2013 | 13

Working Paper

Monetary Policy

Third-order approximation of dynamic models without the use of tensors

Andrew Binning

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 978-82-7553-752-0 (online)

Third-order approximation of dynamic models without the use of tensors

Andrew Binning[1,2]

17 May 2013

Policy and Analysis Department, Norges Bank, Oslo, Norway

Abstract

I outline a new method for finding third-order accurate solutions to dynamic general equilibrium models. I extend the Gomme & Klein (2011) solution for second-order approximations without using tensors, to a third-order. In particular I derive a third-order matrix chain rule and use this to solve the third-order approximation. My solution method is easier to understand and code-up, and faster to implement in Matlab. I provide Matlab code and demonstrate my solution method with a simple RBC model. The resulting code is up to 80 times faster than Matlab code using tensor notation.

Keywords: Solving dynamic models, third-order approximation, third-order matrix chain rule

1. Introduction

Non-linear methods for solving DSGE models have become increasingly popular in recent years. Perturbation methods have become particularly popular due to their relative ease of implementation and their ability to be used with medium and even large scale models. Perturbation methods are now widely available in many software packages and as standalone routines.[3] Attention has shifted from second-order to third-order approximations with Van Binsbergen et al. (2010) showing that third-order approximations are necessary to capture time varying shifts in risk premia. Most of the software and routines currently available that

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 Andreassen, Paolo Gelain, Paul Klein, Junior Maih, Martin Seneca, participants at the 18th International Conference of Computing in Economics and Finance and seminar participants at the Norges Bank for their useful comments. All remaining errors are my own.

[3]Examples of applications include Dynare (see Juillard, 2003), Dynare++ (see Kamenik, 2011), Perturbation AIM (see Swanson et al., 2006) and codes by Schmitt-Grohe & Uribe (2004), Andreasen (2011), Ruge-Murcia (2010) and Gomme & Klein (2011).

solve for third-order approximations use tensor notation.[4] Tensor notation can be difficult to read, difficult to code and in some cases maybe slow to implement. Gomme & Klein (2011) show, using the Magnus & Neudecker (1999) definition of a Hessian matrix, how to solve a second-order approximation without tensors. In this paper I extend their method to a third-order approximation by deriving a third-order matrix chain rule that gives a more efficient representation of the problem. Because the third-order matrix chain rule is linear in the unknown coefficients it is straight forward to solve for the unknown third-order coefficients. I also provide Matlab code for my solution method. The paper is set out as follows; I begin by covering some preliminaries in section 2, in section 3 I present a third-order matrix chain rule, and in section 4 I outline the matrix algebra required to find the solution. In section 5 I demonstrate my method by applying it to a simple RBC model, before I conclude in section 6.

2. Preliminaries

Following Schmitt-Grohe & Uribe (2004) a generic DSGE model can be written in the form

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

where x_t is an $nx \times 1$ vector of predetermined variables, y_t is an $ny \times 1$ vector of 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) the solution of the model will take the form:

$$y_t = g(x_t, \sigma), \tag{2}$$

$$x_{t+1} = h(x_t, \sigma) + \sigma\varepsilon_{t+1}, \tag{3}$$

where g maps $\mathbb{R}^{nx}$ into $\mathbb{R}^{ny}$ and h maps $\mathbb{R}^{nx}$ into $\mathbb{R}^{nx}$. The scalar $\sigma \geq 0$ is known as the perturbation parameter and ε_{t+1} is an $nx \times 1$ vector of shocks. Typically the functions g and h are unknown, do not have exact analytic forms and are highly non-linear. One common strategy to find an approximate solution to the model is to take a Taylor series approximation around the non-stochastic steady state. As mentioned in the introduction, it has become increasingly popular to take a third-order approximation of the policy functions thus allowing for the effects of time varying risk and also the incorporation of skewed shocks. Following such a strategy and deriving a third-order Taylor series approximation of the policy functions, g and h, would result in the system of equations

$$\begin{aligned} y_t = g_x x_t + \tfrac{1}{2}\sigma^2 g_{\sigma\sigma} + \tfrac{1}{2}\left(\underset{ny\times ny}{I} \otimes x_t'\right) g_{xx} x_t \\ + \tfrac{1}{6}\sigma^3 g_{\sigma\sigma\sigma} + \tfrac{1}{2}\sigma^2\left(\underset{ny\times ny}{I} \otimes x_t'\right) g_{\sigma\sigma x} + \tfrac{1}{6}\left(\underset{ny\times ny}{I} \otimes x_t' \otimes x_t'\right) g_{xxx} x_t, \end{aligned} \tag{4}$$

[4] See Lan & Meyer-Gohde (2011) and Chen & Zadrozny (2003) for other examples of matrix based solutions for solving non-linear DSGE models.

and

$$x_{t+1} = h_x x_t + \tfrac{1}{2}\sigma^2 h_{\sigma\sigma} + \tfrac{1}{2}\left(\underset{nx\times nx}{I} \otimes x_t'\right) h_{xx} x_t + \tfrac{1}{6}\sigma^3 h_{\sigma\sigma\sigma} + \tfrac{1}{2}\sigma^2 \left(\underset{nx\times nx}{I} \otimes x_t'\right) h_{\sigma\sigma x} + \tfrac{1}{6}\left(\underset{nx\times nx}{I} \otimes x_t' \otimes x_t'\right) h_{xxx} x_t + \sigma\varepsilon_{t+1}, \quad (5)$$

where g_x and h_x are the partial derivatives of g and h with respect to x_t evaluated at the non-stochastic steady state, such that

$$\underset{ny\times nx}{g_x} = \begin{bmatrix} g_{1,x_1} & \cdots & g_{1,x_{nx}} \\ \vdots & & \vdots \\ g_{ny,x_1} & \cdots & g_{ny,x_{nx}} \end{bmatrix}, \quad \underset{nx\times nx}{h_x} = \begin{bmatrix} h_{1,x_1} & \cdots & h_{1,x_{nx}} \\ \vdots & & \vdots \\ h_{nx,x_1} & \cdots & h_{nx,x_{nx}} \end{bmatrix},$$

with g_i representing the policy function for the ith non-predetermined variable, and h_i representing the policy function for the ith predetermined variable. It then follows that $g_{i,x_j} = \frac{\partial g_i(x_t,\sigma)}{\partial x_{j,t}} |_{x_t=x_{ss},\,\sigma=0}$ and $h_{i,x_j} = \frac{\partial h_i(x_t,\sigma)}{\partial x_{j,t}} |_{x_t=x_{ss},\,\sigma=0}$. These are the coefficient matrices for the first-order approximate solution. Schmitt-Grohe & Uribe (2004) show that g_σ, h_σ are equal to zero when evaluated at the non-stochastic steady state.

The terms: g_{xx}, h_{xx}, and $g_{\sigma\sigma}$ and $h_{\sigma\sigma}$, are the second derivatives of g and h with respect to x and σ evaluated at the non-stochastic steady state,

$$\underset{ny.nx\times nx}{g_{xx}} = \begin{bmatrix} g_{1,x_1x_1} & \cdots & g_{1,x_{nx}x_1} \\ \vdots & & \vdots \\ g_{1,x_1x_{nx}} & \cdots & g_{1,x_{nx}x_{nx}} \\ g_{2,x_1x_1} & \cdots & g_{2,x_{nx}x_1} \\ \vdots & & \vdots \\ g_{ny,x_1x_{nx}} & \cdots & g_{ny,x_{nx}x_{nx}} \end{bmatrix}, \quad \underset{nx^2\times nx}{h_{xx}} = \begin{bmatrix} h_{1,x_1x_1} & \cdots & h_{1,x_{nx}x_1} \\ \vdots & & \vdots \\ h_{1,x_1x_{nx}} & \cdots & h_{1,x_{nx}x_{nx}} \\ h_{2,x_1x_1} & \cdots & h_{2,x_{nx}x_1} \\ \vdots & & \vdots \\ h_{nx,x_1x_{nx}} & \cdots & h_{nx,x_{nx}x_{nx}} \end{bmatrix},$$

$$\underset{ny\times 1}{g_{\sigma\sigma}} = \begin{bmatrix} g_{1,\sigma\sigma} \\ \vdots \\ g_{ny,\sigma\sigma} \end{bmatrix}, \quad \underset{nx\times 1}{h_{\sigma\sigma}} = \begin{bmatrix} h_{1,\sigma\sigma} \\ \vdots \\ h_{nx,\sigma\sigma} \end{bmatrix},$$

with

$$g_{i,x_jx_k} = \frac{\partial^2 g_i(x_t,\sigma)}{\partial x_{j,t}\partial x_{k,t}} |_{x_t=x_{ss},\,\sigma=0}, \quad h_{i,x_jx_k} = \frac{\partial^2 h_i(x_t,\sigma)}{\partial x_{j,t}\partial x_{k,t}} |_{x_t=x_{ss},\,\sigma=0},$$
$$g_{i,\sigma\sigma} = \frac{\partial^2 g_i(x_t,\sigma)}{\partial^2 \sigma} |_{x_t=x_{ss},\,\sigma=0}, \quad h_{i,\sigma\sigma} = \frac{\partial^2 h_i(x_t,\sigma)}{\partial^2 \sigma} |_{x_t=x_{ss},\,\sigma=0}.$$

These are the coefficient matrices in the second-order approximation. Schmitt-Grohe & Uribe (2004) show that $g_{\sigma x}$ and $h_{\sigma x}$ are equal to zero when evaluated at the non-stochastic steady state.

The terms g_{xxx}, h_{xxx}, $g_{\sigma\sigma x}$, $h_{\sigma\sigma x}$ $g_{\sigma\sigma\sigma}$ and $h_{\sigma\sigma\sigma}$ are the third derivatives of g and h with respect to x_t and σ evaluated at the non-stochastic steady state,

$$\underset{ny.nx^2\times nx}{g_{xxx}} = \begin{bmatrix} g_{1,x_1x_1x_1} & \cdots & g_{1,x_{nx}x_1x_1} \\ \vdots & & \vdots \\ g_{1,x_1x_{nx}x_1} & \cdots & g_{1,x_{nx}x_{nx}x_1} \\ g_{1,x_1x_1x_2} & \cdots & g_{1,x_{nx}x_1x_2} \\ \vdots & & \vdots \\ g_{1,x_1x_{nx}x_{nx}} & \cdots & g_{1,x_{nx}x_{nx}x_{nx}} \\ g_{2,x_1x_1x_1} & \cdots & g_{2,x_{nx}x_1x_1} \\ \vdots & & \vdots \\ g_{ny,x_1x_{nx}x_{nx}} & \cdots & g_{ny,x_{nx}x_{nx}x_{nx}} \end{bmatrix}, \quad \underset{nx^3\times nx}{h_{xxx}} = \begin{bmatrix} h_{1,x_1x_1x_1} & \cdots & h_{1,x_{nx}x_1x_1} \\ \vdots & & \vdots \\ h_{1,x_1x_{nx}x_1} & \cdots & h_{1,x_{nx}x_{nx}x_1} \\ h_{1,x_1x_1x_2} & \cdots & h_{1,x_{nx}x_1x_2} \\ \vdots & & \vdots \\ h_{1,x_1x_{nx}x_{nx}} & \cdots & h_{1,x_{nx}x_{nx}x_{nx}} \\ h_{2,x_1x_1x_1} & \cdots & h_{2,x_{nx}x_1x_1} \\ \vdots & & \vdots \\ h_{nx,x_1x_{nx}x_{nx}} & \cdots & h_{nx,x_{nx}x_{nx}x_{nx}} \end{bmatrix},$$

$$\underset{ny.nx\times 1}{g_{\sigma\sigma x}} = \begin{bmatrix} g_{1,\sigma\sigma,x_1} \\ \vdots \\ g_{1,\sigma\sigma,x_{nx}} \\ g_{2,\sigma\sigma,x_1} \\ \vdots \\ g_{ny,\sigma\sigma,x_{nx}} \end{bmatrix}, \quad \underset{nx^2\times 1}{h_{\sigma\sigma x}} = \begin{bmatrix} h_{1,\sigma\sigma,x_1} \\ \vdots \\ h_{1,\sigma\sigma,x_{nx}} \\ h_{2,\sigma\sigma,x_1} \\ \vdots \\ h_{nx,\sigma\sigma,x_{nx}} \end{bmatrix},$$

