solution algorithm that exploits this structure. However Kamenik (2005) uses tensor notation to find the matrices for his algorithm and tensor notation is not well suited to Matlab. In the next section I present second and third-order matrix chain rules that are consistent with Kamenik's generalised Sylvester equation representation. The matrix chain rules are easier to write down and easier to code than tensor notation, and faster to implement in Matlab.

3. A second and a third-order matrix chain rule

As discussed in the introduction, if a problem has a natural Sylvester equation structure, exploiting this structure when solving the system of equations can result in significant performance improvements, both in speed and memory usage. Two particular algorithms that are extremely efficient at solving generalised Sylvester equations are Kamenik (2005) and Martin & Van Loan (2006), especially when the problem has a certain Kronecker product structure. Kamenik (2005) uses higher order chain rules written in tensor notation to solve higher order approximations of DSGE models, but he is missing a theory of matrix chain rules consistent with his Sylvester equation structure. Existing matrix chain rules by Magnus & Neudecker (1999) (see Gomme & Klein, 2011) and Binning (2013) are not consistent with the Kamenik form of the problem, nor are they unique. In this section I present a second and third-order matrix chain rule that with a small amount of matrix algebra can be rewritten into the form of generalised Sylvester equations that are consistent with both the Kamenik, and Martin and Van Loan algorithms.

I begin with the second-order chain rule. Let $\mathbf{x}$ be a vector of variables so that

$$\mathbf{x} = [\mathsf{x}_1, \cdots, \mathsf{x}_i, \cdots, \mathsf{x}_n]',$$

for $i = 1, \cdots, n$. Let f be an m-ary function of g, which in turn is an n-ary function of $\mathbf{x}$ so that

$$\mathsf{y} = \mathsf{f}(\mathsf{g}(\mathbf{x})), \tag{9}$$
$$\mathsf{y} = \mathsf{f}(\mathsf{g}^1(\mathbf{x}), \cdots, \mathsf{g}^a(\mathbf{x}), \cdots, \mathsf{g}^m(\mathbf{x})),$$

for $a = 1, \cdots, m$. By Faà di Bruno's formula (see Johnson, 2002) the second derivative of y with respect to x_i and x_j is given by

$$\frac{\partial^2 \mathsf{y}}{\partial \mathsf{x}_i \partial \mathsf{x}_j} = \sum_{a=1}^{m} \sum_{b=1}^{m} \frac{\partial^2 \mathsf{f}}{\partial \mathsf{g}^a \partial \mathsf{g}^b} \left(\frac{\partial \mathsf{g}^a}{\partial \mathsf{x}_i}\right) \left(\frac{\partial \mathsf{g}^b}{\partial \mathsf{x}_j}\right) + \sum_{a=1}^{m} \frac{\partial \mathsf{f}}{\partial \mathsf{g}^a} \left(\frac{\partial^2 \mathsf{g}^a}{\partial \mathsf{x}_i \partial \mathsf{x}_j}\right). \tag{10}$$

This can be rewritten more compactly as

$$\mathsf{y}_{i,j} = \sum_{a=1}^{m} \sum_{b=1}^{m} \mathsf{f}_{a,b} \mathsf{g}_i^a \mathsf{g}_j^b + \sum_{a=1}^{m} \mathsf{f}_a \mathsf{g}_{i,j}^a, \tag{11}$$

where $\mathsf{y}_{i,j} = \frac{\partial^2 \mathsf{y}}{\partial \mathsf{x}_i \partial \mathsf{x}_j}$, $\mathsf{f}_{a,b} = \frac{\partial^2 \mathsf{f}}{\partial \mathsf{g}^a \partial \mathsf{g}^b}$, $\mathsf{g}_i^a = \frac{\partial \mathsf{g}^a}{\partial \mathsf{x}_i}$, $\mathsf{g}_j^b = \frac{\partial \mathsf{g}^b}{\partial \mathsf{x}_j}$, $\mathsf{f}_a = \frac{\partial \mathsf{f}}{\partial \mathsf{g}_a}$ and $\mathsf{g}_{i,j}^a = \frac{\partial^2 \mathsf{g}^a}{\partial \mathsf{x}_i \partial \mathsf{x}_j}$. The derivative of equation (9) with respect to all possible combinations of x_i and x_j can be written in matrix form (this is a Hessian matrix of sorts). This matrix form is a matrix representation of the second-order chain rule. To write equation (11) in matrix form for all possible combinations of x_i and x_j, I define a matrix Y with all possible second derivatives of y such that

$$\underset{1\times n^2}{\mathsf{Y}} = \left[\ \underset{1\times n}{\tilde{\mathsf{Y}}_1}, \quad \cdots, \quad \underset{1\times n}{\tilde{\mathsf{Y}}_j}, \quad \cdots, \quad \underset{1\times n}{\tilde{\mathsf{Y}}_n}\ \right],$$

where

$$\tilde{\mathsf{Y}}_j = [\mathsf{y}_{1,j}, \cdots, \mathsf{y}_{i,j}, \cdots, \mathsf{y}_{n,j}],$$

and the element in the 1st row and the $i + n(j-1)$th column of Y is given by

$$\tilde{\mathsf{y}}_{1,i+n(j-1)} = \mathsf{y}_{i,j}.$$

Indexing the rows and columns in terms of the derivatives will be useful when it comes to proving the matrix chain rule. In the second-order matrix chain rule of Magnus & Neudecker (1999), the matrix Y is $n \times n$. In order for the matrix chain rule to be consistent with Kamenik's algorithm I require Y to be $1 \times n^2$. The gradient vector for the function f is given by D

$$\underset{1\times m}{\mathsf{D}} = [\mathsf{f}_1, \cdots, \mathsf{f}_a, \cdots, \mathsf{f}_m],$$

where the element in the 1st row and the ath column of D is given by

$$\mathsf{d}_{1,a} = \mathsf{f}_a.$$

I form a matrix H of the second derivatives of the f function

$$\underset{1\times m^2}{\mathsf{H}} = \left[\underset{1\times m}{\tilde{\mathsf{H}}_1}, \quad \cdots, \quad \underset{1\times m}{\tilde{\mathsf{H}}_a}, \quad \cdots, \quad \underset{1\times m}{\tilde{\mathsf{H}}_m} \right],$$

where

$$\tilde{\mathsf{H}}_a = [\mathsf{f}_{a,1}, \cdots, \mathsf{f}_{a,b}, \cdots, \mathsf{f}_{a,m}],$$

and the element in the 1st row and the $b + m(a-1)$th column of H is given by

$$\mathsf{h}_{1,b+m(a-1)} = \mathsf{f}_{a,b}.$$

Because H is a matrix of second derivatives, it can be thought of as a type of Hessian matrix. Conventional Hessians are square matrices, while this is the transpose of a vectorised Hessian. The gradient matrix for the g function is denoted by M

$$\underset{m\times n}{\mathsf{M}} = \begin{bmatrix} \mathsf{g}_1^1 & \cdots & \mathsf{g}_i^1 & \cdots & \mathsf{g}_n^1 \\ \vdots & & \vdots & & \vdots \\ \mathsf{g}_1^a & \cdots & \mathsf{g}_i^a & \cdots & \mathsf{g}_n^a \\ \vdots & & \vdots & & \vdots \\ \mathsf{g}_1^m & \cdots & \mathsf{g}_i^m & \cdots & \mathsf{g}_n^m \end{bmatrix},$$

where

$$\mathsf{m}_{a,i} = \mathsf{g}_i^a,$$

with $\mathsf{m}_{a,i}$ the element in the ath row and the ith column of M. Finally I define the matrix N, the Hessian of the function g

$$\underset{m\times n^2}{\mathsf{N}} = \begin{bmatrix} \mathsf{g}_{1,1}^1 & \cdots & \mathsf{g}_{1,i}^1 & \cdots & \mathsf{g}_{j,1}^1 & \cdots & \mathsf{g}_{j,i}^1 & \cdots & \mathsf{g}_{j,n}^1 & \cdots & \mathsf{g}_{n,1}^1 & \cdots & \mathsf{g}_{n,i}^1 & \cdots & \mathsf{g}_{n,n}^1 \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ \mathsf{g}_{1,1}^a & \cdots & \mathsf{g}_{1,i}^a & \cdots & \mathsf{g}_{j,1}^a & \cdots & \mathsf{g}_{j,i}^a & \cdots & \mathsf{g}_{j,n}^a & \cdots & \mathsf{g}_{n,1}^a & \cdots & \mathsf{g}_{n,i}^a & \cdots & \mathsf{g}_{n,n}^a \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ \mathsf{g}_{1,1}^m & \cdots & \mathsf{g}_{1,i}^m & \cdots & \mathsf{g}_{j,1}^m & \cdots & \mathsf{g}_{j,i}^m & \cdots & \mathsf{g}_{j,n}^m & \cdots & \mathsf{g}_{n,1}^m & \cdots & \mathsf{g}_{n,i}^m & \cdots & \mathsf{g}_{n,n}^m \end{bmatrix},$$

where the element in the ath row and the $i + n(j-1)$th column of N is given by

$$\mathsf{n}_{a,i+n(j-1)} = \mathsf{g}_{j,i}^a.$$

Combining these matrices, I can now write down my representation for the second-order matrix chain rule

Theorem 1. *For* Y,H,M,D *and* N *defined previously and* $\mathsf{y} = \mathsf{f}(\mathsf{g}(\mathbf{x}))$,

$$\mathsf{Y} = \mathsf{H}(\mathsf{M} \otimes \mathsf{M}) + \mathsf{D}\mathsf{N}$$

is a valid representation of a second-order matrix chain rule.

Proof See Appendix B. ■

I follow a similar pattern when defining a third-order matrix chain rule consistent with a recursive generalised Sylvester equation solution. Using Faà di Bruno's formula, the third derivative of equation (9) with respect to x_i, x_j and x_k is given by

$$\begin{aligned}
\frac{\partial^3 \mathsf{y}}{\partial \mathsf{x}_i \partial \mathsf{x}_j \partial \mathsf{x}_k} = & \sum_{a=1}^{m} \sum_{b=1}^{m} \sum_{c=1}^{m} \frac{\partial \mathsf{f}}{\partial \mathsf{g}^a \partial \mathsf{g}^b \partial \mathsf{g}^c} \left(\frac{\partial \mathsf{g}^a}{\partial \mathsf{x}_i} \right) \left(\frac{\partial \mathsf{g}^b}{\partial \mathsf{x}_j} \right) \left(\frac{\partial \mathsf{g}^c}{\partial \mathsf{x}_k} \right) + \cdots \\
& \cdots + \sum_{a=1}^{m} \sum_{b=1}^{m} \frac{\partial^2 \mathsf{f}}{\partial \mathsf{g}^a \partial \mathsf{g}^b} \left(\frac{\partial \mathsf{g}^a}{\partial \mathsf{x}_i} \right) \left(\frac{\partial \mathsf{g}^b}{\partial \mathsf{x}_j \partial \mathsf{x}_k} \right) + \cdots \\
& \cdots + \sum_{a=1}^{m} \sum_{b=1}^{m} \frac{\partial^2 \mathsf{f}}{\partial \mathsf{g}^a \partial \mathsf{g}^b} \left(\frac{\partial \mathsf{g}^a}{\partial \mathsf{x}_j} \right) \left(\frac{\partial \mathsf{g}^b}{\partial \mathsf{x}_i \partial \mathsf{x}_k} \right) + \cdots \\
& \cdots + \sum_{a=1}^{m} \sum_{b=1}^{m} \frac{\partial^2 \mathsf{f}}{\partial \mathsf{g}^a \partial \mathsf{g}^b} \left(\frac{\partial \mathsf{g}^a}{\partial \mathsf{x}_k} \right) \left(\frac{\partial \mathsf{g}^b}{\partial \mathsf{x}_i \partial \mathsf{x}_j} \right) + \cdots \\
& \cdots + \sum_{a=1}^{m} \frac{\partial \mathsf{f}}{\partial \mathsf{g}^a} \left(\frac{\partial^3 \mathsf{g}^a}{\partial \mathsf{x}_i \partial \mathsf{x}_j \partial \mathsf{x}_k} \right).
\end{aligned}$$

Again the derivative of equation (9) with respect to all combinations of x_i, x_j and x_k can be written in matrix form. This will be a third-order matrix chain rule. Before presenting the third-order matrix chain rule consistent with a recursive generalised Sylvester equation form, I define some additional matrices required for the chain rule. I begin by defining Z, the matrix of third derivatives of y

$$\underset{1\times n^3}{\mathsf{Z}} = \left[\underset{1\times n^2}{\hat{\mathsf{Z}}_1}, \cdots, \underset{1\times n^2}{\hat{\mathsf{Z}}_k}, \cdots, \underset{1\times n^2}{\hat{\mathsf{Z}}_n} \right],$$

where

$$\hat{\mathsf{Z}}_k = \left[\underset{1\times n}{\tilde{\mathsf{Z}}_{1,k}}, \cdots, \underset{1\times n}{\tilde{\mathsf{Z}}_{j,k}}, \cdots, \underset{1\times n}{\tilde{\mathsf{Z}}_{n,k}} \right], \quad \text{and} \quad \tilde{\mathsf{Z}}_{j,k} = \left[\mathsf{y}_{1,j,k}, \cdots, \mathsf{y}_{i,j,k}, \cdots, \mathsf{y}_{n,j,k} \right],$$

and the element in the 1st row and the $i + n(j-1) + n^2(k-1)$th column of Z is given by

$$\mathsf{z}_{1,i+n(j-1)+n^2(k-1)} = \mathsf{y}_{i,j,k}.$$

This differs from the representation in Binning (2013). In that paper the matrix Z is $n^2 \times n$, in this paper Z is $1 \times n^3$ which is consistent with Kamenik's representation. I let T represent the matrix of third derivatives of the function f:

$$\underset{1\times m^3}{\mathsf{T}} = \left[\underset{1\times m^2}{\hat{\mathsf{T}}_1}, \cdots, \underset{1\times m^2}{\hat{\mathsf{T}}_c}, \cdots \underset{1\times m^2}{\hat{\mathsf{T}}_m}\right],$$

where

$$\hat{\mathsf{T}}_c = \left[\underset{m\times 1}{\tilde{\mathsf{T}}_{1,c}}, \cdots, \underset{m\times 1}{\tilde{\mathsf{T}}_{b,c}}, \cdots, \underset{m\times 1}{\tilde{\mathsf{T}}_{m,c}}\right], \quad \text{and} \quad \tilde{\mathsf{T}}_{b,c} = \left[\mathsf{f}_{1,b,c}, \cdots, \mathsf{f}_{a,b,c}, \cdots, \mathsf{f}_{m,b,c}\right].$$

