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

DRAFT
This version: May 2012

Andrew Binning¹
Monetary Policy 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. 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.

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.² 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. Current software and routines that solve for third-order approximations use tensor notation. 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

¹The author would like to thank Martin Andreassen, Paul Klein, Junior Maih and Martin Seneca for useful comments.
²For example Dynare (see Juillard, 2003), Dynare++ (see Kamenik, 2011), Perturbation AIM (see Swanson et al., 2006) and codes by Schmitt-Grohe & Uribe (2004), Andreassen (2011), Ruge-Murcia (2010), Gomme & Klein (2011)
approximation without tensors. I extend their method to third-order approximations. I also provide Matlab code for my solution method. The paper is set out as follows, section 2 covers some preliminaries, section 3 outlines the matrix algebra to find the solution, section 4 shows the results with a simple RBC model, while section 5 concludes.

2. Preliminaries

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

\[ E_t(f(x_{t+1}, y_{t+1}, x_t, y_t)) = 0, \] (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 takes the form:

\[ y_t = g(x_t, \sigma), \] (2)
\[ x_{t+1} = h(x_t, \sigma) + \sigma \epsilon_{t+1}, \] (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 \( \epsilon_{t+1} \) is an \( nx \times 1 \) vector of shocks. Typically the functions \( g \) and \( h \) are unknown, so we have to resort to taking approximations around the non-stochastic steady state.

The third-order approximation of the policy functions, \( g \) and \( h \) takes the form:

\[ y_t = g_x x_t + \frac{1}{2} \sigma^2 g_{\sigma x} + \frac{1}{2} \left( I_{ny \times ny} \otimes x'_t \right) g_{xx} x_t \]
\[ + \frac{1}{6} \sigma^3 g_{\sigma x x} + \frac{1}{2} \sigma^2 \left( I_{ny \times ny} \otimes x'_t \right) g_{x x} + \frac{1}{6} \left( I_{ny \times ny} \otimes x'_t \otimes x'_t \right) g_{xx x} x_t, \] (4)

\[ x_{t+1} = h_x x_t + \frac{1}{2} \sigma^2 h_{\sigma x} + \frac{1}{2} \left( I_{nx \times nx} \otimes x'_t \right) h_{xx} x_t \]
\[ + \frac{1}{6} \sigma^3 h_{\sigma x x} + \frac{1}{2} \sigma^2 \left( I_{nx \times nx} \otimes x'_t \right) h_{x x} + \frac{1}{6} \left( I_{nx \times nx} \otimes x'_t \otimes x'_t \right) h_{xx x} x_t, \] (5)

where \( g_x \) and \( h_x \) are the partial derivatives of \( g \) and \( h \) with respect to \( x \) evaluated at the non-stochastic steady state, these form the first-order accurate solution. The terms: \( g_{xx}, h_{xx}, \) and \( g_{\sigma x} \) and \( h_{\sigma x} \), are the second derivatives of \( g \) and \( h \) with respect to \( x \) and \( \sigma \) evaluated
at the non-stochastic steady state, these form the second-order approximation.\(^3\) The terms \(g_{xxx}, g_{xox}, h_{xox}, g_{oxo}, h_{oxo}\) and \(h_{oxo}\) are the third derivatives of \(g\) and \(h\) with respect to \(x\) and \(\sigma\).\(^4\) Note that \(g_{oxo}\) and \(h_{oxo}\) are dependent on the third moment of the shocks. These terms will be important if the shocks come from a distribution with a non-zero third moment.

These terms can be found by substituting the policy functions (equations (2) and (3)) into equation (1) to get

\[
E_t \left( f(h(x_t, \sigma) + \sigma \epsilon_{t+1}, g(h(x_t) + \sigma \epsilon_{t+1}, \sigma), x_t, g(x_t, \sigma)) \right) = 0,
\]

and then differentiating with respect to \(x_t\) and \(\sigma\) the appropriate number of times.