$$\underset{ny\times 1}{g_{\sigma\sigma\sigma}} = \begin{bmatrix} g_{1,\sigma\sigma\sigma} \\ \vdots \\ g_{ny,\sigma\sigma\sigma} \end{bmatrix}, \quad \underset{nx\times 1}{h_{\sigma\sigma\sigma}} = \begin{bmatrix} h_{1,\sigma\sigma\sigma} \\ \vdots \\ h_{nx,\sigma\sigma\sigma} \end{bmatrix},$$

with

$$g_{i,x_jx_kx_l} = \frac{\partial^3 g_i(x_t,\sigma)}{\partial x_{j,t}\partial x_{k,t}\partial x_{l,t}} |_{x_t=x_{ss},\,\sigma=0}, \quad h_{i,x_jx_k,x_l} = \frac{\partial^3 h_i(x_t,\sigma)}{\partial x_{j,t}\partial x_{k,t}\partial x_{l,t}} |_{x_t=x_{ss},\,\sigma=0},$$
$$g_{i,\sigma\sigma x_j} = \frac{\partial^3 g_i(x_t,\sigma)}{\partial^2\sigma\partial x_{l,t}} |_{x_t=x_{ss},\,\sigma=0}, \quad h_{i,\sigma\sigma x_j} = \frac{\partial^3 h_i(x_t,\sigma)}{\partial^2\sigma\partial x_{j,t}} |_{x_t=x_{ss},\,\sigma=0},$$
$$g_{i,\sigma\sigma\sigma} = \frac{\partial^3 g_i(x_t,\sigma)}{\partial^3\sigma} |_{x_t=x_{ss},\,\sigma=0}, \quad h_{i,\sigma\sigma\sigma} = \frac{\partial^3 h_i(x_t,\sigma)}{\partial^3\sigma} |_{x_t=x_{ss},\,\sigma=0.}$$

These are the coefficient matrices in the third-order approximation. Andreasen (2011) shows that $g_{xx\sigma}$ and $h_{xx\sigma}$ are zero when evaluated at the non-stochastic steady state. The coefficients $g_{\sigma\sigma\sigma}$ and $h_{\sigma\sigma\sigma}$ will be non-zero if the third moment of the shocks is non-zero.

Because the policy functions (equations 2 and 3) are unknown, I have to use the implicit function theorem to find the unknown coefficients in the Taylor series expansion around the non-stochastic steady state. To do this I substitute equations (2) and (3) into equation (1) to get

$$E_t\left(f\left(h(x_t,\sigma) + \sigma\varepsilon_{t+1}, g(h(x_t) + \sigma\varepsilon_{t+1},\sigma), x_t, g(x_t,\sigma)\right)\right) = 0. \tag{6}$$

I then proceed to find the third-order approximation as follows:

i) I begin by finding the first-order approximation of the policy functions g and h. This can be done using Klein's algorithm (see Klein, 2000) for example.[5]

ii) The first-order approximation can then be used to find the second-order approximation of the model. Taking the second derivative of f with respect to $x_{i,t}$ and $x_{j,t}$ $i, j = 1, \cdots, nx$, and then substituting in g_x and h_x (the solution to the first-order approximation) results in a system that is linear in g_{xx} and h_{xx}. This is done more efficiently using the second-order matrix chain rule of Magnus & Neudecker (1999) as is done in Gomme & Klein (2011). The unknown coefficient matrices can then be found as the solution to a system of linear equations.

iii) The first-order approximation and the second-order approximation can then be used to find the third-order approximation. Taking derivatives of f with respect to $x_{i,t}$, $x_{j,t}$ and $x_{k,t}$ for $i, j, k = 1, \cdots, nx$, and then substituting in the first and second-order solutions results in a system that is linear in g_{xxx} and h_{xxx}. In this paper, I develop a third-order matrix chain rule that gives a more efficient representation of this problem. As before, the unknown coefficient matrices can be found as the solution to a system of linear equations.

Similar steps can be followed to find the unknown coefficients $g_{\sigma\sigma x}$, $h_{\sigma\sigma x}$, $g_{\sigma\sigma\sigma}$ and $h_{\sigma\sigma\sigma}$.

3. A third-order matrix chain rule

In this section I present a third-order matrix chain rule that is a natural extension of Magnus and Neudecker's second-order matrix chain rule (see Magnus & Neudecker, 1999). This will prove a useful and efficient alternative to the tensor notation that is commonly used. I begin by defining some function g that is an n-ary function of f, where f is an m-ary function of $\mathbf{x}$ so that

$$\mathsf{y} = \mathsf{g}\left(\mathsf{f}^1(\mathbf{x}), \cdots, \mathsf{f}^n(\mathbf{x})\right) \tag{7}$$

where the superscripts denote each f function and $\mathbf{x}$ is a vector of the variables x_i, such that

$$\mathbf{x} = [\mathsf{x}_1, \cdots, \mathsf{x}_m].$$

[5]Because the first derivative of f with respect to x_t results in a quadratic function, a solution method like Klein's algorithm can be used to keep the solution with stable eigenvalues.

By Fáa di Bruno's formula, the third derivative of y with respect to the ith, jth and kth elements in $\mathbf{x}$ is

$$\begin{aligned}\frac{\partial^3 \mathsf{y}}{\partial \mathsf{x}_i \partial \mathsf{x}_j \partial \mathsf{x}_k} \quad = \quad & \sum_{a=1}^{n}\sum_{b=1}^{n}\sum_{c=1}^{n} \frac{\partial^3 \mathsf{g}}{\partial \mathsf{f}^a \partial \mathsf{f}^b \partial \mathsf{f}^c}\left(\frac{\partial \mathsf{f}^a}{\partial \mathsf{x}_i}\right)\left(\frac{\partial \mathsf{f}^b}{\partial \mathsf{x}_k}\right)\left(\frac{\partial \mathsf{f}^c}{\partial \mathsf{x}_j}\right) + \\ & \sum_{a=1}^{n}\sum_{b=1}^{n} \frac{\partial^2 \mathsf{g}}{\partial \mathsf{f}^a \partial \mathsf{f}^b}\left(\frac{\partial^2 \mathsf{f}^a}{\partial \mathsf{x}_i \partial \mathsf{x}_j}\right)\left(\frac{\partial \mathsf{f}^b}{\partial \mathsf{x}_k}\right) + \\ & \sum_{a=1}^{n}\sum_{b=1}^{n} \frac{\partial^2 \mathsf{g}}{\partial \mathsf{f}^a \partial \mathsf{f}^b}\left(\frac{\partial^2 \mathsf{f}^a}{\partial \mathsf{x}_i \partial \mathsf{x}_k}\right)\left(\frac{\partial \mathsf{f}^b}{\partial \mathsf{x}_j}\right) + \\ & \sum_{a=1}^{n}\sum_{b=1}^{n} \frac{\partial \mathsf{g}}{\partial \mathsf{f}^a \partial \mathsf{f}^b}\left(\frac{\partial^2 \mathsf{f}^a}{\partial \mathsf{x}_j \partial \mathsf{x}_k}\right)\left(\frac{\partial \mathsf{f}^b}{\partial \mathsf{x}_i}\right) + \\ & \sum_{a=1}^{n} \frac{\partial \mathsf{g}}{\partial \mathsf{f}^a}\left(\frac{\partial^3 \mathsf{f}^a}{\partial \mathsf{x}_i \partial \mathsf{x}_j \partial \mathsf{x}_k}\right),\end{aligned} \tag{8}$$

for any $i, j, k = 1, \ldots, m$ and $a, b, c = 1, \ldots, n$. This can be written more compactly as

$$\mathsf{y}_{i,j,k} = \sum_{a=1}^{n}\sum_{b=1}^{n}\sum_{c=1}^{n} \mathsf{g}_{a,b,c}\mathsf{f}_i^a\mathsf{f}_k^b\mathsf{f}_j^c + \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}_{i,j}^a\mathsf{f}_k^b + \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}_{i,k}^a\mathsf{f}_j^b + \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}_{j,k}^a\mathsf{f}_i^b + \sum_{a=1}^{n} \mathsf{g}_a\mathsf{f}_{i,j,k}^a, \tag{9}$$

I let S be the $m^2 \times m$ matrix of all possible combinations of the third-derivatives of y with respect to each element x_i in $\mathbf{x}$. This has the form

$$\underset{m^2 \times m}{\mathsf{S}} = \begin{bmatrix} \tilde{\mathsf{S}}_1 \\ \vdots \\ \tilde{\mathsf{S}}_k \\ \vdots \\ \tilde{\mathsf{S}}_m \end{bmatrix}, \quad \text{where} \quad \underset{m \times m}{\tilde{\mathsf{S}}_k} = \begin{bmatrix} \mathsf{y}_{1,1,k} & \cdots & \mathsf{y}_{m,1,k} \\ \vdots & \ddots & \\ \mathsf{y}_{1,m,k} & \cdots & \mathsf{y}_{m,m,k} \end{bmatrix}. \tag{10}$$

The element in the rth row and cth column of S is denoted by $\mathsf{s}_{r,c}$. Alternatively I can use $\mathsf{s}_{j+m(k-1),i}$ to refer to the element in the $j+m(k-1)$th row and the ith column of S where as before $i, j, k = 1, \ldots, m$. This alternative indexation allows the coordinates of an element in S to be matched to the derivative in that position. For example; $\mathsf{y}_{i,j,k} = \mathsf{s}_{j+m(k-1),i}$. The new indexation will be useful for constructing a proof of the chain rule.

Given the definition of S, I can now describe the third-order matrix chain rule consistent with the derivatives in each element in S. Before I do this, I need to define some additional matrices that will be used in the chain rule.

I begin with the gradient matrix for the function f, which I use D to denote, so that

$$\underset{n \times m}{\mathsf{D}} = \begin{bmatrix} \mathsf{f}_1^1 & \cdots & \mathsf{f}_m^1 \\ \vdots & & \vdots \\ \mathsf{f}_1^n & \cdots & \mathsf{f}_m^n \end{bmatrix}. \tag{11}$$

It follows from this definition that $\mathsf{f}_i^a = \mathsf{d}_{a,i}$ for $i = 1, \ldots, m$ and $a = 1, \ldots, n$, where $\mathsf{d}_{a,i}$ is the element in the ath row and the ith column of D. As part of the chain rule I need to perform some transformations on some of the gradient matrices for the f function. This ensures that the gradient with respect to the appropriate x_i, x_j and x_k is used to reconstruct each element of S.[6] I let Q represent one such transformation

$$\underset{n.m\times m^2}{\mathsf{Q}} = \begin{bmatrix} \underset{m\times m}{I} \otimes \mathsf{D}_1 \\ \vdots \\ \underset{m\times m}{I} \otimes \mathsf{D}_n \end{bmatrix} = \begin{bmatrix} \mathsf{f}_1^1 & \mathsf{f}_2^1 & \cdots & \mathsf{f}_m^1 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \mathsf{f}_1^1 & \mathsf{f}_2^1 & \cdots & \mathsf{f}_m^1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & & & & & & & & \vdots & & & & \\ 0 & 0 & & & & & & & \cdots & \mathsf{f}_1^1 & \mathsf{f}_2^1 & \cdots & \mathsf{f}_m^1 \\ \mathsf{f}_1^2 & \mathsf{f}_2^2 & \cdots & \mathsf{f}_m^2 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \mathsf{f}_1^2 & \mathsf{f}_2^2 & \cdots & \mathsf{f}_m^2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & & & & & & & & \vdots & & & & \\ 0 & 0 & & & & & & & \cdots & \mathsf{f}_1^2 & \mathsf{f}_2^2 & \cdots & \mathsf{f}_m^2 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots \\ \mathsf{f}_1^n & \mathsf{f}_2^n & \cdots & \mathsf{f}_m^n & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \mathsf{f}_1^n & \mathsf{f}_2^n & \cdots & \mathsf{f}_m^n & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & & & & & & & & \vdots & & & & \\ 0 & 0 & & & & & & & \cdots & \mathsf{f}_1^n & \mathsf{f}_2^n & \cdots & \mathsf{f}_m^n \end{bmatrix}, \tag{12}$$

where D_i is the ith row of the matrix D. It then follows from this definition that

$$\mathsf{q}_{k+m(b-1),j+m(k-1)} = \mathsf{f}_j^b$$

for $j, k = 1, \ldots, m$ and $b = 1, \ldots, n$, where $\mathsf{q}_{k+m(b-1),j+m(k-1)}$ is the element in the $k + m(b-1)$th row and the $j + m(k-1)$th column of Q.