The element in the 1st row and the $a + m(b-1) + m^2(c-1)$th column of T is given by

$$\mathsf{t}_{1,a+m(b-1)+m^2(c-1)} = \mathsf{f}_{a,b,c}.$$

I let N^* be a variation on the Hessian N so that

$$\underset{m.n\times n^3}{\mathsf{N}^*} = \left[\underset{n\times n}{I} \otimes \mathsf{N}, \cdots, \underset{n\times n}{I} \otimes \mathsf{N}, \cdots, \underset{n\times n}{I} \otimes \mathsf{N}\right].$$

and the element in the $k + n(a-1)$th row and the $k + n(i-1) + n^2(j-1)$th column of N^* is given by

$$\mathsf{n}^*_{k+n(a-1),k+n(i-1)+n^2(j-1)} = \mathsf{g}^a_{j,i}.$$

The matrix K, is the matrix of third derivatives of the g function

$$\underset{m\times n^3}{\mathsf{K}} = \left[\underset{m\times n^2}{\hat{\mathsf{K}}_1}, \cdots, \underset{m\times n^2}{\hat{\mathsf{K}}_k}, \cdots, \underset{m\times n^2}{\hat{\mathsf{K}}_n}\right],$$

where

$$\hat{\mathsf{K}}_k = \left[\ \underset{m\times n}{\tilde{\mathsf{K}}_{1,k}} \cdots \underset{m\times n}{\tilde{\mathsf{K}}_{j,k}} \cdots \underset{m\times n}{\tilde{\mathsf{K}}_{n,k}}\ \right], \quad \text{and} \quad \tilde{\mathsf{K}}_{j,k} = \begin{bmatrix} \mathsf{g}^1_{1,j,k} & \cdots & \mathsf{g}^1_{i,j,k} & \cdots & \mathsf{g}^1_{n,j,k} \\ \vdots & & \vdots & & \vdots \\ \mathsf{g}^a_{1,j,k} & \cdots & \mathsf{g}^a_{i,j,k} & \cdots & \mathsf{g}^a_{n,j,k} \\ \vdots & & \vdots & & \vdots \\ \mathsf{g}^m_{1,j,k} & \cdots & \mathsf{g}^m_{i,j,k} & \cdots & \mathsf{g}^m_{n,j,k} \end{bmatrix}.$$

The element in the ath row and the $i + n(j-1) + n^2(k-1)$th column of K is given by

$$\mathsf{k}_{a,i+n(j-1)+n^2(k-1)} = \mathsf{g}^a_{i,j,k}.$$

Using these matrices, I specify my third-order matrix chain rule as follows

Theorem 2. *For* Z,T,M,H,N,N^*,D *and* K *defined previously and* $\mathsf{y} = \mathsf{f}(\mathsf{g}(\mathbf{x}))$,

$$\mathsf{Z} = \mathsf{T}(\mathsf{M} \otimes \mathsf{M} \otimes \mathsf{M}) + \mathsf{H}(\mathsf{M} \otimes \mathsf{N}) + \mathsf{H}(\mathsf{N} \otimes \mathsf{M}) + \mathsf{H}\left(\mathsf{M} \otimes \underset{m\times m}{I}\right)\mathsf{N}^* + \mathsf{D}\mathsf{K}$$

is a valid representation of the third-order matrix chain rule.

Proof See Appendix C. ■

Theorems 1 and 2 are consistent with a recursive Sylvester equation solution, as will be discussed in the next section.

4. A recursive Sylvester equation solution

In the previous section I outlined a new representation for the second and third-order matrix chain rules. These chain rules are consistent with a recursive Sylvester equation solution method. Two such algorithms are Kamenik (2005) and Martin & Van Loan (2006). I give a brief description of each algorithm in this section.

4.1. Kamenik's algorithm

The recursive Sylvester equation solution described in Kamenik (2005) works on generalised Sylvester equations of the form

$$AX + BX\left(\otimes^k C\right) = D_k, \tag{12}$$

where A and B are known $n \times n$ matrices, C is a known $m \times m$ matrix, D_k is a known $n \times m^k$ matrix and X is an $n \times m^k$ matrix of unknowns. $\otimes^k$ is the kth order Kronecker product of the matrix C. As described in Kamenik (2005), the algorithm consists of three steps. The first step is preconditioning, a suitable linear transformation of the model must be found. This is done by premultiplying equation (12) by A^{-1} which gives

$$X + A^{-1}BX\left(\otimes^k C\right) = A^{-1}D_k. \tag{13}$$

Following Kamenik (2005) I find the real Schur decompositions $K = U(A^{-1}B)U'$ and $F = VCV'$ which allows equation (13) to be written as

$$Y + KY(\otimes^k F) = \bar{D}_k, \tag{14}$$

$$Y = UX(\otimes^k V'), \tag{15}$$

$$\bar{D}_k = UA^{-1}D_k(\otimes^k V'). \tag{16}$$

The second step is the recursive solution of equation (14). I vectorise equation (14) to obtain

$$\left(I + (\otimes^k F' \otimes K)\right)\operatorname{vec}(Y) = \operatorname{vec}(\bar{D}_k). \tag{17}$$

Equation (17) can be solved directly by calculating the Kronecker products and using elementary matrix algebra, but this is inefficient. Instead the Kamenik algorithm can be used to break this into smaller blocks to be solved individually, the results can be used to eliminate columns by updating the system through back substitution. I adopt the more compact notation of Kamenik by using the following definitions

$$F_{[k]} = \otimes^k F' \otimes K, \quad \text{where} \quad F_{[0]} = K.$$

The algorithm exploits the Kronecker product structure by solving the level k problem with the solutions to the same problem at level $k-1$. The matrices F and K will be quasi-triangular, and if the first eigenvalue of F is real (I denote this $r = F_{11}$) and y is the first part of Y chosen to be the same size as $F_{[k-1]}$, then y will be the solution to

$$(I + r \cdot F_{[k-1]})y = d. \tag{18}$$

If the first eigenvalue of F is complex, then the first two parts of Y and $\bar{D}_k$ are chosen. The first two parts of Y will be a solution to

$$\left(I + \begin{pmatrix} \alpha & \beta_1 \\ -\beta_2 & \alpha \end{pmatrix} \otimes F_{[k-1]}\right) \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}, \tag{19}$$

where α, β_1 and β_2 make up the first complex eigenvalue block.

The solution to equation (18) or (19) is then used to eliminate all non-zero elements below the first block (this is because F' is lower quasi-triangular). In the real case this is done as follows

$$d_j \longleftarrow d_j - F_{1j} \cdot (F_{[k-1]})y \qquad \text{for all} \quad j = 2, \cdots, m,$$

and in the complex case

$$d_j \longleftarrow d_j - F_{1j} \cdot (F_{[k-1]})y_1 - F_{2j} \cdot (F_{[k-1]})y_2 \qquad \text{for all} \quad j = 2, \cdots, m.$$

Once the elements have been eliminated and $\bar{D}_k$ has been updated, equation (18) or (19) can be used to find the next block of Y. If $k = 0$ the solution of equation (18) is straight forward, however the solution of equation (18) could depend on the solution of equation (19) which is more complicated. I refer the reader to Kamenik (2005) for a full description of how equations (18) and (19) are solved. To recover the results, the solution to equation (14) is multiplied by $X = U'Y(\otimes^k V)$.

4.2. Martin and Van Loan's algorithm

Martin & Van Loan (2006) take a similar approach to Kamenik (2005) to solve problems like equation (12). To get equation (12) into the correct form, it can be rewritten as

$$X + PX(\otimes^k C) = Z,$$

where $P = A^{-1}B$ and $Z = A^{-1}D_k$. Using the vec operator, I obtain

$$\left((\otimes^k S \otimes P) - \lambda I\right) x = z,$$

where $S = C'$, $x = \text{vec}(X)$, $z = \text{vec}(Z)$ and $\lambda = -1$. Martin & Van Loan (2006) refer to this as a shifted Kronecker product system.
Taking the real Schur decomposition of S and P gives

$$\left((\otimes^k R \otimes W) - \lambda I\right) y = q, \tag{20}$$

where $R = Q^{-1}SQ$, $W = U^{-1}PU$, $y = (\otimes^k Q \otimes U)x$ and $q = (\otimes^k Q \otimes U)z$. The matrices R and W are upper quasi-triangular and the matrices Q and U are unitary matrices. This system is then solved using a similar approach to Kamenik (2005), that is the solutions to the problem at level $k-1$ are used to solve the problem at level k. However, the Martin and Van Loan algorithm differs in the treatment of the complex eigenvalues (if any) in the upper quasi-triangular matrices. Kamenik (2005) uses real algebra to solve these blocks (see equation (19)) while Martin & Van Loan (2006) use the complex Schur decomposition to solve these blocks.

5. Second-order approximation

This section describes how to apply the second-order matrix chain rule from Theorem 1 to find a second-order approximation of a DSGE model, conditional on the solution to the first-order having been found. In particular I describe the steps required to get the matrix chain rule into the form of a system of generalised Sylvester equations that can be solved using a recursive generalised Sylvester equation solution algorithm.

5.1. Finding g_{xx} and h_{xx}

First I define the matrices required for the second-order matrix chain rule in Theorem 1, then I find the generalised Sylvester equation representation of the problem for the unknown coefficient matrices; g_{xx} and h_{xx}.

5.1.1. Matrix definitions

I begin by allowing x_t to represent the $nx \times 1$ vector of predetermined date t variables:

$$\underset{nx\times 1}{x_t} = [x_{1,t}, \cdots, x_{i,t}, \cdots, x_{nx,t}]' . \tag{21}$$

Likewise, the date t vector of non-predetermined variables, y_t is given by

$$\underset{ny\times 1}{y_t} = [y_{1,t}, \cdots, y_{i,t}, \cdots, y_{ny,t}]' . \tag{22}$$

Using definitions (21) and (22) I define the gradient vector of equation (1) to be

$$\underset{n\times 2n}{D} = \left[\begin{array}{cccc} \frac{\partial f}{\partial x'_{t+1}}, & \frac{\partial f}{\partial y'_{t+1}}, & \frac{\partial f}{\partial x'_t}, & \frac{\partial f}{\partial y'_t} \end{array} \right] . \tag{23}$$

It follows from equation (23) that the Hessian of equation (1) can be written as

$$\underset{n\times 4n^2}{H} = \left[\begin{array}{cccc} \frac{\partial D}{\partial x'_{t+1}}, & \frac{\partial D}{\partial y'_{t+1}}, & \frac{\partial D}{\partial x'_t}, & \frac{\partial D}{\partial y'_t} \end{array} \right] .$$

Note that this definition of the Hessian differs from standard definition of the Hessian and the definition used in Gomme & Klein (2011). However it is consistent with the Kamenik form of the problem.

The gradient matrix for the policy functions has the following form

$$\underset{2n \times nx}{M_x} = \begin{bmatrix} h_x \\ g_x h_x \\ \underset{nx \times nx}{I} \\ g_x \end{bmatrix}.$$

This is the same as the gradient matrix used in Gomme & Klein (2011) and Binning (2013).

5.1.2. Solution

Applying the second-order matrix chain rule (from Theorem 1) to equation (8) results in the following system of equations

$$H(M_x \otimes M_x) + D \begin{bmatrix} h_{xx} \\ g_x h_{xx} + g_{xx}\left(h_x \otimes h_x\right) \\ \underset{nx \times nx^2}{0} \\ g_{xx} \end{bmatrix} = \underset{n \times nx^2}{0}. \tag{24}$$

Note that Theorem 1 is applied to equation (8) directly and to $y_{t+1} = g(h(x_t, \sigma) + \sigma\varepsilon_{t+1}, \sigma)$ because it is also a composition function. To get equation (24) into the form of a generalised Sylvester equation I partition the matrix D so that

$$H(M_x \otimes M_x) + \left[\underset{n \times nx}{d_1}, \underset{n \times ny}{d_2}, \underset{n \times nx}{d_3}, \underset{n \times ny}{d_4}\right] \begin{bmatrix} h_{xx} \\ g_x h_{xx} + g_{xx}\left(h_x \otimes h_x\right) \\ \underset{nx \times nx^2}{0} \\ g_{xx} \end{bmatrix} = \underset{n \times nx^2}{0}. \tag{25}$$

From equation (25) I obtain the system of equations

$$H(M_x \otimes M_x) + [d_1 + d_2 g_x,\ d_4] \begin{bmatrix} h_{xx} \\ g_{xx} \end{bmatrix} + \left[\underset{n \times nx}{0},\ d_2\right] \begin{bmatrix} h_{xx} \\ g_{xx} \end{bmatrix} (h_x \otimes h_x) = \underset{n \times nx^2}{0}. \tag{26}$$

Equation (26) takes the form of a generalised Sylvester equation

$$AX + BX\left(C \otimes C\right) = D_2, \tag{27}$$

where

$$\begin{aligned} A &= \left[d_1 + d_2 g_x,\ d_4\right], \\ B &= \left[\underset{n \times nx}{0},\ d_2\right], \\ C &= h_x, \\ X &= \begin{bmatrix} h_{xx} \\ g_{xx} \end{bmatrix}, \\ D_2 &= -H(M_x \otimes M_x). \end{aligned}$$

Pre-multiplying equation (12) by A^{-1} gives

$$X + A^{-1}BX(C \otimes C) = A^{-1}D_2. \tag{28}$$

Equation (28) can be solved by using one of the recursive Sylvester equation algorithms described in this paper.

5.2. Finding $g_{\sigma\sigma}$ and $h_{\sigma\sigma}$

In this subsection, I define some additional matrices required for the solution before outlining the second-order matrix chain rule which can be solved to find the unknown coefficients; $g_{\sigma\sigma}$ and $h_{\sigma\sigma}$.

5.2.1. Matrix definitions

I allow N_σ to denote the first derivative of the policy functions with respect to the perturbation parameter σ

$$\underset{2n \times nx}{N_\sigma} = \begin{bmatrix} \underset{nx \times nx}{I} \\ g_x \\ \underset{n \times nx}{0} \end{bmatrix}.$$

This matrix is the same as the one defined in Gomme & Klein (2011) and Binning (2013).

I also define the variance-covariance matrix for the one step ahead prediction errors for the predetermined variables as

$$\underset{nx \times nx}{\Sigma} = \begin{bmatrix} \sigma_1^2 & \cdots & \sigma_{1,nx} \\ \vdots & & \vdots \\ \sigma_{nx,1} & \cdots & \sigma_{nx}^2 \end{bmatrix},$$

where $\sigma_i^2 = \mathrm{E}_t\left[u_{i,t}^2\right]$, $\sigma_{i,j} = \mathrm{E}_t\left[u_{i,t}u_{j,t}\right]$ and $u_{i,t}$ is the prediction error for the ith predetermined variable.

5.2.2. Solution

Using the second-order matrix chain rule (from Theorem 1) I write the second derivative of equation (8) with respect to σ as follows

$$\mathrm{trm}(H(N_\sigma \otimes N_\sigma\Sigma)) + D \begin{bmatrix} h_{\sigma\sigma} \\ g_x h_{\sigma\sigma} + g_{\sigma\sigma} + \mathrm{trm}\left(g_{xx}\left(\underset{nx \times nx}{I} \otimes \Sigma\right)\right) \\ \underset{nx \times 1}{0} \\ g_{\sigma\sigma} \end{bmatrix} = \underset{n \times 1}{0},$$

where trm is the matrix trace, which for a given matrix G is defined as follows

$$\mathrm{trm}\left(\underset{p \times k^2}{G}\right) = \sum_{i=1}^{k} G(:, i + (i-1)k).$$

This differs from the definition in Gomme & Klein (2011) and Binning (2013). The matrix trace appears in this problem as the consequence of taking the expectation of a matrix of random variables. See Appendix A for a further explanation.