I wish to find a third-order approximation of \(g\) and \(h\) around the non-stochastic steady state. In order to solve for a third-order approximation I need to solve for a first, and then a second order approximation (this is explained in Aruoba et al. 2006). The first order accurate solution to equation (6) takes the form:

\[
\begin{align*}
\mathbf{g}_x &= \begin{bmatrix}
g_{1x,1} & \cdots & g_{1x,nx} \\
0 & \ddots & 0 \\
g_{nx,x,1} & \cdots & g_{nx,xx} 
\end{bmatrix}, \\
\mathbf{h}_x &= \begin{bmatrix}
h_{1x,1} & \cdots & h_{1x,nx} \\
0 & \ddots & 0 \\
h_{nx,x,1} & \cdots & h_{nx,xx} 
\end{bmatrix},
\end{align*}
\]

where \(g_x\) and \(h_x\) are matrices of first derivatives of the policy function evaluated at the non-stochastic steady state.\(^5\) That is \(g_{i,x,j} = \frac{\partial g_i(x_t, \sigma)}{\partial x_{j,t}} \bigg|_{(x_t=0, \sigma=0)}\) and \(h_{i,x,j} = \frac{\partial h_i(x_t, \sigma)}{\partial x_{j,t}} \bigg|_{(x_t=0, \sigma=0)}\) where \(g_i\) is the policy function for the \(i\)th non-predetermined variable and \(h_i\) is the policy function for the \(i\)th predetermined variable.

The second-order accurate solution to the policy functions takes the form:\(^6\)

\[
\begin{align*}
\mathbf{g}_{xx} &= \begin{bmatrix}
g_{1,x11} & \cdots & g_{1,x1nx} \\
0 & \ddots & 0 \\
g_{nx,x11} & \cdots & g_{nx,xnx} \\
g_{1,x21} & \cdots & g_{1,x2nx} \\
0 & \ddots & 0 \\
g_{nx,x21} & \cdots & g_{nx,xnx} \\
\vdots & & \vdots \\
g_{nx,xnx,1} & \cdots & g_{nx,xnx,nnx} 
\end{bmatrix},
\end{align*}
\]

\(^3\)Schmitt-Grohe & Uribe (2004) show that \(g_{x\sigma}, h_{x\sigma}, g_{x\sigma x}\) and \(h_{x\sigma x}\) are zero.

\(^4\)Andreasen (2011) shows that \(g_{xx\sigma}\) and \(h_{xx\sigma}\) are zero.

\(^5\)Equation (6) can be solved using Klein (2000) for example.

\(^6\)Equation (6) can be solved using Gomme & Klein (2011) for example.
where \( g_{xx} \) and \( h_{xx} \) are matrices of the second derivatives of \( g \) and \( h \) with respect to \( x_t \) evaluated at the non-stochastic steady state, 

\[
g_{i,j} = \frac{\partial^2 g}{\partial x_t \partial x_k} \Big|_{(x_t=0, \sigma=0)}
\]

Likewise, \( g_{\sigma\sigma} \) and \( h_{\sigma\sigma} \) are matrices of the second-derivatives of \( g \) and \( h \) with respect to \( \sigma \) evaluated at the non-stochastic steady-state, 

\[
h_{i,j} = \frac{\partial^2 h}{\partial \sigma \partial \sigma} \Big|_{(x_t=0, \sigma=0)}
\]

3. Third-order approximation

This section outlines the steps required to find: \( g_{xxx} \), \( h_{xxx} \), \( g_{\sigma\sigma x} \), \( h_{\sigma\sigma x} \), \( g_{\sigma\sigma\sigma} \) and \( h_{\sigma\sigma\sigma} \), matrices that form the solution for a third-order approximation of the policy functions.

3.1. Solving for \( g_{xxx} \) and \( h_{xxx} \)

Before attempting to solve the third-order approximation, I define some additional matrices that will prove useful for the solution procedure.

3.1.1. Matrix Definitions

I define 

\[
h^*_x = \begin{bmatrix}
I_{nx \times nx} \otimes h_x[1,\cdot] \\
\vdots \\
I_{nx \times nx} \otimes h_x[nx,\cdot]
\end{bmatrix},
\]

where \( h_x[i,\cdot] \) is the \( i \)th 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 \( h_x \).