The Hessian of f is represented by

$$\underset{n.m\times m}{\mathsf{V}} = \begin{bmatrix} \tilde{\mathsf{V}}_1 \\ \vdots \\ \tilde{\mathsf{V}}_a \\ \vdots \\ \tilde{\mathsf{V}}_n \end{bmatrix}, \quad \text{where} \quad \underset{m\times m}{\tilde{\mathsf{V}}_a} = \begin{bmatrix} \mathsf{f}_{1,1}^a & \cdots & \mathsf{f}_{m,1}^a \\ \vdots & & \vdots \\ \mathsf{f}_{1,m}^a & \cdots & \mathsf{f}_{m,m}^a \end{bmatrix} \tag{13}$$

with $\mathsf{f}_{i,j}^a = \mathsf{v}_{j+m(a-1),i}$ for $a = 1, \ldots, n$ and $i, j = 1, \ldots, m$, and $\mathsf{v}_{j+m(a-1),i}$ is the element in the $j + m(a-1)$th row and the ith column of the matrix V.

The chain rule requires the appropriate second derivatives of f to be used at each step when constructing the elements in S. As a consequence some rearrangements need to be

[6]This will be demonstrated in the proof of Theorem for this chain rule.

performed on the Hessian of f. I let P denote one such rearrangement

$$\underset{n\times m^2}{\mathsf{P}} = \left[\begin{array}{ccccc} \tilde{\mathsf{P}}_1 & \cdots & \tilde{\mathsf{P}}_j & \cdots & \tilde{\mathsf{P}}_m \end{array}\right], \quad \text{where} \quad \underset{n\times m}{\tilde{\mathsf{P}}_j} = \left[\begin{array}{ccc} \mathsf{f}^1_{1,j} & \cdots & \mathsf{f}^1_{m,j} \\ \vdots & & \vdots \\ \mathsf{f}^n_{1,j} & \cdots & \mathsf{f}^n_{m,j} \end{array}\right] \tag{14}$$

so that $\mathsf{f}^a_{i,j} = \mathsf{p}_{a,i+m(j-1)}$ for $a = 1,\ldots,n$ and $i,j = 1,\ldots,m$, where $\mathsf{p}_{a,i+m(j-1)}$ is the element in the ath row and the $i+m(j-1)$th column of P.

The matrix T contains the third derivatives of f

$$\underset{n.m^2\times m}{\mathsf{T}} = \left[\begin{array}{c} \tilde{\mathsf{T}}_1 \\ \vdots \\ \tilde{\mathsf{T}}_a \\ \vdots \\ \tilde{\mathsf{T}}_n \end{array}\right], \quad \text{where} \quad \underset{m^2\times m}{\tilde{\mathsf{T}}_a} = \left[\begin{array}{c} \hat{\mathsf{T}}^a_1 \\ \vdots \\ \hat{\mathsf{T}}^a_k \\ \vdots \\ \hat{\mathsf{T}}^a_m \end{array}\right], \quad \text{and} \quad \underset{m\times m}{\hat{\mathsf{T}}^a_k} = \left[\begin{array}{ccc} \mathsf{f}^a_{1,1,k} & \cdots & \mathsf{f}^a_{m,1,k} \\ \vdots & & \vdots \\ \mathsf{f}^a_{1,m,k} & \cdots & \mathsf{f}^a_{m,m,k} \end{array}\right]. \tag{15}$$

It follows from the definition that $\mathsf{f}^a_{i,j,k} = \mathsf{t}_{j+m(k-1)+m^2(a-1),i}$ for $a = 1,\ldots,n$ and $i,j,k = 1,\ldots,m$, where $\mathsf{t}_{j+m(k-1)+m^2(a-1),i}$ is the element in $j+m(k-1)+m^2(a-1)$th row and the ith column of T.

I define the gradient vector for the g function

$$\underset{1\times n}{\mathsf{R}} = [\mathsf{g}_1, \cdots, \mathsf{g}_n] \tag{16}$$

so that $\mathsf{g}_a = \mathsf{r}_{1,a}$ for $a = 1,\ldots,n$, where $\mathsf{r}_{1,a}$ is the ath entry in the row vector R. The Hessian of the function g has the form

$$\underset{n\times n}{\mathsf{W}} = \left[\begin{array}{ccc} \mathsf{g}_{1,1} & \cdots & \mathsf{g}_{n,1} \\ \vdots & & \vdots \\ \mathsf{g}_{1,n} & \cdots & \mathsf{g}_{n,n} \end{array}\right], \tag{17}$$

where $\mathsf{g}_{a,b} = \mathsf{w}_{a,b}$ for $a,b = 1,\ldots,n$, and $\mathsf{w}_{a,b}$ is the element in the ath row and the bth column of W. The matrix Z contains the third derivatives of the g function

$$\underset{n^2\times n}{\mathsf{Z}} = \left[\begin{array}{c} \tilde{\mathsf{Z}}_1 \\ \vdots \\ \tilde{\mathsf{Z}}_c \\ \vdots \\ \tilde{\mathsf{Z}}_n \end{array}\right], \quad \text{where} \quad \underset{n\times n}{\tilde{\mathsf{Z}}_c} = \left[\begin{array}{ccc} \mathsf{g}_{1,1,c} & \cdots & \mathsf{g}_{n,1,c} \\ \vdots & & \vdots \\ \mathsf{g}_{1,n,c} & \cdots & \mathsf{g}_{n,n,c} \end{array}\right], \tag{18}$$

which implies $\mathsf{g}_{a,b,c} = \mathsf{z}_{b+n(c-1),a}$ for $a,b,c = 1,\ldots,n$, where $\mathsf{z}_{b+n(c-1),a}$ is the element in the $b+n(c-1)$th row and the ath column of Z.

Given the definitions of S, D, Z, P, W, V, Q, R and T, I present a Theorem for the third-order matrix chain rule:

Theorem 1. *The third-order matrix chain rule for* $\mathsf{y} = \mathsf{g}(\mathsf{f}(\mathbf{x}))$*, consistent with* S*, takes the form*

$$\mathsf{S} = (\mathsf{D}' \otimes \mathsf{D}')\, \mathsf{Z}\mathsf{D} + \mathsf{P}'\mathsf{W}\mathsf{D} + \left(\mathsf{D}' \otimes \underset{m\times m}{I}\right)\left(\mathsf{W} \otimes \underset{m\times m}{I}\right)\mathsf{V} + \mathsf{Q}'\left(\mathsf{W} \otimes \underset{m\times m}{I}\right)\mathsf{V} + \left(\mathsf{R} \otimes \underset{m^2\times m^2}{I}\right)\mathsf{T}. \quad (19)$$

Proof See Appendix A. ■

4. Third-order approximation

In this section I apply the third-order matrix chain rule (from Theorem 1) to find: g_{xxx}, h_{xxx}, $g_{\sigma\sigma x}$, $h_{\sigma\sigma x}$, $g_{\sigma\sigma\sigma}$ and $h_{\sigma\sigma\sigma}$, the matrices required in a third-order approximation of the policy functions. I begin with the solution of g_{xxx} and h_{xxx} because g_{xxx} is required for the solutions of $g_{\sigma\sigma x}$, $h_{\sigma\sigma x}$, $g_{\sigma\sigma\sigma}$ and $h_{\sigma\sigma\sigma}$.

4.1. Solving for g_{xxx} and h_{xxx}

Before outlining how the third-order matrix chain rule can be applied to find the third-order approximation, I define some additional matrices used in the chain rule.

4.1.1. Matrix Definitions

As was mentioned in section 3, some transformations of the gradient functions (in this case for the policy function) are required to ensure that the correct derivative is used when constructing each element of the matrix chain rule. One such transformation is given by

$$\underset{nx^2\times nx^2}{h_x^*} = \begin{bmatrix} \underset{nx\times nx}{I} \otimes h_{1,x} \\ \vdots \\ \underset{nx\times nx}{I} \otimes h_{nx,x} \end{bmatrix},$$

where $h_{i,x}$ is the ith row of the h_x matrix so that h_x^* is a matrix that consists of the Kronecker product of the $nx \times nx$ identity matrix and each row of h_x. This is the same as the transformation used to construct Q in equation (12).

As is required for the matrix chain rule, some of the Hessian matrices need to be rearranged. Applying these rearrangements to g_{xx} and h_{xx} gives

$$\underset{ny\times nx^2}{g_{xx}^*} = \begin{bmatrix} g_{1,x_1x_1} & \cdots & g_{1,x_{nx}x_{nx}} \\ \vdots & & \vdots \\ g_{ny,x_1x_1} & \cdots & g_{ny,x_{nx}x_{nx}} \end{bmatrix}, \quad \underset{nx\times nx^2}{h_{xx}^*} = \begin{bmatrix} h_{1,x_1x_1} & \cdots & h_{1,x_{nx}x_{nx}} \\ \vdots & & \vdots \\ h_{nx,x_1x_1} & \cdots & h_{nx,x_{nx}x_{nx}} \end{bmatrix}.$$

These follow from the definition of P in equation (14). I let M_x and M_{xx} represent the gradient and Hessian matrices for the policy functions

$$\underset{2n\times nx}{M_x} = \begin{bmatrix} h_x \\ g_x h_x \\ \underset{nx\times nx}{I} \\ g_x \end{bmatrix}, \quad \underset{2n.nx\times nx}{M_{xx}} = \begin{bmatrix} h_{xx} \\ \left(\underset{ny\times ny}{I} \otimes h_x'\right) g_{xx} h_x + \left(g_x \otimes \underset{nx\times nx}{I}\right) h_{xx} \\ \underset{nx^2\times nx}{0} \\ g_{xx} \end{bmatrix}.$$

I apply the required transformations to M_x to get

$$\underset{2n.nx\times nx^2}{M_x^*} = \begin{bmatrix} \underset{nx\times nx}{I} \otimes M_{1,x} \\ \vdots \\ \underset{nx\times nx}{I} \otimes M_{2n,x} \end{bmatrix},$$

where $M_{i,x}$ is the ith row of the M_x matrix so that M_x^* is made up of the Kronecker product of the $nx \times nx$ identity matrix and the rows of M_x. This is the same as the transformation used to construct Q in equation (12). I also need to rearrange the Hessian of the policy functions (M_{xx}), which gives

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

This follows from the definition of P in equation (14). Finally, I define the gradient matrix, Hessian matrix and the matrix of third derivatives for the f function. I let D denote the gradient function

$$\underset{n\times 2n}{D} = \begin{bmatrix} \frac{\partial f_1}{\partial x_{1,t+1}} & \cdots & \frac{\partial f_1}{\partial x_{nx,t+1}} & \frac{\partial f_1}{\partial y_{1,t+1}} & \cdots & \frac{\partial f_1}{\partial y_{ny,t+1}} & \frac{\partial f_1}{\partial x_{1,t}} & \cdots\cdots\cdots & \frac{\partial f_1}{\partial y_{ny,t}} \\ \vdots & & & & & & & & \vdots \\ \frac{\partial f_n}{\partial x_{1,t+1}} & \cdots & \frac{\partial f_n}{\partial x_{nx,t+1}} & \frac{\partial f_n}{\partial y_{1,t+1}} & \cdots & \frac{\partial f_n}{\partial y_{ny,t+1}} & \frac{\partial f_n}{\partial x_{1,t}} & \cdots\cdots\cdots & \frac{\partial f_n}{\partial y_{ny,t}} \end{bmatrix}.$$