To solve for the unknown coefficients $g_{\sigma\sigma}$ and $h_{\sigma\sigma}$ I partition the matrix D in equation (5.2.2) as follows

$$\operatorname{trm}(H(N_\sigma \otimes N_\sigma \Sigma)) + [d_1,\ d_2,\ d_3,\ d_4] \begin{bmatrix} h_{\sigma\sigma} \\ g_x h_{\sigma\sigma} + g_{\sigma\sigma} + \operatorname{trm}\left(g_{xx} \left(\underset{nx \times nx}{I} \otimes \Sigma \right) \right) \\ \underset{nx \times 1}{0} \\ g_{\sigma\sigma} \end{bmatrix} = \underset{n \times 1}{0}. \tag{29}$$

Equation (29) can be rearranged to obtain

$$\operatorname{trm}(H(N_\sigma \otimes N_\sigma \Sigma)) + d_2 \operatorname{trm}\left(g_{xx} \left(\underset{nx \times nx}{I} \otimes \Sigma \right) \right) + [d_1 + d_2 g_x,\ d_2 + d_4] \begin{bmatrix} h_{\sigma\sigma} \\ g_{\sigma\sigma} \end{bmatrix} = \underset{n \times 1}{0}. \tag{30}$$

Equation (30) is then easily written as

$$AX = B, \tag{31}$$

where

$$\begin{aligned} A &= [d_1 + d_2 g_x,\ d_2 + d_4], \\ X &= \begin{bmatrix} h_{\sigma\sigma} \\ g_{\sigma\sigma} \end{bmatrix}, \\ B &= -\operatorname{trm}(H(N_\sigma \otimes N_\sigma \Sigma)) - d_2 \operatorname{trm}\left(g_{xx} \left(\underset{nx \times nx}{I} \otimes \Sigma \right) \right), \end{aligned}$$

which can be solved using standard matrix algebra.

6. Third-order approximation

As mentioned previously, g_x and h_x are known, and g_{xx}, h_{xx}, $g_{\sigma\sigma}$ and $h_{\sigma\sigma}$ can be found using the algorithms described in the previous sections. The rest of this section outlines the application of the third-order matrix chain rule from Theorem 2 to find g_{xxx}, h_{xxx}, $g_{\sigma\sigma x}$, $h_{\sigma\sigma x}$, $g_{\sigma\sigma\sigma}$ and $h_{\sigma\sigma\sigma}$. The additional steps required to get the chain-rules in the form of a generalised Sylvester equation consistent with a recursive solution algorithm are also covered. Note that if the third moment of the shocks is equal to zero, then $g_{\sigma\sigma\sigma}$ and $h_{\sigma\sigma\sigma}$ will also be equal to zero.

6.1. Finding g_{xxx} and h_{xxx}

In this section I define some matrices required for the solution, then I describe the third-order matrix chain rule which can be written as a system of generalised Sylvester equations and solved to find the matrices, g_{xxx} and h_{xxx}.

6.1.1. Matrix definitions

The matrix of third derivatives for the function f in equation (1) is given by

$$\underset{n\times 8n^3}{T} = \left[\begin{array}{cccc} \frac{\partial H}{\partial x'_{t+1}}, & \frac{\partial H}{\partial y'_{t+1}}, & \frac{\partial H}{\partial x'_t}, & \frac{\partial H}{\partial y'_t} \end{array}\right].$$

The Hessian of the policy functions is given by

$$\underset{2n\times nx^2}{M_{xx}} = \left[\begin{array}{c} h_{xx} \\ g_{xx}\left(h_x \otimes h_x\right) + g_x h_{xx} \\ \underset{nx\times nx^2}{0} \\ g_{xx} \end{array}\right].$$

This can be partitioned according to the second derivative of each predetermined variable so that

$$M_{xx} = \left[\underset{2n\times nx}{M^1_{xx}}, \cdots, \underset{2n\times nx}{M^i_{xx}}, \cdots, \underset{2n\times nx}{M^{nx}_{xx}}\right]. \tag{32}$$

Using the partitions from equation (32), I define an alternative Hessian matrix for the policy functions

$$\underset{2n.nx\times nx^3}{M^\dagger_{xx}} = \left[\underset{nx\times nx}{I} \otimes M^1_{xx}, \cdots, \underset{nx\times nx}{I} \otimes M^i_{xx}, \cdots, \underset{nx\times nx}{I} \otimes M^{nx}_{xx}\right].$$

I also partition the second derivative of the policy function $h(\cdot)$ in a similar fashion to equation (32)

$$h_{xx} = \left[\underset{nx\times nx}{h^1_{xx}}, \cdots, \underset{nx\times nx}{h^i_{xx}}, \cdots, \underset{nx\times nx}{h^{nx}_{xx}}\right]. \tag{33}$$

Using the partitions from equation (33), I define an alternative Hessian for the policy function $h(\cdot)$

$$\underset{nx^2\times nx^3}{h^\dagger_{xx}} = \left[\underset{nx\times nx}{I} \otimes h^1_{xx}, \cdots, \underset{nx\times nx}{I} \otimes h^2_{xx}, \cdots, \underset{nx\times nx}{I} \otimes h^n x_{xx}\right].$$

6.1.2. Solution

Using the third-order matrix chain rule (from Theorem 2), I write the system of equations I need to solve to find g_{xxx} and h_{xxx} as

$$\begin{aligned} &T(M_x \otimes M_x \otimes M_x) + H(M_{xx} \otimes M_x) + \cdots \\ &\qquad \cdots + H(M_x \otimes M_{xx}) + H\left(M_x \otimes \underset{2n\times 2n}{I}\right) M^\dagger_{xx} + \cdots \\ &\cdots + D\left[\begin{array}{c} h_{xxx} \\ g_x h_{xxx} + g_{xx}(h_{xx} \otimes h_x) + g_{xx}(h_x \otimes h_{xx}) + \cdots \\ \cdots + g_{xx}\left(h_x \otimes \underset{nx\times nx}{I}\right) h^\dagger_{xx} + g_{xxx}\left(h_x \otimes h_x \otimes h_x\right) \\ \underset{nx\times nx^3}{0} \\ g_{xxx} \end{array}\right] = \underset{n\times nx^3}{0}. \end{aligned} \tag{34}$$

Note that Theorem 2 is applied to equation (8) directly and to $y_{t+1} = g(h(x_t, \sigma) + \sigma\varepsilon_{t+1}, \sigma)$ because it is also a composition function. To get equation (34) in the form of a generalised Sylvester equation I partition the matrix D, as follows

$$\begin{aligned} & T(M_x \otimes M_x \otimes M_x) + H(M_{xx} \otimes M_x) + \cdots \\ & \qquad \cdots + H(M_x \otimes M_{xx}) + H\left(M_x \otimes \underset{2n\times 2n}{I}\right) M_{xx}^{\dagger} + \cdots \\ & \cdots + [d_1, d_2, d_3, d_4] \begin{bmatrix} h_{xxx} \\ g_x h_{xxx} + g_{xx}(h_{xx} \otimes h_x) + g_{xx}(h_x \otimes h_{xx}) + \cdots \\ \cdots + g_{xx}\left(h_x \otimes \underset{nx\times nx}{I}\right) h_{xx}^{\dagger} + g_{xxx}\left(h_x \otimes h_x \otimes h_x\right) \\ \underset{nx\times nx^3}{0} \\ g_{xxx} \end{bmatrix} = \underset{n\times nx^3}{0}. \end{aligned} \tag{35}$$

I can rewrite equation (35) as

$$\begin{aligned} & T(M_x \otimes M_x \otimes M_x) + H(M_{xx} \otimes M_x) + H(M_x \otimes M_{xx}) + H\left(M_x \otimes \underset{2n\times 2n}{I}\right) M_{xx}^{\dagger} + \cdots \\ & \qquad \cdots + d_2\left[g_{xx}(h_x \otimes h_{xx}) + g_{xx}(h_{xx} \otimes h_x) + g_{xx}\left(h_x \otimes \underset{nx\times nx}{I}\right) h_{xx}^{\dagger}\right] + \cdots \\ & \qquad \cdots + [d_1 + d_2 g_x,\ d_4] \begin{bmatrix} h_{xxx} \\ g_{xxx} \end{bmatrix} + \left[\underset{n\times nx}{0},\ d_2\right] \begin{bmatrix} h_{xxx} \\ g_{xxx} \end{bmatrix} (h_x \otimes h_x \otimes h_x) = \underset{n\times nx^3}{0}, \end{aligned} \tag{36}$$

Equation (36) can then be written as the generalised Sylvester equation

$$AX + BX\left(C \otimes C \otimes C\right) = D_3, \tag{37}$$

where

$$\begin{aligned} A &= [d_1 + d_2 g_x,\ d_4], \\ B &= \left[\underset{n\times nx}{0},\ d_2\right], \\ C &= h_x, \\ X &= \begin{bmatrix} h_{xxx} \\ g_{xxx} \end{bmatrix}, \\ D_3 &= -T(M_x \otimes M_x \otimes M_x) - H(M_{xx} \otimes M_x) - H(M_x \otimes M_{xx}) - \cdots \\ & \quad \cdots - H\left(M_x \otimes \underset{2n\times 2n}{I}\right) M_{xx}^{\dagger} - d_2\left[g_{xx}(h_x \otimes h_{xx}) + g_{xx}(h_{xx} \otimes h_x) + g_{xx}\left(h_x \otimes \underset{nx\times nx}{I}\right) h_{xx}^{\dagger}\right]. \end{aligned}$$

Premultiplying equation (37) by A^{-1} gives

$$X + A^{-1}BX(C \otimes C \otimes C) = A^{-1}D_3. \tag{38}$$

Just as was done with the second-order approximation (section 5), equation (38) can be solved using either the algorithm of Kamenik (2005) or the algorithm of Martin & Van Loan (2006).

6.2. Finding $g_{\sigma\sigma x}$ and $h_{\sigma\sigma x}$

Next I find the time varying risk terms $g_{\sigma\sigma x}$ and $h_{\sigma\sigma x}$, but first I define some additional matrices required to write out the problem.

6.2.1. Matrix definitions

I begin by defining the third derivative of the policy functions with respect to x_t and σ

$$\underset{2n\times nx^2}{N_{\sigma x}} = \begin{bmatrix} \underset{nx\times nx^2}{0} \\ g_{xx}\left(h_x \otimes \underset{nx\times nx}{I}\right) \\ \underset{n\times nx^2}{0} \end{bmatrix},$$

I also define the Hessian of equation (8) with respect to σ as

$$\underset{2n\times 1}{P_{\sigma\sigma}} = \begin{bmatrix} h_{\sigma\sigma} \\ g_x h_{\sigma\sigma} + \operatorname{trm}\left(g_{xx}\left(\underset{nx\times nx}{I} \otimes \Sigma\right)\right) + g_{\sigma\sigma} \\ \underset{nx\times 1}{0} \\ g_{\sigma\sigma} \end{bmatrix}.$$

6.2.2. Solution

Using the third-order matrix chain rule (from Theorem 2) I can differentiate equation (8) with respect to σ (twice) and with respect to all elements in x_t to obtain

$$\begin{aligned}&\operatorname{trm}(T(M_x \otimes N_\sigma \otimes N_\sigma\Sigma)) + 2\times\operatorname{trm}(H(N_{\sigma x}\otimes N_\sigma\Sigma)) + H\left(M_x \otimes P_{\sigma\sigma}\right) + \cdots \\ &\cdots + D\begin{bmatrix} h_{\sigma\sigma x} \\ g_x h_{\sigma\sigma x} + \operatorname{trm}\left(g_{xxx}\left(h_x \otimes \underset{nx^2\times nx^2}{I}\right)\left(\underset{nx^2\times nx^2}{I}\otimes\Sigma\right)\right) + \cdots \\ \cdots + g_{xx}(h_x\otimes h_{\sigma\sigma}) + g_{\sigma\sigma x}h_x \\ \underset{nx\times nx}{0} \\ g_{\sigma\sigma x}\end{bmatrix} = \underset{n\times nx}{0}. \end{aligned} \tag{39}$$

To get equation (39) in the form of a generalised Sylvester equation I partition the the matrix D so that

$$\begin{aligned}&\operatorname{trm}(T(M_x \otimes N_\sigma \otimes N_\sigma\Sigma)) + 2\times\operatorname{trm}(H(N_{\sigma x}\otimes N_\sigma\Sigma)) + H\left(M_x \otimes P_{\sigma\sigma}\right) + \cdots \\ &\cdots + [d_1,\ d_2,\ d_3,\ d_4]\begin{bmatrix} h_{\sigma\sigma x} \\ g_x h_{\sigma\sigma x} + \operatorname{trm}\left(g_{xxx}\left(h_x \otimes \underset{nx^2\times nx^2}{I}\right)\left(\underset{nx^2\times nx^2}{I}\otimes\Sigma\right)\right) + \cdots \\ \cdots + g_{xx}(h_x\otimes h_{\sigma\sigma}) + g_{\sigma\sigma x}h_x \\ \underset{n\times nx}{0} \\ g_{\sigma\sigma x}\end{bmatrix} = \underset{n\times nx}{0}. \end{aligned} \tag{40}$$

I rearrange equation (40) to obtain

$$
\begin{aligned}
&\operatorname{trm}(T(M_x \otimes N_\sigma \otimes N_\sigma \Sigma)) + 2 \times \operatorname{trm}(H(N_{\sigma x} \otimes N_\sigma \Sigma)) + H\left(M_x \otimes P_{\sigma\sigma}\right) + \cdots \\
&\quad \cdots + d_2 \left[\operatorname{trm}\left(g_{xxx}\left(h_x \otimes \underset{nx^2 \times nx^2}{I}\right)\left(\underset{nx^2 \times nx^2}{I} \otimes \Sigma\right)\right) + g_{xx}(h_x \otimes h_{\sigma\sigma})\right] + \cdots \\
&\quad\quad \cdots + [d_1 + d_2 g_x,\ d_4] \begin{bmatrix} h_{\sigma\sigma x} \\ g_{\sigma\sigma x} \end{bmatrix} + \left[\underset{n \times nx}{0},\ d_2\right] \begin{bmatrix} h_{\sigma\sigma x} \\ g_{\sigma\sigma x} \end{bmatrix} h_x = \underset{n \times nx}{0}. \qquad (41)
\end{aligned}
$$

Equation (41) can be rewritten as the generalised Sylvester equation

$$
AX + BXC = D_1, \tag{42}
$$

where

$$
\begin{aligned}
A &= [d_1 + d_2 g_x,\ d_4], \\
B &= \left[\underset{n \times nx}{0},\ d_2\right], \\
C &= h_x, \\
D_1 &= -\operatorname{trm}(T(M_x \otimes N_\sigma \otimes N_\sigma \Sigma)) - 2 \times \operatorname{trm}(H(N_{\sigma x} \otimes N_\sigma \Sigma)) - \cdots \\
&\quad \cdots - H\left(M_x \otimes P_{\sigma\sigma}\right) - d_2 \left[\operatorname{trm}\left(g_{xxx}\left(h_x \otimes \underset{nx^2 \times nx^2}{I}\right)\left(\underset{nx^2 \times nx^2}{I} \otimes \Sigma\right)\right) + g_{xx}(h_x \otimes h_{\sigma\sigma})\right].
\end{aligned}
$$

Premultiplying equation (42) by A^{-1} gives

$$
X + A^{-1}BXC = A^{-1}D_1. \tag{43}
$$

As in the previous section, equation (43) can be solved using the recursive solution method of Kamenik (2005), or Martin & Van Loan (2006).

6.3. Finding $g_{\sigma\sigma\sigma}$ and $h_{\sigma\sigma\sigma}$

In this section I solve for $g_{\sigma\sigma\sigma}$ and $h_{\sigma\sigma\sigma}$. If the shocks are symmetrically distributed then $g_{\sigma\sigma\sigma}$ and $h_{\sigma\sigma\sigma}$ will be equal to zero.