The Hessian takes the form

$$\underset{2n^2\times 2n}{H} = \begin{bmatrix} \tilde{H}_1 \\ \vdots \\ \tilde{H}_a \\ \vdots \\ \tilde{H}_n \end{bmatrix}, \quad \text{where} \quad \underset{2n\times 2n}{\tilde{H}_a} = \begin{bmatrix} \frac{\partial^2 f_a}{\partial x_{1,t+1}\partial x_{1,t+1}} & \cdots & \frac{\partial^2 f_a}{\partial y_{ny,t}\partial x_{1,t+1}} \\ \vdots & & \vdots \\ \frac{\partial^2 f_a}{\partial x_{1,t+1}\partial y_{ny,t}} & \cdots & \frac{\partial^2 f_a}{\partial y_{ny,t}\partial y_{ny,t}} \end{bmatrix}.$$

The matrix of third derivatives is given by

$$\underset{4n^3\times 2n}{T} = \begin{bmatrix} \tilde{T}_1 \\ \vdots \\ \tilde{T}_a \\ \vdots \\ \tilde{T}_n \end{bmatrix}, \quad \text{where} \quad \underset{2n^2\times 2n}{\tilde{T}_a} = \begin{bmatrix} \frac{\partial^3 f_a}{\partial x_{1,t+1}\partial x_{1,t+1}\partial x_{1,t+1}} & \cdots & \frac{\partial^3 f_a}{\partial y_{ny,t}\partial x_{1,t+1}\partial x_{1,t+1}} \\ \vdots & & \vdots \\ \frac{\partial^3 f_a}{\partial x_{1,t+1}\partial y_{ny,t}\partial x_{1,t+1}} & \cdots & \frac{\partial^3 f_a}{\partial y_{ny,t}\partial y_{ny,t}\partial x_{1,t+1}} \\ \frac{\partial^3 f_a}{\partial x_{1,t+1}\partial x_{1,t+1}\partial x_{2,t+1}} & \cdots & \frac{\partial^3 f_a}{\partial y_{ny,t}\partial x_{1,t+1}\partial x_{2,t+1}} \\ \vdots & & \vdots \\ \frac{\partial^3 f_a}{\partial x_{1,t+1}\partial y_{ny,t}\partial y_{ny,t}} & \cdots & \frac{\partial^3 f_a}{\partial y_{ny,t}\partial y_{ny,t}\partial y_{ny,t}} \end{bmatrix}.$$

4.1.2. Solution

After solving the first and second-order approximations of the model, I find the third derivatives of equation (6) with respect to all possible combinations of the elements in x_t. I can then substitute the first and second-order derivatives of the policy functions, the gradient matrix, the Hessian matrix and the matrix of third derivatives for the function f (all evaluated at the non-stochastic steady state) into the resulting equations. The unknown third derivatives of the policy function will be the solution to this system of equations. A more efficient approach is to apply Theorem 1 (the third-order matrix chain rule) to equation (6) to get

$$\left(\underset{n\times n}{I} \otimes M_x' \otimes M_x'\right) T M_x + \left(\underset{n\times n}{I} \otimes (M_{xx}^*)'\right) H M_x + \left(\underset{n\times n}{I} \otimes M_x' \otimes \underset{nx\times nx}{I}\right)\left(H \otimes \underset{nx\times nx}{I}\right) M_{xx} +$$
$$\left(\underset{n\times n}{I} \otimes (M_x^*)'\right)\left(H \otimes \underset{nx\times nx}{I}\right) M_{xx} +$$
$$\left(D \otimes \underset{nx^2\times nx^2}{I}\right) \begin{bmatrix} \underset{nx^3\times nx}{h_{xxx}} \\ \left(\underset{ny\times ny}{I} \otimes h_x' \otimes h_x'\right) \underset{ny.nx^2\times nx}{g_{xxx}} h_x + \left(g_x \otimes \underset{nx^2\times nx^2}{I}\right) h_{xxx} + \underset{ny.nx^2\times nx}{K} \\ \underset{nx^3\times nx}{0} \\ g_{xxx} \end{bmatrix} = 0, \tag{20}$$

where

$$K = \left(\underset{ny\times ny}{I} \otimes h_x' \otimes \underset{nx\times nx}{I}\right)\left(g_{xx} \otimes \underset{nx\times nx}{I}\right) h_{xx}$$
$$+ \left(\underset{ny\times ny}{I} \otimes (h_x^*)'\right)\left(g_{xx} \otimes \underset{nx\times nx}{I}\right) h_{xx} + \left(\underset{ny\times ny}{I} \otimes (h_{xx}^*)'\right) g_{xx} h_x.$$

Applying the partition, $D = \left[\underset{n\times nx}{d_1}, \underset{n\times ny}{d_2}, \underset{n\times nx}{d_3}, \underset{n\times ny}{d_4}\right]$, allows me to rearrange equation (20) to get

$$\underset{n.nx^2\times nx}{A} + \left(d_1 \otimes \underset{nx^2\times nx^2}{I}\right) h_{xxx} + \left(d_2 \otimes \underset{nx^2\times nx^2}{I}\right)\left(\underset{ny\times ny}{I} \otimes h_x' \otimes h_x'\right) g_{xxx} h_x \\ + \left(d_2 \otimes \underset{nx^2\times nx^2}{I}\right)\left(g_x \otimes \underset{nx^2\times nx^2}{I}\right) h_{xxx} + \left(d_4 \otimes \underset{nx^2\times nx^2}{I}\right) g_{xxx} = 0, \quad (21)$$

where

$$A = \left(\underset{n\times n}{I} \otimes M_x' \otimes M_x'\right) T M_x + \left(\underset{n\times n}{I} \otimes (M_{xx}^*)'\right) H M_x + \\ \left(\underset{n\times n}{I} \otimes M_x' \otimes \underset{nx\times nx}{I}\right)\left(H \otimes \underset{nx\times nx}{I}\right) M_{xx} + \\ \left(\underset{n\times n}{I} \otimes (M_x^*)'\right)\left(H \otimes \underset{nx\times nx}{I}\right) M_{xx} + \left(d_2 \otimes \underset{nx^2\times nx^2}{I}\right) K.$$

Applying the vec operator to both sides of (21) allows the equation to be factorised as follows

$$\text{vec}(A) + \left(\underset{nx\times nx}{I} \otimes \underset{n.nx^2\times nx^3}{B}\right) \text{vec}(h_{xxx}) + \\ \left(\underset{n.nx^3\times ny.nx^3}{C} + \underset{nx\times nx}{I} \otimes d_4 \otimes \underset{nx^2\times nx^2}{I}\right) \text{vec}(g_{xxx}) = 0, \quad (22)$$

where

$$B = \left(d_1 \otimes \underset{nx^2\times nx^2}{I}\right) + \left(d_2 \otimes \underset{nx^2\times nx^2}{I}\right)\left(g_x \otimes \underset{nx^2\times nx^2}{I}\right),$$

and

$$C = h_x' \otimes \left(\left(d_2 \otimes \underset{nx^2\times nx^2}{I}\right)\left(\underset{ny\times ny}{I} \otimes h_x' \otimes h_x'\right)\right).$$ [7]

Equation (22) can then be written as the linear system

$$\left[C + \left(\underset{nx\times nx}{I} \otimes d_4 \otimes \underset{nx^2\times nx^2}{I}\right), \underset{nx\times nx}{I} \otimes B\right]\left[\begin{array}{c}\text{vec}(g_{xxx}) \\ \text{vec}(h_{xxx})\end{array}\right] = -\text{vec}(A). \quad (23)$$

This is easily solved using standard matrix algebra. Alternatively equation (21) could have been written in the form of a generalised Sylvester equation and solved using the LAPACK routines of Kågström & Poromaa (1996) as explained in Gomme & Klein (2011). This second approach is computationally more efficient and uses less memory.

[7] Using $\text{vec}(XYZ) = (Z' \otimes X)\text{vec}(Y)$.

4.2. Solving for $g_{\sigma\sigma x}$ and $h_{\sigma\sigma x}$

Having found g_{xxx} and h_{xxx} I can now use them along with g_x, h_x, g_{xx}, h_{xx}, $g_{\sigma\sigma}$ and $h_{\sigma\sigma}$, to find $g_{\sigma\sigma x}$ and $h_{\sigma\sigma x}$. However, before I begin I need to define some additional matrices to be used in the solution.

4.2.1. Matrix definitions

I let N_σ be the gradient matrix for the policy functions with respect to σ, and $N^*_{\sigma x}$ be the Hessian matrix for the policy functions with respect to σ and all the elements in x_t

$$\underset{2n\times nx}{N_\sigma} = \begin{bmatrix} \underset{nx\times nx}{I} \\ g_x \\ \underset{n\times nx}{0} \end{bmatrix}, \quad \text{and} \quad \underset{2n\times nx^2}{N^*_{\sigma x}} = \begin{bmatrix} \underset{nx\times nx^2}{0} \\ g^*_{xx}\left(h_x \otimes \underset{nx\times nx}{I}\right) \\ \underset{n\times nx^2}{0} \end{bmatrix},$$

where $N^*_{\sigma x}$ follows from the definition of P in equation (14). The prediction error variance-covariance matrix for the predetermined variables takes the form

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

where σ_i^2 is the variance of the prediction error for the ith predetermined variable. Likewise, $\sigma_{i,j}$ is the covariance between the prediction errors for the ith and jth predetermined variables. I also introduce the the matrix trace (trm). This is defined in Gomme & Klein (2011)) so that for an $n.m \times n$ matrix

$$\begin{bmatrix} Y_1' & Y_2' & \cdots & Y_m' \end{bmatrix}',$$

the matrix trace gives an $m \times 1$ vector

$$\begin{bmatrix} \mathrm{tr}(Y_1) & \mathrm{tr}(Y_2) & \cdots & \mathrm{tr}(Y_m) \end{bmatrix}'.$$

The matrix trace is useful for taking the expectations of a random matrix.

4.2.2. Solution

I differentiate equation (6) with respect to σ twice and with respect to all elements in x_t once. I then substitute the first and second-order approximate solutions, along with the gradient, Hessian and third derivatives of f, and the matrix g_{xxx} into the resulting equations. The unknown coefficients $g_{\sigma\sigma x}$ and $h_{\sigma\sigma x}$ will be the solutions to this system of equations.