6.3.1. Matrix definitions

I define some additional matrices in this subsection. The Hessian of the policy functions (with respect to σ) can be written as

$$
\underset{2n \times nx}{N_{\sigma\sigma}} = \begin{bmatrix} \underset{nx \times nx^2}{0} \\ g_{xx} \\ \underset{n \times nx^2}{0} \end{bmatrix}.
$$

I also define the matrix Γ to be the skewness-coskewness matrix

$$\underset{nx\times nx^2}{\Gamma} = \begin{bmatrix} \gamma_1 & \gamma_{1,1,2} & \cdots & \gamma_{1,nx,nx} \\ \vdots & & & \vdots \\ \gamma_{nx,1,1} & \cdots & \cdots & \gamma_{nx} \end{bmatrix}.$$

The skewness matrix contains the third moments of the prediction errors, where $\gamma_i = E_t\left[u_{i,t}^3\right]$, $\gamma_{i,j,k} = E_t\left[u_{i,t}u_{j,t}u_{k,t}\right]$, and $u_{i,t}$ is the prediction error for the ith predetermined variable. This follows from the definition of the variance-covariance matrix: $\Sigma = E_t\left[u_t \otimes u_t'\right]$, so that $\Gamma = E_t\left[u_t \otimes u_t' \otimes u_t'\right]$, where u_t is a vector of prediction errors. If all the shocks are symmetrically distributed, this matrix will have zeros for all of its entries.

6.3.2. Solution

Using the third-order matrix chain rule defined (from Theorem 2), I can write the third derivative of equation (8) with respect to σ as

$$\begin{aligned} &\operatorname{trm}(T(N_\sigma \otimes N_\sigma \otimes N_\sigma\Gamma)) + 3\times\operatorname{trm}\left(H\left(N_{\sigma\sigma}\otimes N_\sigma\Gamma\right)\right) + \cdots \\ &\cdots + D\begin{bmatrix} h_{\sigma\sigma\sigma} \\ g_x h_{\sigma\sigma\sigma} + g_{\sigma\sigma\sigma} + \operatorname{trm}\left(g_{xxx}\left(\underset{nx^2\times nx^2}{I}\otimes\Gamma\right)\right) \\ \underset{nx\times 1}{0} \\ g_{\sigma\sigma\sigma} \end{bmatrix} = \underset{n\times 1}{0}. \end{aligned} \tag{44}$$

To find the unknown coefficient matrices $g_{\sigma\sigma\sigma}$ and $h_{\sigma\sigma\sigma}$ I partition the matrix D so that

$$\begin{aligned} &\operatorname{trm}(T(N_\sigma \otimes N_\sigma \otimes N_\sigma\Gamma)) + 3\times\operatorname{trm}\left(H\left(N_{\sigma\sigma}\otimes N_\sigma\Gamma\right)\right) + \cdots \\ &\cdots + [d_1,\ d_2,\ d_3,\ d_4]\begin{bmatrix} h_{\sigma\sigma\sigma} \\ g_x h_{\sigma\sigma\sigma} + g_{\sigma\sigma\sigma} + \operatorname{trm}\left(g_{xxx}\left(\underset{nx^2\times nx^2}{I}\otimes\Gamma\right)\right) \\ \underset{nx\times 1}{0} \\ g_{\sigma\sigma\sigma} \end{bmatrix} = \underset{n\times 1}{0}. \end{aligned} \tag{45}$$

I rewrite equation (45) as

$$\begin{aligned} &\operatorname{trm}(T(N_\sigma \otimes N_\sigma \otimes N_\sigma\Gamma)) + 3\times\operatorname{trm}\left(H\left(N_{\sigma\sigma}\otimes N_\sigma\Gamma\right)\right) + \cdots \\ &\cdots + d_2\operatorname{trm}\left(g_{xxx}\left(\underset{nx^2\times nx^2}{I}\otimes\Gamma\right)\right) + [d_1 + d_2 g_x,\ d_2 + d_4]\begin{bmatrix} h_{\sigma\sigma\sigma} \\ g_{\sigma\sigma\sigma} \end{bmatrix} = \underset{n\times 1}{0}. \end{aligned} \tag{46}$$

Equation (46) can be rewritten as

$$AX = B, \tag{47}$$

where

$$
\begin{aligned}
&A = [d_1 + d_2 g_x,\ d_2 + d_4]\,,\\
&X = \begin{bmatrix} h_{\sigma\sigma\sigma} \\ g_{\sigma\sigma\sigma} \end{bmatrix},\\
&B = -\mathrm{trm}(T(N_\sigma \otimes N_\sigma \otimes N_\sigma \Gamma)) + \cdots\\
&\quad \cdots + 3 \times \mathrm{trm}\left(H\left(N_{\sigma\sigma} \otimes N_\sigma \Gamma\right)\right) - d_2 \mathrm{trm}\left(g_{xxx}\left(\underset{nx^2 \times nx^2}{I} \otimes \Gamma\right)\right).
\end{aligned}
$$

Equation (47) can easily be solved using standard matrix algebra.

7. Performance

I demonstrate the performance of my solution method using 4 models of various sizes in this section. I record the times taken to solve each model (in seconds) for a second-order solution (finding the g_{xx}, h_{xx}, $g_{\sigma\sigma}$ and $h_{\sigma\sigma}$ terms) and a third-order solution (finding the g_{xxx}, h_{xxx}, $g_{\sigma\sigma x}$, $h_{\sigma\sigma x}$, $g_{\sigma\sigma\sigma}$ and $h_{\sigma\sigma\sigma}$ terms). I also compare the performance between the Kamenik solution algorithm and the Martin and Van Loan solution method for solving the Sylvester equations, and how the algorithms perform on a 32 bit computer (2 cores), a 64 bit computer (8 cores) and using the Matlab/Fortran mex functions on the 64 bit computer (8 cores).[7] I have only written Fortran code for the Kamenik algorithm, so there are no times recorded in the Matlab/Fortran mex column for the Martin and Van Loan algorithm.

The first model is a very simple RBC model with external habit formation. The model has 4 predetermined variables and 2 non-predetermined variables (6 equations in total). The equations are presented in Appendix D.1.

Table 1: Computation Times: RBC Model with habit

	32 bit Matlab	64 bit Matlab	64 bit Matlab/Fortran Mex
Second-order*	0.002215	0.003068	1.644566×10^{-4}
Second-order**	0.003661	0.003590	NA
Third-order*	0.009165	0.007501	0.001987
Third-order**	0.016965	0.015241	NA

[1] Model size: n = 6, nx = 4, ny = 2.
[2] * = Kamenik algorithm, ** = Martin and Van Loan algorithm.

[7] The 32bit desktop pc has an Intel Core 2Duo 3.0GHz processor with 4096MB RAM. The 64bit computer has an Intel Xeon 3.5GHz processor with 8 cores and 32756MB RAM.

The second model I test is a simple New Keynesian DSGE model with habit formation, Rotemberg pricing with indexation and persistence in the Taylor rule. The model has 11 predetermined variables and 5 non-predetermined variables (16 equations in total). See Appendix D.2 for a description of the model equations.

Table 2: Computation Times: NK Model

	32 bit Matlab	64 bit Matlab	64 bit Matlab/Fortran Mex
Second-order*	0.014846	0.012749	0.002766
Second-order**	0.039877	0.029043	NA
Third-order*	0.209835	0.194977	0.043463
Third-order**	0.525145	0.421423	NA

[1] Model size: n = 16, nx = 11, ny = 5.
[2] * = Kamenik algorithm, ** = Martin and Van Loan algorithm.

The third model is a Gali and Monacelli type open economy model (see Gali & Monacelli, 2008), with habit formation, Calvo pricing with indexation and persistence in the Taylor rule. The model has 21 predetermined variables and 11 non-predetermined variables (32 equations in total). The model equations are presented in Appendix D.3.

Table 3: Computation Times: Open Economy NK Model

	32 bit Matlab	64 bit Matlab	64 bit Matlab/Fortran Mex
Second-order*	0.079749	0.062015	0.017121
Second-order**	0.219507	0.180435	NA
Third-order*	3.832276	2.326913	1.002582
Third-order**	7.015808	5.138845	NA

[1] Model size: n = 32, nx = 21, ny = 11.
[2] * = Kamenik algorithm, ** = Martin and Van Loan algorithm.

The fourth model is a two country open economy model with Epstein Zin preferences, Rotemberg pricing with indexation and persistence in the Taylor rule. The model has 20 predetermined variables and 33 non-predetermined variables (53 equations in total). See Appendix D.4 for a description of the model equations.

Table 4: Computation Times: Open Economy NK EZ Model

	32 bit Matlab	64 bit Matlab	64 bit Matlab/Fortran Mex
Second-order*	0.127550	0.099659	0.043101
Second-order**	0.327706	0.245445	NA
Third-order*	8.665429	5.026511	3.089053
Third-order**	12.715805	8.773993	NA

[1] Model size: n = 53, nx = 20, ny = 33.
[2] * = Kamenik algorithm, ** = Martin and Van Loan algorithm.

From all four examples, the Matlab/Fortran mex code is always faster than the pure Matlab code, as would be expected. The Matlab/Fortran mex version of the code using the Kamenik algorithm is between 1.6 and 18 times faster than the same code written in Matlab on the same platform. The Kamenik algorithm is always faster than the Martin and Van Loan algorithm. This is probably because it is faster to solve the complex eigenvalues from the real Schur decomposition using real algebra than it is using complex algebra.

Using the Matlab/Fortran mex code, it takes slightly more than 3 seconds to find a second and a third-order approximation of a model with 53 equations in total. This time is comparable to Dynare/Dynare++.[8] Using the Matlab version of the code on the 64 bit computer it takes approximately 5.13 seconds to find a second and a third-order approximation of the same model, and less than 9 seconds to find the same second and third-order solutions on a 32 bit desk top pc. Trying to find a third-order approximation of the same model using Matlab code from Andreasen (2011) or from Binning (2013) on the 32 bit desk top pc would be impossible as it would require too much memory.

The solution times for the third-order approximation include the solution of the $g_{\sigma\sigma\sigma}$ and $h_{\sigma\sigma\sigma}$ terms, if these were not required because the shocks were symmetrically distributed, the third-order solution would take less time to solve.

8. Conclusion

In this paper, I have presented a new method for solving second and third-order approximations of DSGE models. In particular I have presented the matrix chain rules and algebra that are consistent with using a recursive Sylvester equation solution. I have also shown that this solution method can solve small and medium size DSGE models in a competitive

[8]Based on times from the Dynare++ website for models of comparable size.

time, using only Matlab code. My Matlab/Fortran mex code provides a quick solution and is comparable in speed to Dynare/Dynare++. The routines that accompany this paper are standalone, so they do not require any additional toolboxes to run and they can easily be combined with other Matlab code, something that should make them attractive to practitioners who require a lot of flexibility, for example those developing estimation routines for non-linear DSGE models. Existing routines in Dynare/Dynare++ are fast, but are not so flexible in there implementation, making them more difficult to combine with external routines in an efficient way.

Appendix A. Matrix trace and expectations

In this appendix I explain how to derive the definition of the matrix trace I use in this paper. Let ϵ be an $m \times 1$ random vector, X be an $n \times m$ matrix and A be an $n \times n$ matrix then the expectation of the matrix product $\epsilon' X' A X \epsilon$ is given by

$$\mathrm{E}\left[\epsilon' X' A X \epsilon\right] = \left[\mathrm{E}(\epsilon)'\mathrm{E}(X)'\right] A \left[\mathrm{E}(x)\mathrm{E}(\epsilon)\right] + \mathrm{tr}(X' A X \Sigma) \tag{A.1}$$

where $\Sigma = \mathrm{E}\left[\epsilon\epsilon'\right]$, (see Rice, 2007, chapter 14).
If I assume that $\mathrm{E}\left[\epsilon\right] = 0$, then I have

$$\mathrm{E}\left[\epsilon' X' A X \epsilon\right] = \mathrm{tr}(X' A X \Sigma) \tag{A.2}$$

Using $\mathrm{tr}(BC) = \mathrm{vec}(B')'\mathrm{vec}(C)$ I can rewrite equation (A.2) as

$$\begin{aligned} \mathrm{E}\left[\epsilon' X' A X \epsilon\right] &= \mathrm{tr}(I \Sigma X' A X) \\ &= \mathrm{vec}(I')'\mathrm{vec}(\Sigma X' A X) \\ &= \mathrm{vec}(I)'(X' \otimes \Sigma X')\mathrm{vec}(A) \end{aligned} \tag{A.3}$$

Taking the transpose gives

$$\begin{aligned} \mathrm{E}\left[\epsilon' X' A' X \epsilon\right] &= \mathrm{vec}(A)'(X \otimes X\Sigma)\mathrm{vec}(I) \\ &= A^*(X \otimes X\Sigma)\mathrm{vec}(I) \\ &= G\mathrm{vec}(I) \\ &= \sum_{i=1}^{n} G(:, i + n(i-1)) \end{aligned}$$

where $A^* = \mathrm{vec}(A)'$, and $G = A^*(X \otimes X\Sigma)$.

Appendix B. Second-order matrix chain rule proof

Proof From Theorem 1 the proposed second-order matrix chain rule takes the form

$$\mathsf{Y} = \mathsf{H}(\mathsf{M} \otimes \mathsf{M}) + \mathsf{D}\mathsf{N}. \tag{B.1}$$

To prove this a second-order matrix chain rule I need to show that Faà di Bruno's formula holds for each element in Y. Equation (B.1) can be rewritten as

$$\mathsf{Y} = \mathsf{S}^1 + \mathsf{S}^2,$$

where $\mathsf{S}^1 = \mathsf{H}(\mathsf{M} \otimes \mathsf{M})$ and $\mathsf{S}^2 = \mathsf{DN}$. Showing that Faà di Bruno's formula holds for each element in Y means showing that

$$\begin{aligned} y_{1,i+n(j-1)} &= s^1_{1,i+n(j-1)} + s^2_{1,i+n(j-1)}, \\ &= \sum_{a=1}^{m}\sum_{b=1}^{m} f_{a,b} g^a_i g^b_j + \sum_{a=1}^{m} f_a g^a_{i,j}, \end{aligned}$$

where

$$\begin{aligned} s^1_{1,i+n(j-1)} &= \sum_{a=1}^{m}\sum_{b=1}^{m} f_{a,b} g^a_i g^b_j, \\ s^2_{1,i+n(j-1)} &= \sum_{a=1}^{m} f_a g^a_{i,j}, \\ y_{1,i+n(j-1)} &= y_{i,j}. \end{aligned}$$

This can be done in 2 steps, first I need to show that $s^1_{1,i+n(j-1)} = \sum_{a=1}^{m}\sum_{b=1}^{m} f_{a,b} g^a_i g^b_j$ and then I need to show that $s^2_{1,i+n(j-1)} = \sum_{a=1}^{m} f_a g^a_{i,j}$.