This is done more efficiently by applying Theorem 1 to equation (6) to get

$$\operatorname{trm}\left(\left(\underset{n\times n}{I}\otimes M_x'\otimes N_\sigma'\right)TN_\sigma\Sigma\right)+2\times\operatorname{trm}\left(\left(\underset{n\times n}{I}\otimes (N_{\sigma x}^*)'\right)HN_\sigma\Sigma\right)$$
$$+\left(\underset{n\times n}{I}\otimes M_x'\right)H\left[\begin{array}{c} h_{\sigma\sigma} \\ \operatorname{trm}\left(\left(\underset{ny\times ny}{I}\otimes\Sigma\right)g_{xx}\right)+g_xh_{\sigma\sigma}+g_{\sigma\sigma} \\ \underset{nx\times 1}{0} \\ g_{\sigma\sigma}\end{array}\right]+\left(D\otimes\underset{nx\times nx}{I}\right)\left[\begin{array}{c}\underset{nx^2\times 1}{h_{\sigma\sigma x}} \\ \underset{ny.nx\times 1}{P} \\ \underset{nx^2\times 1}{0} \\ \underset{ny.nx\times 1}{g_{\sigma\sigma x}}\end{array}\right]=0, \tag{24}$$

where

$$P=\left(\underset{ny\times ny}{I}\otimes h_x'\right)g_{xx}h_{\sigma\sigma}+\left(g_x\otimes\underset{nx\times nx}{I}\right)h_{\sigma\sigma x}+\left(\underset{ny\times ny}{I}\otimes h_x'\right)g_{\sigma\sigma x}+$$
$$\operatorname{trm}\left(\left(\underset{ny.nx\times ny.nx}{I}\otimes\underset{nx\times nx}{\Sigma}\right)\left(\underset{ny\times ny}{I}\otimes h_x'\otimes\underset{nx\times nx}{I}\right)g_{xxx}\right).$$

Substituting $D=[d_1,d_2,d_3,d_4]$ into equation (24) and rearranging gives

$$\underset{n\times 1}{G}+\left(d_1\otimes\underset{nx\times nx}{I}\right)h_{\sigma\sigma x}+\left(d_2\otimes\underset{nx\times nx}{I}\right)\left(g_x\otimes\underset{nx\times nx}{I}\right)h_{\sigma\sigma x}$$
$$+\left(d_2\otimes\underset{nx\times nx}{I}\right)\left(\underset{ny\times ny}{I}\otimes h_x'\right)g_{\sigma\sigma x}+\left(d_4\otimes\underset{nx\times nx}{I}\right)g_{\sigma\sigma x}=0, \tag{25}$$

where

$$G=\operatorname{trm}\left(\left(\underset{n\times n}{I}\otimes M_x'\otimes N_\sigma'\right)TN_\sigma\Sigma\right)+2\times\operatorname{trm}\left(\left(\underset{n\times n}{I}\otimes N_{\sigma x}'\right)HN_\sigma\Sigma\right)$$
$$+\left(\underset{n\times n}{I}\otimes M_x'\right)H\left[\begin{array}{c} h_{\sigma\sigma} \\ \operatorname{trm}\left(\left(\underset{ny\times ny}{I}\otimes\Sigma\right)g_{xx}\right)+g_xh_{\sigma\sigma}+g_{\sigma\sigma} \\ \underset{nx\times 1}{0} \\ g_{\sigma\sigma}\end{array}\right]$$
$$+\left(d_2\otimes\underset{nx\times nx}{I}\right)\left[\begin{array}{c}\left(\underset{ny\times ny}{I}\otimes h_x'\right)g_{xx}h_{\sigma\sigma}+\cdots \\ \cdots+\operatorname{trm}\left(\left(\underset{ny.nx\times ny.nx}{I}\otimes\underset{nx\times nx}{\Sigma}\right)\left(\underset{ny\times ny}{I}\otimes h_x'\otimes\underset{nx\times nx}{I}\right)g_{xxx}\right)\end{array}\right].$$

Equation (25) can be written as the linear system

$$\underset{n.nx\times n.nx}{Q}\left[\begin{array}{c}g_{\sigma\sigma x}\\ h_{\sigma\sigma x}\end{array}\right]=-G, \tag{26}$$

where

$$Q = \left[\left(d_2 \otimes \underset{nx \times nx}{I} \right) \left(\underset{ny \times ny}{I} \otimes h'_x \right) + \left(d_4 \otimes \underset{nx \times nx}{I} \right), \right.$$
$$\left. \left(d_1 \otimes \underset{nx \times nx}{I} \right) + \left(d_2 \otimes \underset{nx \times nx}{I} \right) \left(g_x \otimes \underset{nx \times nx}{I} \right) \right],$$

which is easily solved using standard matrix algebra.

4.3. Solving for $g_{\sigma\sigma\sigma}$ and $h_{\sigma\sigma\sigma}$

If the shocks in the model are assumed to be skewed, or even co-skewed, this will have further implications for agents behaviour. As a consequence an additional intercept correction needs to be made to the solution in the form of the vectors $g_{\sigma\sigma\sigma}$ and $h_{\sigma\sigma\sigma}$. In this section I solve for $g_{\sigma\sigma\sigma}$ and $h_{\sigma\sigma\sigma}$. But before I do this, I define some additional matrices used in the solution.

4.3.1. Matrix definitions

I let $N^*_{\sigma\sigma}$ be a rearrangement of the Hessian of the policy functions with respect to σ so that

$$N^*_{\sigma\sigma} = \begin{bmatrix} \underset{nx \times nx^2}{0} \\ g^*_{xx} \\ \underset{n \times nx^2}{0} \end{bmatrix}.$$

This follows from the definition of P in equation (14). I also define the skewness (co-skewness) matrix

$$\underset{nx \times nx^2}{S} = \begin{bmatrix} s_1 & s_{1,1,2} & \cdots & s_{1,nx,nx} \\ \vdots & & & \vdots \\ s_{nx,1,1} & \cdots & \cdots & s_{nx} \end{bmatrix}.$$

The skewness matrix contains the third moments of the prediction errors, where $s_i = E_t \left[u^3_{i,t} \right]$, $s_{i,j,k} = E_t \left[u_{i,t} u_{j,t} u_{k,t} \right]$, 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 $S = E_t \left[u_t \otimes u'_t \otimes u'_t \right]$, 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.

4.3.2. Solution

I differentiate equation (6) with respect to σ three times. Then, as was done in previous sections, I substitute in the first and second derivatives of the policy functions, and the first, second and third derivatives of the function f. The unknown coefficient matrices $g_{\sigma\sigma\sigma}$ and $h_{\sigma\sigma\sigma}$ can then be found as the solution to this system of equations. As was suggested earlier, using the third-order matrix chain rule from Theorem 1 greatly improves efficiency, and applied to this problem results in the following system of equations

$$\operatorname{trm}\left(\left(\underset{n\times n}{I}\otimes N_\sigma'\otimes N_\sigma'\right)TN_\sigma S\right)+3\times\operatorname{trm}\left(\left(\underset{n\times n}{I}\otimes (N_{\sigma\sigma}^*)'\right)HN_\sigma S\right)$$
$$+D\begin{bmatrix}\underset{nx\times 1}{h_{\sigma\sigma\sigma}}\\ \operatorname{trm}\left(\left(\underset{ny\times ny}{I}\otimes S\right)g_{xxx}\right)+g_x h_{\sigma\sigma\sigma}+g_{\sigma\sigma\sigma}\\ \underset{nx\times 1}{0}\\ \underset{ny\times 1}{g_{\sigma\sigma\sigma}}\end{bmatrix}=0. \quad (27)$$

Substituting $D=[d_1,d_2,d_3,d_4]$ into equation (27) and rearranging gives the linear system

$$[d_2+d_4,d_1+d_2g_x]\begin{bmatrix}g_{\sigma\sigma\sigma}\\ h_{\sigma\sigma\sigma}\end{bmatrix}=-\underset{n\times 1}{J}, \quad (28)$$

where

$$J=\operatorname{trm}\left(\left(\underset{n\times n}{I}\otimes N_\sigma'\otimes N_\sigma'\right)TN_\sigma S\right)+3\times\operatorname{trm}\left(\left(\underset{n\times n}{I}\otimes (N_{\sigma\sigma}^*)'\right)HNS\right)$$
$$+d_2\times\operatorname{trm}\left(\left(\underset{ny\times ny}{I}\otimes S\right)g_{xxx}\right).$$

Equation (28) can then be solved using standard matrix algebra.

5. A simple example

In this section I demonstrate my solution procedure using a simple 3 equation RBC model. The model can be written in the following form:

$$\begin{aligned}0&=c_t^{-\gamma}-\beta E_t\left\{\left(1+\alpha a_{t+1}k_t^{\alpha-1}-\delta\right)c_{t+1}^{-\gamma}\right\},\\ 0&=k_t+c_t-a_tk_{t-1}^{\alpha}-(1-\delta)\,k_{t-1},\\ 0&=a_t-a_{t-1}^{\rho}\exp(\sigma\varepsilon_t),\end{aligned}$$

where c_t is consumption, k_t is the capital stock, a_t is technology and ε_t is a technology shock. Dropping the time subscripts from the model allows the non-stochastic steady states to be calculated, $a=1$, $k=\left[\frac{\alpha\beta a}{1-\beta(1-\delta)}\right]^{\frac{1}{1-\alpha}}$, $c=ak^\alpha-\delta k$, $\varepsilon=0$. I calibrate the model such that: $\alpha=0.3$, $\beta=0.99$, $\delta=0.025$, $\gamma=1.1$, $\rho=0.8$, $\sigma=0.01$.

I find the solution of the model in terms of log deviations from the non-stochastic steady state, which requires making the following substitutions: $\hat{a}_t=\log(a_t)$, $\hat{k}_t=\log(k_t)$, $\hat{c}_t=\log(c_t)$, $\hat{\varepsilon}_t=0$, $\hat{a}_t^*=\log(a_t)$.[8] In addition, I include an auxiliary variable for technology

[8]Finding the solution in terms of level deviations from the non-stochastic steady state is also acceptable, but I stick with convention and find the solution in log terms.

because it appears in the model in periods $t-1$, t and $t+1$. I also include an additional equation for the $t+1$ technology shock (under this representation the shock is treated as a state variable). The model is now a 5 equation system:

$$
\begin{aligned}
0 &= \exp(\hat{c}_t)^{-\gamma} - \beta\left(1 + \alpha\exp(\hat{a}^*_{t+1})\exp(\hat{k}_t)^{\alpha-1} - \delta\right)\exp(\hat{c}_{t+1})^{-\gamma},\\
0 &= \exp(\hat{k}_t) + \exp(\hat{c}_t) - \exp(\hat{a}_t)\exp(\hat{k}_{t-1})^{\alpha} - (1-\delta)\exp(\hat{k}_{t-1}),\\
0 &= \hat{a}_t - \rho\hat{a}_{t-1} - \sigma\hat{\varepsilon}_t,\\
0 &= \hat{a}^*_t - \hat{a}_t,\\
0 &= \hat{\varepsilon}_{t+1},
\end{aligned}
$$

with the non-stochastic steady states: $\hat{a} = \log(a)$, $\hat{k} = \log(k)$, $\hat{c} = \log(c)$, $\hat{\varepsilon} = 0$, $\hat{a}^* = \log(a)$. I then define the vector of predetermined variables

$$x_t = \begin{bmatrix} \hat{k}_{t-1} & \hat{a}_{t-1} & \hat{\varepsilon}_t \end{bmatrix}',$$

and the vector of non-predetermined variables

$$y_t = \begin{bmatrix} \hat{c}_t & \hat{a}^*_t \end{bmatrix}'.$$

The variance-covariance matrix for the prediction errors for the predetermined variables only has a single non-zero element for the technology shock

$$\Sigma = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix}.$$

Likewise the skewness matrix only has a single non-zero element for the technology shock

$$S = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^3 \end{bmatrix}.$$

I set the skewness of the technology shock to be the cube of the standard deviation of the technology shock, so that it has a standardised skewness statistic equal to one.

The derivatives are solved in Matlab using automatic derivatives (see for example Bischof et al. 2008). Automatic derivatives are relatively quick to calculate and extremely accurate.[9]

I solve for the first-order terms using the method from Klein (2000), the second-order terms using the method from Gomme & Klein (2011), and the third-order terms using the

[9] Using a desktop pc with a 2993 Mhz Intel processor and 4GB RAM it takes 0.2431 seconds to read-in the model (equations, parameters etc), solve the non-stochastic steady state, and calculate first, second and third derivatives of the RBC model described in this section. Using the same computer it takes 0.3157 seconds to read-in the model, solve the non-stochastic steady state, and find first, second and third derivatives of an 8 equation (10 in total) NK DSGE model.

method outlined in this paper. The results are presented in Appendix B to allow readers to verify their accuracy.