Step 1

I need to show that $s^1_{1,i+n(j-1)} = \sum_{a=1}^{m}\sum_{b=1}^{m} f_{a,b} g^a_i g^b_j$. I begin by defining Θ^1

$$\underset{m^2\times n^2}{\Theta^1} = \mathsf{M} \otimes \mathsf{M} = \begin{bmatrix} P^1_1 & \cdots & P^1_i & \cdots & P^1_n \\ \vdots & & \vdots & & \vdots \\ P^a_1 & \cdots & P^a_i & \cdots & P^a_n \\ \vdots & & \vdots & & \vdots \\ P^m_1 & \cdots & P^m_i & \cdots & P^m_n \end{bmatrix},$$

where

$$\underset{m\times n}{P^a_i} = \begin{bmatrix} g^a_i g^1_1 & \cdots & g^a_i g^1_j & \cdots & g^a_i g^1_n \\ \vdots & & \vdots & & \vdots \\ g^a_i g^b_1 & \cdots & g^a_i g^b_j & \cdots & g^a_i g^b_n \\ \vdots & & \vdots & & \vdots \\ g^a_i g^m_1 & \cdots & g^a_i g^m_j & \cdots & g^a_i g^m_n \end{bmatrix}.$$

The element in the $b+m(a-1)$th row and the $i+n(j-1)$th column of Θ^1 is given by

$$\theta^1_{b+m(a-1),i+n(j-1)} = \mathsf{g}^a_i \mathsf{g}^b_j.$$

The elements in H are indexed such that $\mathsf{h}_{1,b+m(a-1)} = f_{a,b}$. Premultiplying Θ^1 by H

$$\mathsf{S}^1 = \mathsf{H}(\mathsf{M} \otimes \mathsf{M}) = \begin{bmatrix} \sum_{a=1}^m \sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^a_1\mathsf{g}^b_1 \\ \vdots \\ \sum_{a=1}^m \sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^a_i\mathsf{g}^b_1 \\ \vdots \\ \sum_{a=1}^m \sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^a_n\mathsf{g}^b_1 \\ \vdots \\ \sum_{a=1}^m \sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^a_1\mathsf{g}^b_j \\ \vdots \\ \sum_{a=1}^m \sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^a_i\mathsf{g}^b_j \\ \vdots \\ \sum_{a=1}^m \sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^a_n\mathsf{g}^b_j \\ \vdots \\ \sum_{a=1}^m \sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^a_1\mathsf{g}^b_n \\ \vdots \\ \sum_{a=1}^m \sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^a_i\mathsf{g}^b_n \\ \vdots \\ \sum_{a=1}^m \sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^a_n\mathsf{g}^b_n \end{bmatrix}'$$

where

$$\mathsf{s}^1_{1,i+n(j-1)} = \sum_{a=1}^m \sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^a_i\mathsf{g}^b_j,$$

as required.

Step 2

I need to show that $\mathsf{s}^2_{1,i+n(j-1)} = \sum_{a=1}^m \mathsf{f}_a \mathsf{g}^a_{i,j}$ where $\mathsf{S}^2 = \mathsf{DN}$. As defined in section 3 the elements in D and N are indexed as follows: $\mathsf{d}_{1,a} = \mathsf{f}_a$ and $\mathsf{n}_{a,i+n(j-1)} = \mathsf{g}^a_{j,i}$. So that

$$\mathsf{S}^2 = \mathsf{DN} = \begin{bmatrix} \sum_{a=1}^{m} \mathsf{f}_a \mathsf{g}^a_{1,1} \\ \vdots \\ \sum_{a=1}^{m} \mathsf{f}_a \mathsf{g}^a_{i,1} \\ \vdots \\ \sum_{a=1}^{m} \mathsf{f}_a \mathsf{g}^a_{n,1} \\ \vdots \\ \sum_{a=1}^{m} \mathsf{f}_a \mathsf{g}^a_{1,j} \\ \vdots \\ \sum_{a=1}^{m} \mathsf{f}_a \mathsf{g}^a_{i,j} \\ \vdots \\ \sum_{a=1}^{m} \mathsf{f}_a \mathsf{g}^a_{n,j} \\ \vdots \\ \sum_{a=1}^{m} \mathsf{f}_a \mathsf{g}^a_{1,n} \\ \vdots \\ \sum_{a=1}^{m} \mathsf{f}_a \mathsf{g}^a_{i,n} \\ \vdots \\ \sum_{a=1}^{m} \mathsf{f}_a \mathsf{g}^a_{n,n} \end{bmatrix}',$$

with

$$\mathsf{s}^2_{1,i+n(j-1)} = \sum_{a=1}^{m} \mathsf{f}_a \mathsf{g}^a_{i,j},$$

as required.

Appendix C. Third-order matrix chain rule proof

Proof From Theorem 2, the proposed third-order chain rule is given by

$$\mathsf{Z} = \mathsf{T}(\mathsf{M} \otimes \mathsf{M} \otimes \mathsf{M}) + \mathsf{H}(\mathsf{M} \otimes \mathsf{N}) + \mathsf{H}(\mathsf{N} \otimes \mathsf{M}) + \mathsf{H}\left(\mathsf{M} \otimes \underset{m \times m}{I}\right)\mathsf{N}^* + \mathsf{DK},$$

this can be rewritten as

$$\mathsf{Z} = \mathsf{S}^1 + \mathsf{S}^2 + \mathsf{S}^3 + \mathsf{S}^4 + \mathsf{S}^5,$$

where

$$\begin{aligned}
\mathsf{S}^1 &= \mathsf{T}(\mathsf{M} \otimes \mathsf{M} \otimes \mathsf{M}),\\
\mathsf{S}^2 &= \mathsf{H}(\mathsf{M} \otimes \mathsf{N}),\\
\mathsf{S}^3 &= \mathsf{H}(\mathsf{N} \otimes \mathsf{M}),\\
\mathsf{S}^4 &= \mathsf{H}\left(\mathsf{M} \otimes \underset{m\times m}{I}\right)\mathsf{N}^*,\\
\mathsf{S}^5 &= \mathsf{DK}.
\end{aligned}$$

The proof for the Theorem 2 proceeds in the same fashion as the proof for Theorem 1, namely I need to show that Faà di Bruno's formula holds for each element in Z or more specifically the following 5 equations hold

$$\begin{aligned}
\mathsf{s}^1_{1,i+n(j-1)+n^2(k-1)} &= \sum_{a=1}^{m}\sum_{b=1}^{m}\sum_{c=1}^{m} \mathsf{f}_{a,b,c}\mathsf{g}^a_i\mathsf{g}^b_j\mathsf{g}^c_k,\\
\mathsf{s}^2_{1,i+n(j-1)+n^2(k-1)} &= \sum_{a=1}^{m}\sum_{b=1}^{m} \mathsf{f}_{a,b}\mathsf{g}^a_k\mathsf{g}^b_{i,j},\\
\mathsf{s}^3_{1,i+n(j-1)+n^2(k-1)} &= \sum_{a=1}^{m}\sum_{b=1}^{m} \mathsf{f}_{a,b}\mathsf{g}^b_{k,j}\mathsf{g}^a_i,\\
\mathsf{s}^4_{1,i+n(j-1)+n^2(k-1)} &= \sum_{a=1}^{m}\sum_{b=1}^{m} \mathsf{f}_{a,b}\mathsf{g}^a_j\mathsf{g}^b_{i,k},\\
\mathsf{s}^5_{1,i+n(j-1)+n^2(k-1)} &= \sum_{a=1}^{m} \mathsf{f}_a\mathsf{g}^a_{i,j,k}.
\end{aligned}$$

I do this in 5 steps.

Step 1

In this step I need to show that $\mathsf{s}^1_{1,i+n(j-1)+n^2(k-1)} = \sum_{a=1}^{m}\sum_{b=1}^{m}\sum_{c=1}^{m} \mathsf{f}_{a,b,c}\mathsf{g}^a_i\mathsf{g}^b_j\mathsf{g}^c_k$. I begin by defining Θ^2 such that

$$\underset{m^3\times n^3}{\Theta^2} = \mathsf{M}\otimes\mathsf{M}\otimes\mathsf{M} = \begin{bmatrix}
\mathsf{Q}^1_1 & \cdots & \mathsf{Q}^1_k & \cdots & \mathsf{Q}^1_n\\
\vdots & & \vdots & & \vdots\\
\mathsf{Q}^c_1 & \cdots & \mathsf{Q}^c_k & \cdots & \mathsf{Q}^c_n\\
\vdots & & \vdots & & \vdots\\
\mathsf{Q}^m_1 & \cdots & \mathsf{Q}^m_k & \cdots & \mathsf{Q}^m_n
\end{bmatrix}$$

where

$$\underset{m^2\times n^2}{\mathsf{Q}_k^c} = (\mathsf{M}\otimes\mathsf{M})\mathsf{g}_k^c = \begin{bmatrix} \mathsf{Q}_{1,k}^{1,c} & \cdots & \mathsf{Q}_{j,k}^{1,c} & \cdots & \mathsf{Q}_{n,k}^{1,c} \\ \vdots & & \vdots & & \vdots \\ \mathsf{Q}_{1,k}^{b,c} & \cdots & \mathsf{Q}_{j,k}^{b,c} & \cdots & \mathsf{Q}_{n,k}^{b,c} \\ \vdots & & \vdots & & \vdots \\ \mathsf{Q}_{1,k}^{m,c} & \cdots & \mathsf{Q}_{j,k}^{m,c} & \cdots & \mathsf{Q}_{n,k}^{m,c} \end{bmatrix}$$

and

$$\underset{m\times n}{\mathsf{Q}_{j,k}^{b,c}} = \mathsf{M}\mathsf{g}_j^b\mathsf{g}_k^c = \begin{bmatrix} \mathsf{g}_1^1\mathsf{g}_j^b\mathsf{g}_k^c & \cdots & \mathsf{g}_i^1\mathsf{g}_j^b\mathsf{g}_k^c & \cdots & \mathsf{g}_n^1\mathsf{g}_j^b\mathsf{g}_k^c \\ \vdots & & \vdots & & \vdots \\ \mathsf{g}_1^a\mathsf{g}_j^b\mathsf{g}_k^c & \cdots & \mathsf{g}_i^a\mathsf{g}_j^b\mathsf{g}_k^c & \cdots & \mathsf{g}_n^a\mathsf{g}_j^b\mathsf{g}_k^c \\ \vdots & & \vdots & & \vdots \\ \mathsf{g}_1^m\mathsf{g}_j^b\mathsf{g}_k^c & \cdots & \mathsf{g}_i^m\mathsf{g}_j^b\mathsf{g}_k^c & \cdots & \mathsf{g}_n^m\mathsf{g}_j^b\mathsf{g}_k^c \end{bmatrix}$$

so that the element in the $a+m(b-1)+m^2(c-1)$th row and the $i+n(j-1)+n^2(k-1)$th column is given by

$$\theta^2_{a+m(b-1)+m^2(c-1),i+n(j-1)+n^2(k-1)} = \mathsf{g}_i^a\mathsf{g}_j^b\mathsf{g}_k^c.$$

From section 3, the element in the 1st row and the $a+m(b-1)+m^2(c-1)$th column of T is

$$\mathsf{t}_{1,a+m(b-1)+m^2(c-1)} = f_{a,b,c}.$$

Combining these terms gives

$$\underset{1\times n^3}{\mathsf{S}^1} = \mathsf{T}\Theta^2 = \mathsf{T}(\mathsf{M}\otimes\mathsf{M}\otimes\mathsf{M}) = \begin{bmatrix} \sum_{a=1}^{m}\sum_{b=1}^{m}\sum_{c=1}^{m}\mathsf{f}_{a,b,c}\mathsf{g}_1^a\mathsf{g}_1^b\mathsf{g}_1^c \\ \sum_{a=1}^{m}\sum_{b=1}^{m}\sum_{c=1}^{m}\mathsf{f}_{a,b,c}\mathsf{g}_2^a\mathsf{g}_1^b\mathsf{g}_1^c \\ \vdots \\ \sum_{a=1}^{m}\sum_{b=1}^{m}\sum_{c=1}^{m}\mathsf{f}_{a,b,c}\mathsf{g}_n^a\mathsf{g}_1^b\mathsf{g}_1^c \\ \sum_{a=1}^{m}\sum_{b=1}^{m}\sum_{c=1}^{m}\mathsf{f}_{a,b,c}\mathsf{g}_1^a\mathsf{g}_2^b\mathsf{g}_1^c \\ \vdots \\ \sum_{a=1}^{m}\sum_{b=1}^{m}\sum_{c=1}^{m}\mathsf{f}_{a,b,c}\mathsf{g}_n^a\mathsf{g}_n^b\mathsf{g}_1^c \\ \sum_{a=1}^{m}\sum_{b=1}^{m}\sum_{c=1}^{m}\mathsf{f}_{a,b,c}\mathsf{g}_1^a\mathsf{g}_1^b\mathsf{g}_2^c \\ \vdots \\ \sum_{a=1}^{m}\sum_{b=1}^{m}\sum_{c=1}^{m}\mathsf{f}_{a,b,c}\mathsf{g}_i^a\mathsf{g}_j^b\mathsf{g}_k^c \\ \vdots \\ \sum_{a=1}^{m}\sum_{b=1}^{m}\sum_{c=1}^{m}\mathsf{f}_{a,b,c}\mathsf{g}_n^a\mathsf{g}_n^b\mathsf{g}_n^c \end{bmatrix}',$$

where it can be easily verified that

$$\mathsf{s}^1_{1,i+n(j-1)+n^2(k-1)} = \sum_{a=1}^{m}\sum_{b=1}^{m}\sum_{c=1}^{m}\mathsf{f}_{a,b,c}\mathsf{g}_i^a\mathsf{g}_j^b\mathsf{g}_k^c.$$

Step 2

In this step I need to show that $\mathsf{s}^2_{1,i+n(j-1)+n^2(k-1)} = \sum_{a=1}^{m}\sum_{b=1}^{m}\mathsf{f}_{a,b}\mathsf{g}_k^a\mathsf{g}_{i,j}^b$. I begin by defining Θ^3 so that

$$\underset{m^2\times n^3}{\Theta^3} = \mathsf{M}\otimes\mathsf{N} = \begin{bmatrix} \mathsf{R}_1^1 & \cdots & \mathsf{R}_k^1 & \cdots & \mathsf{R}_n^1 \\ \vdots & & \vdots & & \vdots \\ \mathsf{R}_1^a & \cdots & \mathsf{R}_k^a & \cdots & \mathsf{R}_n^a \\ \vdots & & \vdots & & \vdots \\ \mathsf{R}_1^m & \cdots & \mathsf{R}_k^m & \cdots & \mathsf{R}_n^m \end{bmatrix},$$

where

$$\underset{m\times n^2}{\mathsf{R}_k^a} = \mathsf{g}_k^a\mathsf{N}.$$

The element in the $b+m(a-1)$th row and the $i+n(j-1)+n^2(k-1)$ column of Θ^3 is given by

$$\theta^3_{b+m(a-1),i+n(j-1)+n^2(k-1)} = \mathsf{g}^a_k \mathsf{g}^b_{i,j}.$$

From section 3, the elements in H are referenced so that: $\mathsf{h}_{1,b+m(a-1)} = \mathsf{f}_{b,a}$. So that

$$\mathsf{S}^2 = \mathsf{H}\Theta^3 = \mathsf{H}(\mathsf{M}\otimes\mathsf{N}) = \begin{bmatrix} \sum_{a=1}^m\sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^a_1\mathsf{g}^b_{1,1} \\ \sum_{a=1}^m\sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^a_1\mathsf{g}^b_{2,1} \\ \vdots \\ \sum_{a=1}^m\sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^a_1\mathsf{g}^b_{n,1} \\ \sum_{a=1}^m\sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^a_1\mathsf{g}^b_{1,2} \\ \vdots \\ \sum_{a=1}^m\sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^a_1\mathsf{g}^b_{1,n} \\ \vdots \\ \sum_{a=1}^m\sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^a_1\mathsf{g}^b_{n,n} \\ \sum_{a=1}^m\sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^a_2\mathsf{g}^b_{1,1} \\ \vdots \\ \sum_{a=1}^m\sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^a_k\mathsf{g}^b_{i,j} \\ \vdots \\ \sum_{a=1}^m\sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^a_n\mathsf{g}^b_{n,n} \end{bmatrix}',$$

where

$$\mathsf{s}^2_{1,i+n(j-1)+n^2(k-1)} = \sum_{a=1}^m\sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^a_k\mathsf{g}^b_{i,j},$$

as required.

Step 3

In this step I need to show that $\mathsf{s}^3_{1,i+n(j-1)+n^2(k-1)} = \sum_{a=1}^m\sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^b_{k,j}\mathsf{g}^a_i$. I begin by defining

Θ^4 so that

$$\underset{m^2\times n^3}{\Theta^4} = \mathsf{N}\otimes\mathsf{M} = \begin{bmatrix} \mathsf{U}^1_{1,1} & \cdots & \mathsf{U}^1_{k,j} & \cdots & \mathsf{U}^1_{n,n} \\ \vdots & & \vdots & & \vdots \\ \mathsf{U}^b_{1,1} & \cdots & \mathsf{U}^b_{k,j} & \cdots & \mathsf{U}^b_{n,n} \\ \vdots & & \vdots & & \vdots \\ \mathsf{U}^m_{1,1} & \cdots & \mathsf{U}^m_{k,j} & \cdots & \mathsf{U}^m_{n,n} \end{bmatrix},$$

where

$$\mathsf{U}^b_{k,j} = \mathsf{g}^b_{k,j}\mathsf{M}.$$

The element in the $a+m(b-1)$th row and the $i+n(j-1)+n^2(k-1)$th column is given by

$$\theta^4_{a+m(b-1),i+n(j-1)+n^2(k-1)} = \mathsf{g}^b_{k,j}\mathsf{g}^a_i.$$

From section 3, the elements in H are referenced so that: $\mathsf{h}_{1,b+m(a-1)} = \mathsf{f}_{b,a}$. From the definition of S^3

$$\mathsf{S}^3 = \mathsf{H}\Theta^4 = \mathsf{H}(\mathsf{N}\otimes\mathsf{M}) = \begin{bmatrix} \sum_{a=1}^m\sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^b_{1,1}\mathsf{g}^a_1 \\ \sum_{a=1}^m\sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^b_{1,1}\mathsf{g}^a_2 \\ \vdots \\ \sum_{a=1}^m\sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^b_{1,1}\mathsf{g}^a_n \\ \sum_{a=1}^m\sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^b_{1,2}\mathsf{g}^a_1 \\ \vdots \\ \sum_{a=1}^m\sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^b_{1,n}\mathsf{g}^a_n \\ \sum_{a=1}^m\sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^b_{2,1}\mathsf{g}^a_1 \\ \vdots \\ \sum_{a=1}^m\sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^b_{k,j}\mathsf{g}^a_i \\ \vdots \\ \sum_{a=1}^m\sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^b_{n,n}\mathsf{g}^a_n \end{bmatrix}',$$

where

$$\mathsf{s}^3_{1,i+n(j-1)+n^2(k-1)} = \sum_{a=1}^m\sum_{b=1}^m \mathsf{f}_{a,b}\mathsf{g}^b_{k,j}\mathsf{g}^a_i,$$

as required.

Step 4

In this step I need to show that $\mathsf{s}^4_{1,i+n(j-1)+n^2(k-1)} = \sum_{a=1}^{m}\sum_{b=1}^{m} \mathsf{f}_{a,b}\mathsf{g}^a_j\mathsf{g}^b_{i,k}$. I begin by defining Θ^5 so that

$$\Theta^5 = \left(\mathsf{M} \otimes \underset{m\times m}{I}\right)\mathsf{N}^* = \left[\begin{array}{ccccc} \mathsf{M}\otimes\tilde{\mathsf{N}}^1 & \cdots & \mathsf{M}\otimes\tilde{\mathsf{N}}^k & \cdots & \mathsf{M}\otimes\tilde{\mathsf{N}}^n \end{array}\right],$$

where

$$\mathsf{M}\otimes\tilde{\mathsf{N}}^k = \begin{bmatrix}
\mathsf{g}^1_1\mathsf{g}^1_{1,k} & \mathsf{g}^1_1\mathsf{g}^1_{2,k} & \cdots & \mathsf{g}^1_1\mathsf{g}^1_{n,k} & \mathsf{g}^1_2\mathsf{g}^1_{1,k} & \cdots & \mathsf{g}^1_j\mathsf{g}^1_{i,k} & \cdots & \mathsf{g}^1_n\mathsf{g}^1_{n,k} \\
\mathsf{g}^1_1\mathsf{g}^2_{1,k} & \mathsf{g}^1_1\mathsf{g}^2_{2,k} & \cdots & \mathsf{g}^1_1\mathsf{g}^2_{n,k} & \mathsf{g}^1_2\mathsf{g}^2_{1,k} & \cdots & \mathsf{g}^1_j\mathsf{g}^2_{i,k} & \cdots & \mathsf{g}^1_n\mathsf{g}^2_{n,k} \\
\vdots & \vdots & & \vdots & \vdots & & \vdots & & \vdots \\
\mathsf{g}^1_1\mathsf{g}^b_{1,k} & \mathsf{g}^1_1\mathsf{g}^b_{2,k} & \cdots & \mathsf{g}^1_1\mathsf{g}^b_{n,k} & \mathsf{g}^1_2\mathsf{g}^b_{1,k} & \cdots & \mathsf{g}^1_j\mathsf{g}^b_{i,k} & \cdots & \mathsf{g}^1_n\mathsf{g}^b_{n,k} \\
\vdots & \vdots & & \vdots & \vdots & & \vdots & & \vdots \\
\mathsf{g}^1_1\mathsf{g}^m_{1,k} & \mathsf{g}^1_1\mathsf{g}^m_{2,k} & \cdots & \mathsf{g}^1_1\mathsf{g}^m_{n,k} & \mathsf{g}^1_2\mathsf{g}^m_{1,k} & \cdots & \mathsf{g}^1_j\mathsf{g}^m_{i,k} & \cdots & \mathsf{g}^1_n\mathsf{g}^m_{n,k} \\
\mathsf{g}^2_1\mathsf{g}^1_{1,k} & \mathsf{g}^2_1\mathsf{g}^1_{2,k} & \cdots & \mathsf{g}^2_1\mathsf{g}^1_{n,k} & \mathsf{g}^2_2\mathsf{g}^1_{1,k} & \cdots & \mathsf{g}^2_j\mathsf{g}^1_{i,k} & \cdots & \mathsf{g}^2_n\mathsf{g}^1_{n,k} \\
\vdots & \vdots & & \vdots & \vdots & & \vdots & & \vdots \\
\mathsf{g}^2_1\mathsf{g}^1_{1,k} & \mathsf{g}^2_1\mathsf{g}^1_{2,k} & \cdots & \mathsf{g}^2_1\mathsf{g}^1_{n,k} & \mathsf{g}^2_2\mathsf{g}^1_{1,k} & \cdots & \mathsf{g}^2_j\mathsf{g}^1_{i,k} & \cdots & \mathsf{g}^2_n\mathsf{g}^1_{n,k} \\
\vdots & \vdots & & \vdots & \vdots & & \vdots & & \vdots \\
\mathsf{g}^a_1\mathsf{g}^b_{1,k} & \mathsf{g}^a_1\mathsf{g}^b_{2,k} & \cdots & \mathsf{g}^a_1\mathsf{g}^b_{n,k} & \mathsf{g}^a_2\mathsf{g}^b_{1,k} & \cdots & \mathsf{g}^a_j\mathsf{g}^b_{i,k} & \cdots & \mathsf{g}^a_n\mathsf{g}^b_{n,k} \\
\vdots & \vdots & & \vdots & \vdots & & \vdots & & \vdots \\
\mathsf{g}^m_1\mathsf{g}^m_{1,k} & \mathsf{g}^m_1\mathsf{g}^m_{2,k} & \cdots & \mathsf{g}^m_1\mathsf{g}^m_{n,k} & \mathsf{g}^m_2\mathsf{g}^m_{1,k} & \cdots & \mathsf{g}^m_j\mathsf{g}^m_{i,k} & \cdots & \mathsf{g}^m_n\mathsf{g}^m_{n,k}
\end{bmatrix}.$$

The element in the $b+m(a-1)$th row and the $i+n(j-1)+n^2(k-1)$th column is given by

$$\theta^5_{b+m(a-1),i+n(j-1)+n^2(k-1)}.$$

From the definition of S^4

$$\mathsf{S}^4 = \mathsf{H}\left(\mathsf{M} \otimes \underset{m\times m}{I}\right)\mathsf{N}^* = \begin{bmatrix} \sum_{a=1}^{m}\sum_{b=1}^{m} \mathsf{f}_{a,b}\mathsf{g}_1^a\mathsf{g}_{1,1}^b \\ \sum_{a=1}^{m}\sum_{b=1}^{m} \mathsf{f}_{a,b}\mathsf{g}_1^a\mathsf{g}_{2,1}^b \\ \vdots \\ \sum_{a=1}^{m}\sum_{b=1}^{m} \mathsf{f}_{a,b}\mathsf{g}_1^a\mathsf{g}_{n,1}^b \\ \sum_{a=1}^{m}\sum_{b=1}^{m} \mathsf{f}_{a,b}\mathsf{g}_2^a\mathsf{g}_{1,1}^b \\ \vdots \\ \sum_{a=1}^{m}\sum_{b=1}^{m} \mathsf{f}_{a,b}\mathsf{g}_2^a\mathsf{g}_{n,1}^b \\ \vdots \\ \sum_{a=1}^{m}\sum_{b=1}^{m} \mathsf{f}_{a,b}\mathsf{g}_j^a\mathsf{g}_{i,k}^b \\ \vdots \\ \sum_{a=1}^{m}\sum_{b=1}^{m} \mathsf{f}_{a,b}\mathsf{g}_n^a\mathsf{g}_{n,n}^b \end{bmatrix}',$$

where

$$\mathsf{s}^4_{1,i+n(j-1)+n^2(k-1)} = \sum_{a=1}^{m}\sum_{b=1}^{m} \mathsf{f}_{a,b}\mathsf{g}_j^a\mathsf{g}_{i,k}^b,$$

as required.

Step 5

Finally, I need to show that $\mathsf{s}^5_{1,i+n(j-1)+n^2(k-1)} = \sum_{a=1}^{m} \mathsf{f}_a\mathsf{g}^a_{i,j,k}$. From section 3 the elements in K and D are referenced as follows

$$\mathsf{k}_{a,i+n(j-1)+n^2(k-1)} = \mathsf{g}^a_{i,j,k},$$

$$\mathsf{d}_{1,a} = \mathsf{f}_a.$$

From the definition of S^5

$$\mathsf{S}^5 = \mathsf{D}\mathsf{K} = \begin{bmatrix} \sum_{a=1}^{m} \mathsf{f}_a \mathsf{g}^a_{1,1,1} \\ \sum_{a=1}^{m} \mathsf{f}_a \mathsf{g}^a_{2,1,1} \\ \vdots \\ \sum_{a=1}^{m} \mathsf{f}_a \mathsf{g}^a_{n,1,1} \\ \sum_{a=1}^{m} \mathsf{f}_a \mathsf{g}^a_{1,2,1} \\ \vdots \\ \sum_{a=1}^{m} \mathsf{f}_a \mathsf{g}^a_{n,n,1} \\ \sum_{a=1}^{m} \mathsf{f}_a \mathsf{g}^a_{1,1,2} \\ \vdots \\ \sum_{a=1}^{m} \mathsf{f}_a \mathsf{g}^a_{i,j,k} \\ \vdots \\ \sum_{a=1}^{m} \mathsf{f}_a \mathsf{g}^a_{n,n,n} \end{bmatrix}' .$$

It follows from the indexation in K and D that

$$\mathsf{s}^5_{1,i+n(j-1)+n^2(k-1)} = \sum_{a=1}^{m} \mathsf{f}_a \mathsf{g}^a_{i,j,k},$$

which is required for the proof.

Appendix D. Example models

Appendix D.1. An RBC model with external habit formation

This section presents the equations for the simple RBC model with external habit formation in Section 7. The variable and parameter descriptions are given in tables D.5 and D.6 respectively.

$$(c_t - \chi c_{t-1})^{-\gamma} - \mathrm{E}_t \left\{ \beta(1 + \alpha a_{t+1} k_t^{\alpha-1} - \delta)(c_{t+1} - \chi c_t)^{-\sigma} \right\} = 0, \tag{D.1}$$

$$k_t + c_t - a_t k_{t-1}^{\alpha} - (1-\delta) k_{t-1} = 0, \tag{D.2}$$

$$a_t - a_{t-1}^{\rho} a_{ss}^{1-\rho} \exp(\varepsilon_t) = 0. \tag{D.3}$$

Table D.5: Variables

Symbol	Description
c_t	Consumption
k_t	Capital
a_t	Technology
ε_t	Technology shock

Table D.6: Parameters

Symbol	Description
γ	Intertemporal elasticity of substitution
χ	Habit persistence parameter
β	Discount factor
α	Capital's share of income
δ	Depreciation rate
ρ	Persistence parameter on technology

Appendix D.2. New Keynesian DSGE

This is the second model used in Section 7. It is a simple New Keynesian DSGE model alá Galí (2009) with external habit formation, Rotemberg price adjustment costs, price indexation and persistence in the Taylor rule. Variable and parameter descriptions are given in tables D.7 and D.8 respectively. The equations of the model are given by

$$(\theta-1)Y_t - \lambda\theta\phi\left(\frac{1}{A_t^c}\right)(C_t-\chi C_{t-1})^{\sigma}\left(\frac{Y_t}{A_t}\right)^{\phi(\nu+1)} + \psi\pi_t Y_t\left(\pi_t - \left(\xi\pi_{t-1}+(1-\xi)\,\bar{\pi}\right)\mu_t\right)$$
$$-\beta\mathrm{E}_t\left\{\begin{array}{c}\left(\frac{C_{t+1}-\chi C_t}{C_t-\chi C_{t-1}}\right)^{-\sigma}\left(\frac{A_t^c}{\bar{A}^c}\right)^{\rho_c-1}\exp(\varepsilon_{t+1}^c)\times\cdots \\ \cdots\times\psi\pi_{t+1}Y_{t+1}\left(\pi_{t+1}-(\xi\pi_t+(1-\xi)\bar{\pi})\,\mu_t^{\rho_\mu}\bar{\mu}^{1-\rho_\mu}\exp(\varepsilon_{t+1}^{\mu})\right)\end{array}\right\} = 0, \quad \text{(D.4)}$$

$$Y_t - C_t - \psi Y_t\left(\pi_t - (\xi\pi_{t-1}+(1-\xi)\pi_{t-1})\,\mu_t\right)^2 = 0, \quad \text{(D.5)}$$

$$\beta\mathrm{E}_t\left\{\left(\frac{C_{t+1}-\chi C_t}{C_t-\chi C_{t-1}}\right)^{-\sigma}\left(\frac{1}{\pi_{t+1}}\right)\left(\frac{A_t^c}{\bar{A}^c}\right)^{\rho_c-1}\exp(\varepsilon_{t+1}^c)\right\} - \frac{1}{R_t} = 0, \quad \text{(D.6)}$$

$$R_t - R_{t-1}^{\rho_r}\left(\bar{R}\left(\frac{Y_t}{\bar{Y}}\right)^{\kappa_y}\left(\frac{\pi_t}{\bar{\pi}}\right)^{\kappa_\pi}\right)^{1-\rho_r}\exp(\varepsilon_t^r) = 0, \quad \text{(D.7)}$$

$$A_t - A_{t-1}^{\rho_a}\bar{A}^{1-\rho_a}\exp(\varepsilon_t^a) = 0, \quad \text{(D.8)}$$

$$A_t^c - \left(A_{t-1}^c\right)^{\rho_c} \left(\bar{A}^c\right)^{1-\rho_c} \exp(\varepsilon_t^c) = 0, \tag{D.9}$$

$$\mu_t - \mu_{t-1}^{\rho_\mu} \bar{\mu}^{1-\rho_\mu} \exp(\varepsilon_t^\mu) = 0. \tag{D.10}$$