The same model was coded in Dynare and using Matlab code from Andreasen (2011). The third-order approximations using the method outlined in this paper were checked against the third-order approximations from Dynare and Andreasen's code and found to be the same.[10] I tested my code for speed against Andreasen's code. The code from Andreasen (2011) uses tensor notation which allows me to compare the speed difference between the different solution methods. The tests were performed using a desktop pc with a 2993 Mhz Intel processor and 4GB RAM. I repeated the exercise with an 8 equation (10 equations when counting auxiliary variables) New Keynesian DSGE model. The times (in seconds) from both experiments are recorded in the table below:

Table 1: Computation Times

Model	Tensors	Without Tensors	Without Tensors (Opt)
RBC	0.1211	0.0040	0.0037
NK DSGE	1.1288	0.0447	0.0137

My solution method took 0.0040 seconds to find the third-order terms: g_{xxx}, h_{xxx}, $g_{\sigma\sigma x}$, $h_{\sigma\sigma x}$, $g_{\sigma\sigma\sigma}$ and $h_{\sigma\sigma\sigma}$, for the simple RBC model. Andreasen's code took 0.1211 seconds to find the same terms. For the New Keynesian DSGE model my solution method took 0.0447 seconds to solve, while Andreasens's code took 1.1288 seconds to solve. The procedure outlined in this paper appears to be orders of magnitude faster when using Matlab. The third column provides speeds for solving both the models using an optimised version of my code. More specifically I vectorise the Kronecker products as explained in Acklam (2003) and I remove the auxiliary equations (which are linear) from the system to solve for the second and third order solutions.[11] This results in further performance improvements, especially for the New Keynesian DSGE model which solves more than 80 times faster than when tensor notation is used. Andreasen's code has been optimised to exploit the symmetry of the derivatives. This decreases the number of derivatives that need to be calculated and shrinks the size of the matrices, which results in some speed gains. The procedure I outline in this paper does not exploit the symmetry of the derivatives, as I find the extra time required to shrink the matrices is more than the time saved in the matrix division.[12] The speed gains from my approach come from having a vectorised solution. Andreasen's code has 142 For loops and uses 621 lines of code. Because my solution uses matrix algebra, my Matlab code is vectorised with just 4 For loops and 67 lines of code. Vectorising the Kronecker products further improves the codes performance.

[10] I have also checked my solution method against Dynare and Andreasen's code using other small DSGE models.

[11] This requires making the distinction between predetermined variables, non-predetermined variables and variables that are both (e.g. the variable appears in the model equations in periods $t-1$, t and $t+1$).

[12] Exploiting the symmetry in the derivatives would improve memory usage allowing for larger models, but this would come at the expense of speed as For loops are slow to implement in Matlab.

6. Conclusion

In this paper I have demonstrated a new method for solving third-order approximations for DSGE models. The method does not involve tensor notation making it easier to understand and code, and faster to implement using Matlab. While much code exists in Matlab for solving third-order approximations, my procedure and code, due to it's simplicity, can form a blueprint for those wanting to write code in other programming languages, or it can be used by those wanting more flexibility and speed over existing Matlab routines. The Matlab code I provide results in speed gains of up to 80 times over existing Matlab codes using tensor notation.

Appendix A. Third order matrix chain rule

This appendix outlines the proof for the third-order matrix chain rule in Theorem 1.

Proof The proof proceeds as follows; from Theorem 1, the third-order matrix chain rule

$$\mathsf{S} = (\mathsf{D}' \otimes \mathsf{D}')\,\mathsf{ZD} + \mathsf{P}'\mathsf{WD} + \left(\mathsf{D}' \otimes \underset{m\times m}{I}\right)\left(\mathsf{W} \otimes \underset{m\times m}{I}\right)\mathsf{V} + \mathsf{Q}'\left(\mathsf{W} \otimes \underset{m\times m}{I}\right)\mathsf{V} + \left(\mathsf{R} \otimes \underset{m^2\times m^2}{I}\right)\mathsf{T}. \tag{A.1}$$

This can be written as the sum of 5 matrices

$$\mathsf{S}^1 = (\mathsf{D}' \otimes \mathsf{D}')\,\mathsf{ZD}, \tag{A.2}$$
$$\mathsf{S}^2 = \mathsf{P}'\mathsf{WD}, \tag{A.3}$$
$$\mathsf{S}^3 = \left(\mathsf{D}' \otimes \underset{m\times m}{I}\right)\left(\mathsf{W} \otimes \underset{m\times m}{I}\right)\mathsf{V}, \tag{A.4}$$
$$\mathsf{S}^4 = \mathsf{Q}'\left(\mathsf{W} \otimes \underset{m\times m}{I}\right)\mathsf{V}, \tag{A.5}$$
$$\mathsf{S}^5 = \left(\mathsf{R} \otimes \underset{m^2\times m^2}{I}\right)\mathsf{T}, \tag{A.6}$$

so that equation (A.1) can be rewritten as

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

To prove the Theorem I need to show that for each element in S, the following holds

$$\mathsf{y}_{i,j,k} = \mathsf{s}_{j+m(k-1),i} = \mathsf{s}^1_{j+m(k-1),i} + \mathsf{s}^2_{j+m(k-1),i} + \mathsf{s}^3_{j+m(k-1),i} + \mathsf{s}^4_{j+m(k-1),i} + \mathsf{s}^5_{j+m(k-1),i}.$$

That is the corresponding entries in S^1, S^2, S^3, S^4 and S^5 must add to the entry in the same position in the matrix S. This is equivalent to showing that

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

so that Fáa di Bruno's formula holds for each element in S. I proceed to do this in five steps, showing that for each of the five matrices making up the chain rule, the indexation matches up with the appropriate derivatives.

Step 1

From equation (A.3), $\mathsf{S}^1 = (\mathsf{D}' \otimes \mathsf{D}')\,\mathsf{Z}\mathsf{D}$. I need to show that $\mathsf{s}^1_{j+m(k-1),i} = \sum_{a=1}^{n}\sum_{b=1}^{n}\sum_{c=1}^{n} \mathsf{g}_{a,b,c}\mathsf{f}^a_i\mathsf{f}^b_k\mathsf{f}^c_j$

I define Ω^1 so that

$$\underset{m^2\times n^2}{\Omega^1} = \mathsf{D}'\otimes\mathsf{D}' = \begin{bmatrix}
\mathsf{f}^1_1\mathsf{f}^1_1 & \mathsf{f}^2_1\mathsf{f}^1_1 & \cdots & & \cdots & \mathsf{f}^n_1\mathsf{f}^n_1 \\
\mathsf{f}^1_1\mathsf{f}^1_2 & \mathsf{f}^2_1\mathsf{f}^1_2 & \cdots & & \cdots & \mathsf{f}^n_1\mathsf{f}^n_2 \\
\vdots & \vdots & & & & \vdots \\
\mathsf{f}^1_1\mathsf{f}^1_m & \mathsf{f}^2_1\mathsf{f}^1_m & \cdots & & \cdots & \mathsf{f}^n_1\mathsf{f}^n_m \\
\mathsf{f}^1_2\mathsf{f}^1_1 & \mathsf{f}^2_2\mathsf{f}^1_1 & \cdots & & \cdots & \mathsf{f}^n_2\mathsf{f}^n_1 \\
\vdots & \vdots & & \vdots & & \vdots \\
\vdots & \vdots & \cdots & \mathsf{f}^b_k\mathsf{f}^c_j & \cdots & \vdots \\
\vdots & \vdots & & \vdots & & \vdots \\
\mathsf{f}^1_m\mathsf{f}^1_m & \mathsf{f}^2_m\mathsf{f}^1_m & \cdots & & \cdots & \mathsf{f}^n_m\mathsf{f}^n_m
\end{bmatrix},$$

where

$$\omega^1_{j+m(k-1),b+n(c-1)} = \mathsf{f}^b_k\mathsf{f}^c_j,$$

is the element in the $j+m(k-1)$th row and the $b+n(c-1)$th column of Ω^1 for $j,k=1,\ldots,m$ and $b,c=1,\ldots,n$. I can then define Ω^2 to be

$$\underset{m^2\times n}{\Omega^2} = \Omega^1 \underset{n^2\times n}{\mathsf{Z}} =$$

$$\begin{bmatrix}
\sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{1,b,c}\mathsf{f}_1^b\mathsf{f}_1^c & \sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{2,b,c}\mathsf{f}_1^b\mathsf{f}_1^c & \cdots & & \cdots & \sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{n,b,c}\mathsf{f}_1^b\mathsf{f}_1^c \\
\sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{1,b,c}\mathsf{f}_1^b\mathsf{f}_2^c & \sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{2,b,c}\mathsf{f}_1^b\mathsf{f}_2^c & \cdots & & \cdots & \sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{n,b,c}\mathsf{f}_1^b\mathsf{f}_2^c \\
\vdots & \vdots & & \cdots & & \vdots \\
\sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{1,b,c}\mathsf{f}_1^b\mathsf{f}_m^c & \sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{2,b,c}\mathsf{f}_1^b\mathsf{f}_m^c & \cdots & & \cdots & \sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{n,b,c}\mathsf{f}_1^b\mathsf{f}_m^c \\
\sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{1,b,c}\mathsf{f}_2^b\mathsf{f}_1^c & \sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{2,b,c}\mathsf{f}_2^b\mathsf{f}_1^c & \cdots & & \cdots & \sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{n,b,c}\mathsf{f}_2^b\mathsf{f}_1^c \\
\vdots & \vdots & & \cdots & & \vdots \\
\vdots & \vdots & & \sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{a,b,c}\mathsf{f}_k^b\mathsf{f}_j^c & & \vdots \\
\vdots & \vdots & & \cdots & & \vdots \\
\sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{1,b,c}\mathsf{f}_m^b\mathsf{f}_m^c & \sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{2,b,c}\mathsf{f}_m^b\mathsf{f}_m^c & \cdots & & \cdots & \sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{n,b,c}\mathsf{f}_m^b\mathsf{f}_m^c
\end{bmatrix},$$

where

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

is the element in the $j+m(k-1)$th row and the ath column of Ω^2 for $j,k=1,\ldots,m$ and $a=1,\ldots,n$. The matrix D as defined in (11)

$$\underset{n\times m}{\mathsf{D}} = \begin{bmatrix} \mathsf{f}_1^1 & \cdots & \mathsf{f}_i^1 & \cdots & \mathsf{f}_m^1 \\ \vdots & & \vdots & & \vdots \\ \mathsf{f}_1^a & \cdots & \mathsf{f}_i^a & \cdots & \mathsf{f}_m^a \\ \vdots & & \vdots & & \vdots \\ \mathsf{f}_1^n & \cdots & \mathsf{f}_i^n & \cdots & \mathsf{f}_m^n \end{bmatrix}.$$

Here I use i to index the derivative and a to index the function so that I can write

$$\mathsf{f}_i^a = \mathsf{d}_{a,i}.$$

Multiplying Ω^2 by D gives S^1

$$\mathsf{S}^1 =$$

$$\begin{bmatrix}
\sum_{a=1}^{n}\sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{a,b,c}\mathsf{f}_1^b\mathsf{f}_1^c\mathsf{f}_1^a & \sum_{a=1}^{n}\sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{a,b,c}\mathsf{f}_1^b\mathsf{f}_1^c\mathsf{f}_2^a & \cdots & \sum_{a=1}^{n}\sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{a,b,c}\mathsf{f}_1^b\mathsf{f}_1^c\mathsf{f}_m^a \\
\sum_{a=1}^{n}\sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{a,b,c}\mathsf{f}_1^b\mathsf{f}_2^c\mathsf{f}_1^a & \sum_{a=1}^{n}\sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{a,b,c}\mathsf{f}_1^b\mathsf{f}_2^c\mathsf{f}_2^a & \cdots & \sum_{a=1}^{n}\sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{a,b,c}\mathsf{f}_1^b\mathsf{f}_2^c\mathsf{f}_m^a \\
\vdots & \vdots & & \vdots \\
\vdots & \vdots & \sum_{a=1}^{n}\sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{a,b,c}\mathsf{f}_k^b\mathsf{f}_j^c\mathsf{f}_i^a & \vdots \\
\vdots & \vdots & & \vdots \\
\sum_{a=1}^{n}\sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{a,b,c}\mathsf{f}_m^b\mathsf{f}_m^c\mathsf{f}_1^a & \sum_{a=1}^{n}\sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{a,b,c}\mathsf{f}_m^b\mathsf{f}_m^c\mathsf{f}_2^a & \cdots & \sum_{a=1}^{n}\sum_{b=1}^{n}\sum_{c=1}^{n}\mathsf{g}_{a,b,c}\mathsf{f}_m^b\mathsf{f}_m^c\mathsf{f}_m^a
\end{bmatrix}. \tag{A.7}$$

From the indexation in equation (A.7) it can be verified that[13]

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

as required.

Step 2

From equation (A.4), $\mathsf{S}^2 = \mathsf{P}'\mathsf{W}\mathsf{D}$. I need to show that $\mathsf{s}^2_{j+m(k-1),i} = \sum_{a=1}^{n}\sum_{b=1}^{n}\mathsf{g}_{a,b}\mathsf{f}_i^a\mathsf{f}_{j,k}^b$. I define

[13]The ordering of the derivatives of the f functions does not matter because these are scalars.

$\Omega^3 = \mathsf{P}'\mathsf{W}$, so that

$$\underset{m^2\times n}{\Omega^3} = \underset{m^2\times n}{\mathsf{P}'}\underset{n\times n}{W} = \begin{bmatrix} \sum_{b=1}^{n} \mathsf{g}_{1,b}\mathsf{f}^b_{1,1} & \sum_{b=1}^{n} \mathsf{g}_{2,b}\mathsf{f}^b_{1,1} & \cdots & \sum_{b=1}^{n} \mathsf{g}_{n,b}\mathsf{f}^b_{1,1} \\ \vdots & & & \vdots \\ \sum_{b=1}^{n} \mathsf{g}_{1,b}\mathsf{f}^b_{m,1} & \sum_{b=1}^{n} \mathsf{g}_{2,b}\mathsf{f}^b_{m,1} & \cdots & \sum_{b=1}^{n} \mathsf{g}_{n,b}\mathsf{f}^b_{m,1} \\ \sum_{b=1}^{n} \mathsf{g}_{1,b}\mathsf{f}^b_{1,2} & \sum_{b=1}^{n} \mathsf{g}_{2,b}\mathsf{f}^b_{1,2} & \cdots & \sum_{b=1}^{n} \mathsf{g}_{n,b}\mathsf{f}^b_{1,2} \\ \vdots & & & \vdots \\ \vdots & \sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_{j,k} & & \vdots \\ \vdots & & & \vdots \\ \sum_{b=1}^{n} \mathsf{g}_{1,b}\mathsf{f}^b_{m,m} & \sum_{b=1}^{n} \mathsf{g}_{2,b}\mathsf{f}^b_{m,m} & \cdots & \sum_{b=1}^{n} \mathsf{g}_{n,b}\mathsf{f}^b_{m,m} \end{bmatrix},$$

where

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

is the element in the $j+m(k-1)$th row and the ath column of Ω^3, for $j,k = 1,\ldots,m$ and $a = 1,\ldots,n$.

$$\mathrm{S}^2 = \Omega^3\mathrm{D}$$

$$\mathrm{S}^2 = \begin{bmatrix} \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_{1,1}\mathsf{f}^a_1 & \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_{1,1}\mathsf{f}^a_2 & \cdots & \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_{1,1}\mathsf{f}^a_m \\ \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_{2,1}\mathsf{f}^a_1 & \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_{2,1}\mathsf{f}^a_2 & \cdots & \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_{2,1}\mathsf{f}^a_m \\ \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_{m,1}\mathsf{f}^a_1 & \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_{m,1}\mathsf{f}^a_2 & \cdots & \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_{m,1}\mathsf{f}^a_m \\ & & \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_{j,k}\mathsf{f}^a_i & \\ \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_{m,m}\mathsf{f}^a_1 & \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_{m,m}\mathsf{f}^a_2 & \cdots & \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_{m,m}\mathsf{f}^a_m \end{bmatrix},$$

so that

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

as required.

Step 3

From equation (A.5), $\mathsf{S}^3 = \left(\mathsf{D}' \otimes \underset{m\times m}{I}\right)\left(\mathsf{W} \otimes \underset{m\times m}{I}\right)\mathsf{V}$. I need to show that $\mathsf{s}^3_{j+m(k-1),i} = \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^a_{i,j}\mathsf{f}^b_k$. I define Ω^4 so that

$$\underset{m^2\times n.m}{\Omega^4} = \left(D' \otimes \underset{m\times m}{I}\right)\left(W \otimes \underset{m\times m}{I}\right) =$$

$$\begin{bmatrix}
\sum_{b=1}^{n}\mathsf{g}_{1,b}\mathsf{f}^b_1 & 0 & 0 & \cdots & 0 & \cdots & \sum_{b=1}^{n}\mathsf{g}_{n,b}\mathsf{f}^b_1 & 0 & 0 & \cdots & 0 \\
0 & \sum_{b=1}^{n}\mathsf{g}_{1,b}\mathsf{f}^b_1 & 0 & \cdots & 0 & \cdots & 0 & \sum_{b=1}^{n}\mathsf{g}_{n,b}\mathsf{f}^b_1 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & & & \cdots & \vdots & \vdots & \vdots & & \\
0 & 0 & \cdots & \cdots & \sum_{b=1}^{n}\mathsf{g}_{1,b}\mathsf{f}^b_1 & \cdots & 0 & 0 & \cdots & \cdots & \sum_{b=1}^{n}\mathsf{g}_{n,b}\mathsf{f}^b_1 \\
\vdots & \vdots & \vdots & & & \cdots & \vdots & \vdots & \vdots & & \\
\vdots & \vdots & \vdots & & & \cdots & \vdots & \vdots & \vdots & & \\
\sum_{b=1}^{n}\mathsf{g}_{1,b}\mathsf{f}^b_m & 0 & 0 & \cdots & 0 & \cdots & \sum_{b=1}^{n}\mathsf{g}_{n,b}\mathsf{f}^b_m & 0 & 0 & \cdots & 0 \\
0 & \sum_{b=1}^{n}\mathsf{g}_{1,b}\mathsf{f}^b_m & 0 & \cdots & 0 & \cdots & 0 & \sum_{b=1}^{n}\mathsf{g}_{n,b}\mathsf{f}^b_m & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & & & \cdots & \vdots & \vdots & \vdots & & \\
0 & 0 & \cdots & \cdots & \sum_{b=1}^{n}\mathsf{g}_{1,b}\mathsf{f}^b_m & \cdots & 0 & 0 & \cdots & \cdots & \sum_{b=1}^{n}\mathsf{g}_{n,b}\mathsf{f}^b_m
\end{bmatrix},$$

where

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

is the element in the $j+m(k-1)$th row and the $j+m(a-1)$th column of Ω^4, for $j,k = 1,\ldots,m$ and $a = 1,\ldots,n$. Using the definition of S^3 I can write

$$\mathsf{S}^3 = \Omega^4\mathsf{V},$$

$$
S^3 = \begin{bmatrix}
\sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}_1^b\mathsf{f}_{1,1}^a & \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}_1^b\mathsf{f}_{2,1}^a & \cdots & \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}_1^b\mathsf{f}_{m,1}^a \\
\sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}_1^b\mathsf{f}_{1,2}^a & \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}_1^b\mathsf{f}_{2,2}^a & \cdots & \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}_1^b\mathsf{f}_{m,2}^a \\
\vdots & & & \vdots \\
\vdots & & \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}_k^b\mathsf{f}_{i,j}^a & \vdots \\
\vdots & & & \vdots \\
\sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}_m^b\mathsf{f}_{1,m}^a & \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}_m^b\mathsf{f}_{2,m}^a & \cdots & \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}_m^b\mathsf{f}_{m,m}^a
\end{bmatrix},
$$

where

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

as required.

Step 4

From equation (A.6), $S^4 = \mathsf{Q}'\left(W \otimes \underset{m\times m}{I}\right)\mathsf{V}$. I need to show $\mathsf{s}^4_{j+m(k-1),i} = \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^a_{i,k}\mathsf{f}^b_j$.

From the definition of Q

$$
\mathsf{Q}' = \begin{bmatrix}
\mathsf{f}_1^1 & 0 & \cdots & 0 & \mathsf{f}_1^2 & 0 & \cdots & \mathsf{f}_1^n & 0 & \cdots & 0 \\
\mathsf{f}_2^1 & 0 & \cdots & 0 & \mathsf{f}_2^2 & 0 & \cdots & \mathsf{f}_2^n & 0 & \cdots & 0 \\
\vdots & & \vdots & & \vdots & & & \vdots & & \vdots & 0 \\
\mathsf{f}_m^1 & 0 & \cdots & 0 & \mathsf{f}_m^2 & 0 & \cdots & \mathsf{f}_m^n & 0 & \cdots & 0 \\
0 & \mathsf{f}_1^1 & \cdots & 0 & 0 & \mathsf{f}_1^2 & \cdots & 0 & \mathsf{f}_1^n & \cdots & 0 \\
0 & \mathsf{f}_2^1 & \cdots & 0 & 0 & \mathsf{f}_2^2 & \cdots & 0 & \mathsf{f}_2^n & \cdots & 0 \\
\vdots & & \vdots & & \vdots & & & \vdots & & \vdots & 0 \\
0 & \mathsf{f}_m^1 & \cdots & 0 & 0 & \mathsf{f}_m^2 & \cdots & 0 & \mathsf{f}_m^n & \cdots & 0 \\
\vdots & & \vdots & & \vdots & & & \vdots & & \vdots & \vdots \\
0 & 0 & \cdots & \mathsf{f}_m^1 & 0 & 0 & \cdots & \mathsf{f}_m^2 & 0 & \cdots & \mathsf{f}_m^n
\end{bmatrix},
$$

where

$$
\mathsf{q}_{j+m(k-1),k+m(b-1)} = \mathsf{f}_j^b,
$$

is the element in the $j+m(k-1)$th row and the $k+m(b-1)$th column of Q', for $j, k = 1, \ldots, m$

and $b = 1, \ldots, n$. The Kronecker product of W and the $m \times m$ identity matrix is given by

$$\mathsf{W} \otimes \underset{m\times m}{I} = \begin{bmatrix} \mathsf{g}_{1,1} & 0 & \cdots & 0 & \cdots & \mathsf{g}_{n,1} & 0 & \cdots & 0 \\ 0 & \mathsf{g}_{1,1} & \cdots & 0 & \cdots & 0 & \mathsf{g}_{n,1} & \cdots & 0 \\ \vdots & \vdots & & & & & & & \\ 0 & 0 & \cdots & \mathsf{g}_{1,1} & \cdots & 0 & 0 & \cdots & \mathsf{g}_{n,1} \\ \vdots & \vdots & & & & & & & \\ \mathsf{g}_{1,n} & 0 & \cdots & 0 & \cdots & \mathsf{g}_{n,n} & 0 & \cdots & 0 \\ 0 & \mathsf{g}_{1,n} & \cdots & 0 & \cdots & 0 & \mathsf{g}_{n,n} & \cdots & 0 \\ \vdots & \vdots & & & & & & & \\ 0 & 0 & \cdots & \mathsf{g}_{1,n} & \cdots & 0 & 0 & \cdots & \mathsf{g}_{n,n} \end{bmatrix},$$

where

$$\mathsf{w}_{k+m(b-1),k+m(a-1)} = \mathsf{g}_{a,b},$$

is the element in the $k + m(b-1)$th row and the $k + m(a-1)$th column of $\mathsf{W} \otimes \underset{m\times m}{I}$, for $k = 1, \ldots, m$ and $a, b = 1, \ldots, n$. I define Ω^5 so that

$$\underset{m^2\times n.m}{\Omega^5} = \mathsf{Q}'\left(\mathsf{W} \otimes \underset{m\times m}{I}\right) =$$