Table D.7: Variables

Symbol	Description
Y_t	Output
C_t	Consumption
π_t	Inflation
R_t	Interest Rate
A_t	Technology
A_t^c	Consumption Shock Process
μ_t	Indexation Shock Process
ε_t^a	Technology Shock
ε_t^c	Consumption Preference Shock
ε_t^μ	Indexation Shock

Table D.8: Parameters

Symbol	Description
θ	Elasticity of substitution between differentiated inputs
λ	Weight on disutility of labour
ϕ	Labour's share of income
χ	Habit parameter
σ	Inverse of the intertemporal EOS
ν	Frisch elasticity of labour supply
ψ	Weight on price adjustment costs
ξ	Degree of price indexation
κ_y	Weight on output in the Taylor rule
κ_π	Weight on inflation in the Taylor rule
ρ_r	Persistence in Taylor rule
ρ_a	Persistence term on Technology
ρ_c	Persistence term on Consumption shock
ρ_μ	Persistence term on Indexation shock

Appendix D.3. Small Open Economy Model

The third model used in Section 7 is a small open economy model similar to Gali & Monacelli (2008). Firms in both countries are subject to a Calvo pricing friction, those that are unable to update prices optimally update prices according to an indexation rule. Households in both countries are subject to external habit formation and there is a persistence term in the interest rate rule. The variable descriptions for the home and foreign countries are presented in tables D.13 and D.14 respectively. The parameter definitions are presented in tables D.15 and D.16.

$$\exp(\varepsilon_t^c)\left(C_t - \chi C_{t-1}\right)^{\sigma} - \exp(\varepsilon_t^{c*})\vartheta\left(Y_t^* - \chi^* Y_{t-1}^*\right)^{\sigma^*} S_t \tilde{P}_{H,t} = 0, \tag{D.11}$$

$$Y_t - (1-\mu)\left(\tilde{P}_{H,t}\right)^{-\nu} C_t - \mu^* S_t^{\nu} Y_t^* = 0, \tag{D.12}$$

$$\begin{aligned} &K_{1,t} - \exp(-\varepsilon_t^c)\left(\frac{1}{\tilde{P}_{H,t}}\right)\lambda\left(C_t - \chi C_{t-1}\right)^{\sigma}\left(\frac{Y_t}{A_t}\right)^{\frac{\eta+1}{1-\alpha}}\Delta_t^{\eta} - \\ &\quad \mathrm{E}_t\left\{\theta\left(\pi_{H,t}^{\rho_\pi}\bar{\pi}_H^{1-\rho_\pi}\right)^{\frac{-\epsilon_t}{1-\alpha}}\pi_{H,t+1}^{\frac{\epsilon_t+1-\alpha}{1-\alpha}} K_{1,t+1}\exp(-\varepsilon_t^c)\beta\left(\frac{C_t - \chi C_{t-1}}{C_{t+1} - \chi C_t}\right)^{\sigma}\left(\frac{1}{\pi_{t+1}}\right)\right\} = 0, \end{aligned} \tag{D.13}$$

$$\begin{aligned} &K_{2,t} - Y_t - \\ &\quad \mathrm{E}_t\left\{\theta\left(\bar{\pi}_H^{1-\rho_\pi}\pi_{H,t}^{\rho_\pi}\right)^{1-\epsilon_t}\pi_{H,t+1}^{\epsilon_t} K_{2,t+1}\exp(-\varepsilon_t^c)\beta\left(\frac{C_t - \chi C_{t-1}}{C_{t+1} - \chi C_t}\right)^{\sigma}\left(\frac{1}{\pi_{t+1}}\right)\right\} = 0, \end{aligned} \tag{D.14}$$

$$\tilde{P}_{H,t} - \left[(1-\mu) + \mu S_t^{1-\nu}\right]^{\frac{-1}{1-\nu}} = 0, \tag{D.15}$$

$$\pi_{H,t} - \bar{\pi}_H^{1-\rho_\pi}\pi_{H,t-1}^{\rho_\pi}\left[\frac{\theta}{1-(1-\theta)\left(\frac{\epsilon_t}{(1-\alpha)(\epsilon_t-1)}\right)\left(\frac{K_{1,t}}{K_{2,t}}\right)^{\frac{(1-\epsilon_t)(1-\alpha)}{1-\alpha+\epsilon_t\alpha}}}\right]^{\frac{1}{1-\epsilon_t}} = 0, \tag{D.16}$$

$$\Delta_t - (1-\theta)\left[\frac{1-\theta\left(\frac{\bar{\pi}_H^{1-\rho_\pi}\pi_{H,t-1}^{\rho_\pi}}{\pi_{H,t}}\right)^{1-\epsilon_t}}{1-\theta}\right]^{\frac{-\epsilon_t}{(1-\epsilon_t)(1-\alpha)}} - \theta\pi_{H,t}^{\frac{\epsilon_t}{1-\alpha}}\left(\bar{\pi}_H^{1-\rho_\pi}\pi_{H,t-1}^{\rho_\pi}\right)^{\frac{-\epsilon_t}{1-\alpha}}\Delta_t = 0, \tag{D.17}$$

$$R_t - \mathrm{E}_t\left\{R_t^*\left(\frac{S_{t+1}}{S_t}\right)\left(\frac{\pi_{H,t+1}}{\pi_{t+1}^*}\right)\right\} = 0, \tag{D.18}$$

$$R_t - \left(\left(\frac{\bar{\pi}}{\beta}\right)\left(\frac{Y_t}{\bar{Y}}\right)^{\kappa_y}\left(\frac{\pi_t}{\bar{\pi}}\right)^{\kappa_\pi}\right)^{1-\rho_r} R_{t-1}^{\rho_r}\exp(\varepsilon_t^r) = 0, \tag{D.19}$$

$$\frac{\pi_{H,t}}{\pi_t} - \frac{\tilde{P}_{H,t}}{\tilde{P}_{H,t-1}} = 0, \tag{D.20}$$

$$\beta\mathrm{E}_t\left\{\exp(-\varepsilon_t^{c*})\left(\frac{R_t^*}{\Pi_{t+1}^*}\right)\left(\frac{Y_{t+1}^* - \chi^* Y_t^*}{Y_t^* - \chi^* Y_{t-1}^*}\right)^{-\sigma^*}\right\} - 1 = 0, \tag{D.21}$$

$$K^*_{1,t}-\exp(-\varepsilon^{c*}_t)\lambda^*\left(Y^*_t-\chi^*Y^*_{t-1}\right)^{\sigma^*}\left(\frac{Y^*_t}{A^*_t}\right)^{\frac{\eta^*+1}{1-\alpha^*}}(\Delta^*_t)^{\eta^*}-$$

$$\mathrm{E}_t\left\{\theta^*\left((\pi^*_t)^{\rho^*_\pi}(\bar{\pi}^*)^{1-\rho^*_\pi}\right)^{\frac{\epsilon^*_t}{1-\alpha^*}}(\pi^*_{t+1})^{\frac{\epsilon^*_t+1-\alpha^*}{1-\alpha^*}}K^*_{1,t+1}\exp(-\varepsilon^{c*}_t)\beta\left(\frac{C^*_t-\chi^*C^*_{t-1}}{C^*_{t+1}-\chi^*C^*_t}\right)^{\sigma^*}\left(\frac{1}{\pi^*_{t+1}}\right)\right\}=0, \tag{D.22}$$

$$K^*_{2,t}-Y^*_t-\mathrm{E}_t\left\{\theta^*\left((\bar{\pi}^*)^{1-\rho^*_\pi}(\pi^*_t)^{\rho^*_\pi}\right)^{1-\epsilon^*_t}(\pi^*_{t+1})^{\epsilon^*_t}K^*_{2,t+1}\exp(-\varepsilon^{c*}_t)\beta\left(\frac{C^*_t-\chi^*C^*_{t-1}}{C^*_{t+1}-\chi^*C^*_t}\right)^{\sigma^*}\left(\frac{1}{\pi^*_{t+1}}\right)\right\}=0, \tag{D.23}$$

$$\pi^*_t-(\bar{\pi}^*)^{1-\rho^*_\pi}(\pi^*_{t-1})^{\rho^*_\pi}\left[\frac{\theta^*}{1-(1-\theta^*)\left(\frac{\epsilon^*_t}{(1-\alpha^*)(\epsilon^*_t-1)}\right)\left(\frac{K^*_{1,t}}{K^*_{2,t}}\right)^{\frac{(1-\epsilon^*_t)(1-\alpha^*)}{1-\alpha^*+\epsilon^*_t\alpha^*}}}\right]^{\frac{1}{1-\epsilon^*_t}}=0, \tag{D.24}$$

$$\Delta^*_t-(1-\theta^*)\left[\frac{1-\theta^*\left(\frac{(\bar{\pi}^*)^{1-\rho_\pi}(\pi^*_{t-1})^{\rho_\pi}}{\pi^*_t}\right)^{1-\epsilon^*_t}}{1-\theta^*}\right]^{\frac{-\epsilon^*_t}{(1-\epsilon^*_t)(1-\alpha^*)}}-\theta^*(\pi^*_t)^{\frac{\epsilon^*_t}{1-\alpha^*}}\left((\bar{\pi}^*)^{1-\rho^*_\pi}(\pi^*_{t-1})^{\rho^*_\pi}\right)^{\frac{-\epsilon^*_t}{1-\alpha^*}}\Delta^*_t=0, \tag{D.25}$$

$$R^*_t-\left(\left(\frac{\bar{\pi}^*}{\beta}\right)\left(\frac{Y^*_t}{\bar{Y}^*}\right)^{\kappa^*_y}\left(\frac{\pi^*_t}{\bar{\pi}^*}\right)^{\kappa^*_\pi}\right)^{1-\rho^*_r}(R^*_{t-1})^{\rho^*_r}\exp(\varepsilon^{r*}_t)=0, \tag{D.26}$$

$$A_t-A^{\rho}_{t-1}\bar{A}^{1-\rho}\exp(\varepsilon^a_t)=0, \tag{D.27}$$

$$A^*_t-(A^*)^{\rho^*}_{t-1}(\bar{A}^*)^{1-\rho^*}\exp(\varepsilon^{a*}_t)=0, \tag{D.28}$$

$$\epsilon_t-\epsilon^{\rho_\epsilon}_{t-1}\bar{\epsilon}^{1-\rho_\epsilon}\exp(\varepsilon^{\epsilon}_t)=0, \tag{D.29}$$

$$\epsilon^*_t-(\epsilon^*_{t-1})^{\rho^*_\epsilon}(\bar{\epsilon}^*)^{1-\rho^*_\epsilon}\exp(\varepsilon^{\epsilon*}_t)=0. \tag{D.30}$$

Table D.9: Domestic Variables

Symbol	Description
Y_t	Output
C_t	Consumption
π_t	Inflation
$\pi_{H,t}$	Tradable inflation
$\tilde{P}_{H,t}$	Relative price of domestically produced goods
S_t	Terms of trade
R_t	Interest rate
$K_{1,t}$	Discounted sum of marginal cost
$K_{2,t}$	Discounted sum of demand
A_t	Technology
ϵ_t	Elasticity of substitution between domestically produced tradable goods
ε_t^a	Technology shock
ε_t^c	Consumption preference shock
ε_t^ϵ	Markup shock

Table D.10: Foreign Variables

Symbol	Description
Y_t^*	Output
π_t^*	Inflation
R_t^*	Interest rate
$K_{1,t}^*$	Discounted sum of marginal cost
$K_{2,t}^*$	Discounted sum of demand
A_t^*	Technology
ϵ_t^*	Elasticity of substitution between foreign produced goods
ε_t^{a*}	Technology shock
ε_t^{c*}	Consumption preference shock
$\varepsilon_t^{\epsilon*}$	Markup shock

Table D.11: Domestic Parameters

Symbol	Description
λ	Weight on disutility of labour
$1-\alpha$	Labour's share of income
χ	Domestic habit parameter
σ	Domestic inverse of the intertemporal EOS
η	Frisch elasticity of labour supply
ν	Elasticity of substitution between domestic and foreign goods
$1-\mu$	Home bias
θ	Probability of adjusting prices optimally
ϑ	Scale parameter
ρ_π	Degree of price indexation
κ_y	Weight on output in the Taylor rule
κ_π	Weight on inflation in the Taylor rule
ρ_a	Persistence term on Technology
ρ_c	Persistence term on Consumption shock
ρ_r	Persistence in Taylor rule

Table D.12: Domestic Parameters

Symbol	Description
λ^*	Weight on disutility of labour
$1-\alpha^*$	Labour's share of income
χ^*	Domestic habit parameter
σ^*	Domestic inverse of the intertemporal EOS
θ	Probability of adjusting prices optimally
ρ_π^*	Degree of price indexation
κ_y^*	Weight on output in the Taylor rule
κ_π^*	Weight on inflation in the Taylor rule
ρ_a^*	Persistence term on Technology
ρ_c^*	Persistence term on Consumption shock
ρ_r^*	Persistence in Taylor rule

Appendix D.4. Small Open Economy Model: Epstein Zin Preferences

The final model is a small open economy model with Epstein Zin preferences, Rotemberg price adjustment costs, habit formation, price indexation and persistence in the interest rule. There are also some additional equations included to measure the term premia in both the home and the foreign country.