$$\begin{bmatrix} \sum_{b=1}^{n} \mathsf{g}_{1,b}\mathsf{f}_1^b & 0 & \cdots & 0 & \cdots & \sum_{b=1}^{n} \mathsf{g}_{n,b}\mathsf{f}_1^b & 0 & \cdots & 0 \\ \sum_{b=1}^{n} \mathsf{g}_{1,b}\mathsf{f}_2^b & 0 & \cdots & 0 & \cdots & \sum_{b=1}^{n} \mathsf{g}_{n,b}\mathsf{f}_2^b & 0 & \cdots & 0 \\ \vdots & \vdots & & & & & & & \\ \sum_{b=1}^{n} \mathsf{g}_{1,b}\mathsf{f}_m^b & 0 & \cdots & 0 & \cdots & \sum_{b=1}^{n} \mathsf{g}_{n,b}\mathsf{f}_m^b & 0 & \cdots & 0 \\ 0 & \sum_{b=1}^{n} \mathsf{g}_{1,b}\mathsf{f}_1^b & \cdots & 0 & \cdots & 0 & \sum_{b=1}^{n} \mathsf{g}_{n,b}\mathsf{f}_1^b & \cdots & 0 \\ 0 & \sum_{b=1}^{n} \mathsf{g}_{1,b}\mathsf{f}_2^b & \cdots & 0 & \cdots & 0 & \sum_{b=1}^{n} \mathsf{g}_{n,b}\mathsf{f}_2^b & \cdots & 0 \\ \vdots & \vdots & & & & & & & \\ 0 & \sum_{b=1}^{n} \mathsf{g}_{1,b}\mathsf{f}_m^b & \cdots & 0 & \cdots & 0 & \sum_{b=1}^{n} \mathsf{g}_{n,b}\mathsf{f}_m^b & \cdots & 0 \\ \vdots & \vdots & & & & & & & \\ 0 & 0 & \cdots & \sum_{b=1}^{n} \mathsf{g}_{1,b}\mathsf{f}_1^b & \cdots & 0 & 0 & \cdots & \sum_{b=1}^{n} \mathsf{g}_{n,b}\mathsf{f}_1^b \\ 0 & 0 & \cdots & \sum_{b=1}^{n} \mathsf{g}_{1,b}\mathsf{f}_2^b & \cdots & 0 & 0 & \cdots & \sum_{b=1}^{n} \mathsf{g}_{n,b}\mathsf{f}_2^b \\ \vdots & \vdots & & & & & & & \\ 0 & 0 & \cdots & \sum_{b=1}^{n} \mathsf{g}_{1,b}\mathsf{f}_m^b & \cdots & 0 & 0 & \cdots & \sum_{b=1}^{n} \mathsf{g}_{n,b}\mathsf{f}_m^b \end{bmatrix},$$

where

$$\omega^5_{j+m(k-1),k+m(a-1)} = \sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_j,$$

is the element in the $j+m(k-1)$th row and the $k+m(a-1)$th column of Ω^5, for $j,k = 1,\ldots,m$ and $a = 1,\ldots,n$. Using the definition of S^4, I can write

$$\mathsf{S}^4 = \Omega^5\mathsf{V},$$

$$\mathsf{S}^4 = \begin{bmatrix} \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_1\mathsf{f}^a_{1,1} & \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_1\mathsf{f}^a_{2,1} & \cdots & \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_1\mathsf{f}^a_{m,1} \\ \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_2\mathsf{f}^a_{1,1} & \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_2\mathsf{f}^a_{2,1} & \cdots & \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_2\mathsf{f}^a_{m,1} \\ & & \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_j\mathsf{f}^a_{i,k} & \\ \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_m\mathsf{f}^a_{1,m} & \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_m\mathsf{f}^a_{2,m} & \cdots & \sum_{a=1}^{n}\sum_{b=1}^{n} \mathsf{g}_{a,b}\mathsf{f}^b_m\mathsf{f}^a_{m,m} \end{bmatrix},$$

where

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

as required.

Step 5

From equation (A.6), $\mathsf{S}^5 = \left(\mathsf{R} \otimes \underset{m^2\times m^2}{I}\right)\mathsf{T}$. I need to show that $\mathsf{s}^5_{j+m(k-1),i} = \sum_{a=1}^{n} \mathsf{g}_a\mathsf{f}^a_{i,j,k}$. I begin by defining Ω^6 such that

$$\underset{m^2\times n.m^2}{\Omega_6} = \mathsf{R} \otimes \underset{m^2\times m^2}{I} = \begin{bmatrix} \mathsf{g}_1 & 0 & \cdots & 0 & \mathsf{g}_2 & 0 & \cdots & \mathsf{g}_n & 0 & \cdots & 0 \\ 0 & \mathsf{g}_1 & \cdots & 0 & 0 & \mathsf{g}_2 & \cdots & 0 & \mathsf{g}_n & \cdots & 0 \\ \vdots & & & & & & & & & & \\ 0 & 0 & \cdots & \mathsf{g}_1 & 0 & 0 & \cdots & 0 & 0 & \cdots & \mathsf{g}_n \end{bmatrix},$$

where

$$\omega^6_{j+m(k-1),j+m(k-1)+m^2(a-1)} = \mathsf{g}_a,$$

is the element in the $j+m(k-1)$th row and the $j+m(k-1)+m^2(a-1)$th column in Ω^6, for $j,k = 1,\ldots,m$ and $a = 1,\ldots,n$. Using the definition of S^5, I can write

$$\mathsf{S}^5 = \Omega^6\mathsf{T},$$

$$S5 = \begin{bmatrix} \sum_{a=1}^{n} \mathsf{g}_a \mathsf{f}^a_{1,1,1} & \sum_{a=1}^{n} \mathsf{g}_a \mathsf{f}^a_{2,1,1} & \cdots & \sum_{a=1}^{n} \mathsf{g}_a \mathsf{f}^a_{m,1,1} \\ \sum_{a=1}^{n} \mathsf{g}_a \mathsf{f}^a_{1,2,1} & \sum_{a=1}^{n} \mathsf{g}_a \mathsf{f}^a_{2,2,1} & \cdots & \sum_{a=1}^{n} \mathsf{g}_a \mathsf{f}^a_{m,2,1} \\ \vdots & \vdots & & \vdots \\ \sum_{a=1}^{n} \mathsf{g}_a \mathsf{f}^a_{1,m,1} & \sum_{a=1}^{n} \mathsf{g}_a \mathsf{f}^a_{2,m,1} & \cdots & \sum_{a=1}^{n} \mathsf{g}_a \mathsf{f}^a_{m,m,1} \\ \sum_{a=1}^{n} \mathsf{g}_a \mathsf{f}^a_{1,1,2} & \sum_{a=1}^{n} \mathsf{g}_a \mathsf{f}^a_{2,1,2} & \cdots & \sum_{a=1}^{n} \mathsf{g}_a \mathsf{f}^a_{m,1,2} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & \sum_{a=1}^{n} \mathsf{g}_a \mathsf{f}^a_{i,j,k} & \vdots \\ \vdots & \vdots & & \vdots \\ \sum_{a=1}^{n} \mathsf{g}_a \mathsf{f}^a_{1,m,m} & \sum_{a=1}^{n} \mathsf{g}_a \mathsf{f}^a_{2,m,m} & \cdots & \sum_{a=1}^{n} \mathsf{g}_a \mathsf{f}^a_{m,m,m} \end{bmatrix},$$

where

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

as required. This completes the proof. ■

Appendix B. Model solution

The solved matrices from section 5 are presented below.

$$g_x = \begin{bmatrix} 0.538516074338190 & 0.128222800563108 & 0.160278500703885 \\ 0 & 0.800000000000000 & 1.000000000000000 \end{bmatrix},$$

$$h_x = \begin{bmatrix} 0.960555718076461 & 0.081805764224287 & 0.102257205280358 \\ 0 & 0.800000000000000 & 1.000000000000000 \\ 0 & 0 & 0 \end{bmatrix},$$

$$g_{xx} = \begin{bmatrix} 0.050410880298460 & -0.056379980258910 & -0.070474975323637 \\ -0.056379980258910 & 0.048554933367482 & 0.060693666709352 \\ -0.070474975323637 & 0.060693666709352 & 0.075867083386690 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$h_{xx} = \begin{bmatrix} 0.031544108616856 & -0.051663874599147 & -0.064579843248933 \\ -0.051663874599147 & 0.062210119144458 & 0.077762648930573 \\ -0.064579843248933 & 0.077762648930573 & 0.097203311163216 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$g_{\sigma\sigma} = \begin{bmatrix} 0.526512345088850 \times 10^{-4} \\ 0 \end{bmatrix},$$

$$h_{\sigma\sigma} = \begin{bmatrix} -0.484409085170130 \times 10^{-5} \\ 0 \\ 0 \end{bmatrix},$$

$$g_{xxx} = \begin{bmatrix} 0.000886224176982 & 0.018042424368051 & 0.022553030460064 \\ 0.018042424368051 & -0.016047638262395 & -0.020059547827995 \\ 0.022553030460064 & -0.020059547827994 & -0.025074434784993 \\ 0.018042424368051 & -0.016047638262395 & -0.020059547827995 \\ -0.016047638262396 & 0.019412734848266 & 0.024265918560332 \\ -0.020059547827995 & 0.024265918560332 & 0.030332398200415 \\ 0.022553030460064 & -0.020059547827995 & -0.025074434784993 \\ -0.020059547827995 & 0.024265918560332 & 0.030332398200415 \\ -0.025074434784993 & 0.030332398200415 & 0.037915497750519 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$h_{xxx} = \begin{bmatrix} -0.020956383687171 & 0.029527273689885 & 0.036909092112356 \\ 0.029527273689885 & -0.035680637163452 & -0.044600796454315 \\ 0.036909092112356 & -0.044600796454315 & -0.055750995567894 \\ 0.029527273689885 & -0.035680637163452 & -0.044600796454315 \\ -0.035680637163452 & 0.040392437073006 & 0.050490546341257 \\ -0.044600796454315 & 0.050490546341257 & 0.063113182926571 \\ 0.036909092112356 & -0.044600796454315 & -0.055750995567894 \\ -0.044600796454315 & 0.050490546341257 & 0.063113182926571 \\ -0.055750995567894 & 0.063113182926571 & 0.078891478658214 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$g_{\sigma\sigma x} = \begin{bmatrix} 0.199558292329446 \times 10^{-4} \\ 0.059796933577375 \times 10^{-4} \\ 0.074746166971719 \times 10^{-4} \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$h_{\sigma\sigma x} = \begin{bmatrix} 0.208394896512764 \times 10^{-6} \\ -0.775000263651503 \times 10^{-6} \\ -0.968750329564378 \times 10^{-6} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$g_{\sigma\sigma\sigma} = \begin{bmatrix} -0.138593020922434 \times 10^{-6} \\ 0 \end{bmatrix},$$

$$h_{\sigma\sigma\sigma} = \begin{bmatrix} 0.127510245680320 \times 10^{-7} \\ 0 \\ 0 \end{bmatrix}.$$

References

Acklam, P. J. (2003). Matlab array manipulation tips and tricks. Draft Book. URL http://home.online.no/~pjacklam/matlab/doc/mtt/doc/mtt.pdf.

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.

Bischof, C., Bücker, H., & Hovland, P. (2008). *Advances in automatic differentiation*. Lecture notes in computational science and engineering, Springer.

Chen, B. & Zadrozny, P. (2003). Higher-moments in perturbation solution of the linear-quadratic exponential gaussian optimal control problem. *Computational Economics*, *21*(1), 45–64. URL http://ideas.repec.org/a/kap/compec/v21y2003i1p45-64.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.

Juillard, M. (2003). *DYNARE: A Program for Solving Rational Expectations Models*. Paris, France: CEPREMAP and University of Paris 8, 2nd edn.

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. (2011). Dsge models with dynare++. a tutorial. Mimeo. URL http://www.dynare.org/documentation-and-support/dynarepp/dynare-tutorial.pdf.

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.

Lan, H. & Meyer-Gohde, A. (2011). Solving dsge models with a nonlinear moving average. SFB 649 Discussion Papers SFB649DP2011-087, Sonderforschungsbereich 649, Humboldt University, Berlin, Germany. URL http://ideas.repec.org/p/hum/wpaper/sfb649dp2011-087.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.

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.

Swanson, E., Anderson, G., & Levin, A. (2006). Higher-order perturbation solutions to dynamic, discrete-time rational expectations models. Working Paper Series 2006-01, Federal Reserve Bank of San Francisco. URL http://ideas.repec.org/p/fip/fedfwp/2006-01.html.

Van Binsbergen, J., Fernndez-Villaverde, J., Koijen, R. S., & Rubio-Ramrez, J. F. (2010). The term structure of interest rates in a dsge model with recursive preferences. NBER Working Papers 15890, National Bureau of Economic Research, Inc. URL http://ideas.repec.org/p/nbr/nberwo/15890.html.