$$V_t^* - \left[\exp(\varepsilon_t^{c*})\left(\left(C_t^* - \chi^* C_{t-1}^*\right)^{\nu^*}(1-N_t^*)^{1-\nu^*}\right)^{1-\rho^*} + \beta\left(\mathrm{E}_t\left\{(V_{t+1}^*)^{1-\gamma^*}\right\}\right)^{\frac{1-\rho^*}{1-\gamma^*}}\right]^{\frac{1}{1-\rho^*}} = 0, \tag{D.31}$$

$$\exp(-\varepsilon_t^{c*})\beta\left(\frac{(V_{t+1}^*)^{1-\gamma^*}}{\mathrm{E}_t V_{t+1}^*}\right)^{\frac{\rho^*-\gamma^*}{1-\gamma^*}}\left(\frac{(C_{t+1}^*-\chi^* C_t^*)^{\nu^*}(1-N_{t+1}^*)^{1-\nu^*}}{(C_t^*-\chi^* C_{t-1}^*)^{\nu^*}(1-N_t^*)^{1-\nu^*}}\right)^{1-\gamma^*}\left(\frac{C_t^*-\chi^* C_{t-1}^*}{C_{t+1}^*-\chi C_t^*}\right)\left(\frac{1}{\pi_{t+1}^*}\right) - \frac{1}{R_t^*} = 0, \tag{D.32}$$

$$\begin{aligned}
&-\Omega_t^* + \left(\tfrac{1-\nu^*}{\nu^*}\right)\left(\tfrac{C_t^*-\chi C_{t-1}^*}{1-N_t^*}\right)\left(\tfrac{1}{A_t^*}\right)^{\frac{1}{1-\alpha^*}}(C_t^*)^{\frac{\alpha^*}{1-\alpha^*}}\left(\tfrac{\theta^*}{\theta^*-1}\right)\left(\tfrac{1}{1-\alpha^*}\right) - \\
&\quad -\left(\tfrac{\phi^*}{\theta^*-1}\right)\pi_t^* C_t^*\left(\pi_t^*-\tilde{\pi}_t^*\right) + \exp(-\varepsilon_t^{c*})\left(\tfrac{\phi^*}{\theta^*-1}\right)\beta\left(\tfrac{(V_{t+1}^*)^{1-\gamma^*}}{\mathrm{E}_t V_{t+1}^*}\right)^{\frac{\rho^*-\gamma^*}{1-\gamma^*}} \times \cdots \\
&\quad \cdots \times \left(\tfrac{(C_{t+1}^*-\chi^* C_t^*)^{\nu^*}(1-N_{t+1}^*)^{1-\nu^*}}{(C_t^*-\chi^* C_{t-1}^*)^{\nu^*}(1-N_t^*)^{1-\nu^*}}\right)^{1-\gamma^*}\left(\tfrac{C_t^*-\chi^* C_{t-1}^*}{C_{t+1}^*-\chi C_t^*}\right)\pi_{t+1}^* C_{t+1}^*\left(\pi_{t+1}^*-\tilde{\pi}_t^*\right) = 0,
\end{aligned} \tag{D.33}$$

$$C_t^* - A_t^*(N_t^*)^{1-\alpha^*} = 0, \tag{D.34}$$

$$R_t^* - \exp(\varepsilon_t^{r*})\left(\left(\frac{\bar{\pi}^*}{\beta}\right)\left(\frac{C_t^*}{\bar{C}^*}\right)^{\kappa_y^*}\left(\frac{\pi_t^*}{\bar{\pi}^*}\right)^{\kappa_\pi^*}\right)^{1-\rho_r^*}(R_{t-1}^*)^{\rho_r^*} = 0, \tag{D.35}$$

$$\tilde{\pi}_t^* - (\pi_{t-1}^*)^{\xi^*}(\bar{\pi}^*)^{1-\xi^*} = 0, \tag{D.36}$$

$$P_t^{NB*} - 1 - \frac{\delta^{c*} P_{t+1}^{NB*}}{R_t} = 0, \tag{D.37}$$

$$\begin{aligned}
&P_t^{B*} - 1 - \exp(-\varepsilon_t^{c*})\delta^{c*} P_{t+1}^{B*}\beta\left(\frac{(V_{t+1}^*)^{1-\gamma^*}}{\mathrm{E}_t V_{t+1}}\right)^{\frac{\rho^*-\gamma^*}{1-\gamma^*}}\left(\frac{(C_{t+1}^*-\chi^* C_t^*)^{\nu^*}(1-N_{t+1}^*)^{1-\nu^*}}{(C_t^*-\chi^* C_{t-1}^*)^{\nu^*}(1-N_t^*)^{1-\nu^*}}\right)^{1-\gamma^*} \times \cdots \\
&\qquad \cdots \times \left(\frac{C_t^*-\chi^* C_{t-1}^*}{C_{t+1}^*-\chi C_t^*}\right)\left(\frac{1}{\pi_{t+1}^*}\right) = 0,
\end{aligned} \tag{D.38}$$

$$Y_t^{T*} - \frac{\delta^{c*} P_t^{B*}}{P^{B*}-1} = 0, \tag{D.39}$$

$$Y_t^{TN*} - \frac{\delta^{c*} P_t^{BN*}}{P^{BN*}-1} = 0, \tag{D.40}$$

$$T_t^* - \frac{Y_t^{T*}}{Y_t^{TN*}} = 0, \tag{D.41}$$

$$V_t - \left[\exp(\varepsilon_t^c)\left((C_t - \chi C_{t-1})^{\nu}(1-N_t)^{1-\nu}\right)^{1-\rho} + \beta\left(\mathrm{E}_t\left\{(V_{t+1})^{1-\gamma}\right\}\right)^{\frac{1-\rho}{1-\gamma}}\right]^{\frac{1}{1-\rho}} = 0, \tag{D.42}$$

$$R_t - \mathrm{E}_t\left\{R_t^*\left(\frac{S_{t+1}}{S_t}\right)\left(\frac{\pi_{H,t+1}}{\pi_{t+1}^*}\right)\right\} = 0, \tag{D.43}$$

$$\frac{\pi_{H,t}}{\pi_t} - \frac{\tilde{P}_{H,t}}{\tilde{P}_{H,t-1}} = 0, \tag{D.44}$$

$$\tilde{P}_{H,t} - \left[(1-\mu) + \mu S_t^{1-\nu}\right]^{\frac{-1}{1-\nu}} = 0, \tag{D.45}$$

$$Y_t - (1-\mu)\left(\tilde{P}_{H,t}\right)^{-\nu} C_t - \mu^* S_t^{\nu} C_t^* = 0, \tag{D.46}$$

$$(C_t - \chi C_{t-1}) - \exp(\varepsilon_t^{c*} - \varepsilon_t^c)\left(\frac{(C_t-\chi C_{t-1})^{\nu}(1-N_t)^{1-\nu}}{(C_t^*-\chi^* C_{t-1}^*)^{\nu *}(1-N_t^*)^{1-\nu^*}}\right)^{1-\gamma}\left(\frac{C^*-\chi^* C_{t-1}^*}{G}\right)\tilde{P}_{H,t} S_t = 0, \tag{D.47}$$

$$Y_t - A_t N_t^{1-\alpha} = 0, \tag{D.48}$$

$$\begin{aligned} &-\Omega_t + \left(\tfrac{1-\nu}{\nu}\right)\left(\tfrac{C_t-\chi C_{t-1}}{1-N_t}\right)\left(\tfrac{1}{A_t}\right)^{\frac{1}{1-\alpha}}\left(\tfrac{\theta}{\theta-1}\right)\left(\tfrac{1}{1-\alpha}\right) - \left(\tfrac{\phi}{\theta-1}\right)\pi_{H,t} Y_t\left(\pi_{H,t} - \tilde{\pi}_t\right) + \\ &\exp(-\varepsilon_t^c)\left(\tfrac{\phi}{\theta-1}\right)\beta\left(\tfrac{V_{t+1}^{1-\gamma}}{\mathrm{E}_t V_{t+1}}\right)^{\frac{\rho-\gamma}{1-\gamma}}\left(\tfrac{(C_{t+1}-\chi C_t)^{\nu}(1-N_{t+1})^{1-\nu}}{(C_t-\chi C_{t-1})^{\nu}(1-N_t)^{1-\nu}}\right)^{1-\gamma}\left(\tfrac{C_t-\chi C_{t-1}}{C_{t+1}-\chi C_t}\right)\pi_{t+1} C_{t+1}\left(\pi_{t+1} - \tilde{\pi}_t\right) = 0, \end{aligned} \tag{D.49}$$

$$\tilde{\pi}_t - (\pi_{t-1})^{\xi}(\bar{\pi})^{1-\xi} = 0, \tag{D.50}$$

$$R_t - \exp(\varepsilon_t^r)\left(\left(\tfrac{\bar{\pi}}{\beta}\right)\left(\tfrac{Y_t}{\bar{Y}}\right)^{\kappa_y}\left(\tfrac{\pi_t}{\bar{\pi}}\right)^{\kappa_\pi}\right)^{1-\rho_r} R_{t-1}^{\rho_r} = 0, \tag{D.51}$$

$$P_t^{NB} - 1 - \frac{\delta^c P_{t+1}^{NB}}{R_t} = 0, \tag{D.52}$$

$$\begin{aligned} P_t^B - 1 - \exp(-\varepsilon_t^c)\delta^c P_{t+1}^B \beta\left(\tfrac{V_{t+1}^{1-\gamma}}{\mathrm{E}_t V_{t+1}}\right)^{\frac{\rho-\gamma}{1-\gamma}}\left(\tfrac{(C_{t+1}-\chi^* C_t)^{\nu}(1-N_{t+1})^{1-\nu}}{(C_t-\chi C_{t-1})^{\nu}(1-N_t)^{1-\nu}}\right)^{1-\gamma} \times \cdots \\ \cdots \times \left(\tfrac{C_t-\chi C_{t-1}}{C_{t+1}-\chi C_t}\right)\left(\tfrac{1}{\pi_{t+1}}\right) = 0, \end{aligned} \tag{D.53}$$

$$Y_t^T - \frac{\delta^c P_t^B}{P^B - 1} = 0, \tag{D.54}$$

$$Y_t^{TN} - \frac{\delta^c P_t^{BN}}{P^{BN} - 1} = 0, \tag{D.55}$$

$$T_t - \frac{Y_t^T}{Y_t^{TN}} = 0, \tag{D.56}$$

$$A_t - A_{t-1}^{\rho}\bar{A}^{1-\rho}\exp(\varepsilon_t^a) = 0, \tag{D.57}$$

$$A_t^* - (A^*)_{t-1}^{\rho^*}(\bar{A}^*)^{1-\rho^*}\exp(\varepsilon_t^{a*}) = 0, \tag{D.58}$$

$$\Omega_t - \Omega_{t-1}^{\rho_\omega}\bar{\Omega}^{1-\rho_\omega}\exp(\varepsilon_t^{\omega}) = 0, \tag{D.59}$$

$$\Omega_t^* - (\Omega_{t-1}^*)^{\rho_\omega^*}(\bar{\Omega}^*)^{1-\rho_\omega^*}\exp(\varepsilon_t^{\omega *}) = 0. \tag{D.60}$$

Table D.13: Domestic Variables

Symbol	Description
V_t	Welfare
Y_t	Output
C_t	Consumption
N_t	Hours worked
π_t	Inflation
$\pi_{H,t}$	Tradable inflation
$\tilde{\pi}_{H,t}$	Tradable inflation index
$\tilde{P}_{H,t}$	Relative price of domestically produced goods
S_t	Terms of trade
R_t	Interest rate
P_t^{NB}	Price of a safe bond
P_t^B	Price of a risky bond
Y_t^{TN}	Yield on a safe bond
Y_t^T	Yield on a risky bond
T_t	Risk premia
A_t	Technology
Ω_t	Cost push shock process
ε_t^a	Technology shock
ε_t^c	Consumption preference shock
ε_t^ω	Cost-push shock

Table D.14: Foreign Variables

Symbol	Description
V_t^*	Welfare
Y_t^*	Output
C_t	Consumption
N^*	Hours worked
π_t^*	Inflation
$\tilde{\pi}_t$	Inflation index
R_t^*	Interest rate
P_t^{NB*}	Price of a safe bond
P_t^{B*}	Price of a risky bond
Y_t^{TN*}	Yield on a safe bond
Y_t^{T*}	Yield on a risky bond
T_t^*	Risk premia
A_t^*	Technology
Ω_t^*	Cost push shock process
ε_t^{a*}	Technology shock
ε_t^{c*}	Consumption preference shock
$\varepsilon_t^{\omega*}$	Cost push shock

Table D.15: Domestic Parameters

Symbol	Description
λ	Weight on disutility of labour
$1-\alpha$	Labour's share of income
χ	Domestic habit parameter
σ	Domestic inverse of the intertemporal EOS
η	Frisch elasticity of labour supply
ν	Elasticity of substitution between domestic and foreign goods
$1-\mu$	Home bias
θ	Probability of adjusting prices optimally
ϑ	Scale parameter
ρ_π	Degree of price indexation
κ_y	Weight on output in the Taylor rule
κ_π	Weight on inflation in the Taylor rule
ρ_a	Persistence term on Technology
ρ_c	Persistence term on Consumption shock
ρ_r	Persistence in Taylor rule

Table D.16: Domestic Parameters

Symbol	Description
λ^*	Weight on disutility of labour
$1-\alpha^*$	Labour's share of income
χ^*	Domestic habit parameter
σ^*	Domestic inverse of the intertemporal EOS
θ	Probability of adjusting prices optimally
ρ_π^*	Degree of price indexation
κ_y^*	Weight on output in the Taylor rule
κ_π^*	Weight on inflation in the Taylor rule
ρ_a^*	Persistence term on Technology
ρ_c^*	Persistence term on Consumption shock
ρ_r^*	Persistence in Taylor rule

References

Andreasen, M. M. (2011). On the effects of rare disasters and uncertainty shocks for risk premia in non-linear DSGE models. *Review of Economic Dynamics*. URL http://dx.doi.org/10.1016/j.red.2011.08.001.

Binning, A. J. (2013). Third-order approximation of dynamic models without the use of tensors. Norges Bank Working Paper 2013/13. URL http://www.norges-bank.no/no/om/publisert/publikasjoner/working-papers/2013/13/.

Galí, J. (2009). *Monetary Policy, Inflation, and the Business Cycle: An Introduction to the New Keynesian Framework*. Princeton University Press. URL http://books.google.no/books?id=idVZotm_ZroC.

Gali, J. & Monacelli, T. (2008). Optimal monetary and fiscal policy in a currency union. *Journal of International Economics*, *76*(1), 116–132. URL http://ideas.repec.org/a/eee/inecon/v76y2008i1p116-132.html.

Gomme, P. & Klein, P. (2011). Second-order approximation of dynamic models without the use of tensors. *Journal of Economic Dynamics and Control*, *35*(4), 604–615. URL http://ideas.repec.org/a/eee/dyncon/v35y2011i4p604-615.html.

Johnson, W. P. (2002). The curious history of Faá di Brunos formula. *Amer. Math. Monthly*, *109*, 217–234.

Kågström, B. & Poromaa, P. (1996). Lapack-style algorithms and software for solving the generalized sylvester equation and estimating the separation between regular matrix pairs. *ACM Trans. Math. Softw.*, *22*(1), 78–103. URL http://doi.acm.org/10.1145/225545.225552.

Kamenik, O. (2005). Solving sdge models: A new algorithm for the sylvester equation. Working Papers 2005/10, Czech National Bank, Research Department. URL http://ideas.repec.org/p/cnb/wpaper/2005-10.html.

Kim, J. & Ruge-Murcia, F. J. (2011). Monetary policy when wages are downwardly rigid: Friedman meets tobin. *Journal of Economic Dynamics and Control*, *35*(12), 2064–2077. URL http://ideas.repec.org/a/eee/dyncon/v35y2011i12p2064-2077.html.

Klein, P. (2000). Using the generalized schur form to solve a multivariate linear rational expectations model. *Journal of Economic Dynamics and Control*, *24*(10), 1405–1423. URL http://ideas.repec.org/a/eee/dyncon/v24y2000i10p1405-1423.html.

Magnus, J. & Neudecker, H. (1999). *Matrix differential calculus with applications in statistics and econometrics*. Wiley series in probability and statistics, John Wiley. URL http://books.google.com.au/books?id=0CXXdKKiIpQC.

Martin, C. D. M. & Van Loan, C. F. (2006). Shifted kronecker product systems. *SIAM J. Matrix Analysis Applications*, *29*(1), 184–198.

Rice, J. (2007). *Mathematical Statistics and Data Analysis*. No. p. 3 in Advanced series, Brooks/Cole CENGAGE Learning. URL http://books.google.co.uk/books?id=EKA-yeX2GVgC.

Ruge-Murcia, F. J. (2010). Estimating nonlinear dsge models by the simulated method of moments. Working Paper Series 49_10, Rimini Centre for Economic Analysis. URL http://ideas.repec.org/p/rim/rimwps/49_10.html.

Schmitt-Grohe, 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 http://ideas.repec.org/a/eee/dyncon/v28y2004i4p755-775.html.