The matrices \( g^*_x \) and \( h^*_x \) are defined as
\[
g_{xx}^{*} = \begin{bmatrix}
g_{1,nx,1} & \cdots & g_{1,nx,nx} \\
\vdots & & \vdots \\
g_{ny,nx,1} & \cdots & g_{ny,nx,nx}
\end{bmatrix},
\]

and are just rearrangements of \(g_{xx}\) and \(h_{xx}\). The matrices \(M_x\) and \(M_{xx}\) are constructed so that:

\[
M_x = \begin{bmatrix}
h_x \\
g_xh_x \\
0 \\
g_x
\end{bmatrix},
\]

and

\[
M_{xx} = \begin{bmatrix}
h_{xx} \\
\left(I_{ny \times ny} \otimes h_x^t\right)g_{xx}h_x + \left(g_x \otimes I_{nx \times nx}\right)h_{xx} \\
0 \\
g_{xx}
\end{bmatrix}.
\]

I define the matrix \(M_x^*\):

\[
M_x^* = \begin{bmatrix}
I_{n_x \times n_x} \otimes M_x[1, \cdot] \\
\vdots \\
I_{n_x \times n_x} \otimes M_x[2n, \cdot]
\end{bmatrix},
\]

where \(M_x[i, \cdot]\) is the \(i\)th row of the \(M_x\) matrix so that \(M_x^*\) is made up of the kronecker product of the \(n_x \times n_x\) identity matrix and the rows of \(M_x\).

I also define the matrix \(M_{xx}^*\):

\[
M_{xx}^* = \begin{bmatrix}
h_{xx}^* \\
g_{xx}^* (h_x \otimes h_x) + g_x h_{xx}^* \\
0 \\
g_{xx}^*
\end{bmatrix}.
\]

\(M_{xx}^*\) can be thought of as a rearrangement of \(M_{xx}\).

---

7. More specifically, each row is the transpose of the vec of the policy function for each variable.

8. Note that \(M_x\) is the same as the matrix \(M\) in Gomme & Klein (2011).
The matrix $D$ is the gradient matrix for $f$ (equation 1):\(^9\)

$$
D = \begin{bmatrix}
\frac{\partial f_1}{\partial x_{1,t+1}} & \cdots & \frac{\partial f_1}{\partial y_{1,t+1}} & \cdots & \frac{\partial f_n}{\partial x_{1,t+1}} & \cdots & \frac{\partial f_n}{\partial y_{n,y,t+1}} \\
\vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\
\frac{\partial f_1}{\partial x_{1,t+1}} & \cdots & \frac{\partial f_n}{\partial y_{1,t+1}} & \cdots & \frac{\partial f_1}{\partial y_{n,y,t+1}} & \cdots & \frac{\partial f_n}{\partial y_{n,y,t+1}} \\
\end{bmatrix}
$$

The matrix $H$ is the Hessian of $f$ (equation 1):\(^10\)

$$
H = \begin{bmatrix}
\frac{\partial^2 f_1}{\partial x_{1,t+1}^2} & \cdots & \frac{\partial^2 f_1}{\partial x_{1,t+1} \partial y_{1,y,t+1}} & \cdots & \frac{\partial^2 f_n}{\partial x_{1,t+1}^2} & \cdots & \frac{\partial^2 f_n}{\partial x_{1,t+1} \partial y_{n,y,t+1}} \\
\vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\
\frac{\partial^2 f_1}{\partial x_{1,t+1} \partial y_{1,y,t+1}} & \cdots & \frac{\partial^2 f_1}{\partial x_{1,t+1}^2} & \cdots & \frac{\partial^2 f_n}{\partial x_{1,t+1} \partial y_{n,y,t+1}} & \cdots & \frac{\partial^2 f_n}{\partial x_{1,t+1} \partial y_{n,y,t+1}} \\
\end{bmatrix}
$$

I introduce $T$ which is the matrix of third derivatives of $f$:

$$
T = \begin{bmatrix}
\frac{\partial^3 f_1}{\partial x_{1,t+1}^3} & \cdots & \frac{\partial^3 f_1}{\partial x_{1,t+1} \partial y_{1,y,t+1}} & \cdots & \frac{\partial^3 f_n}{\partial x_{1,t+1}^3} & \cdots & \frac{\partial^3 f_n}{\partial x_{1,t+1} \partial y_{n,y,t+1}} \\
\vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\
\frac{\partial^3 f_1}{\partial x_{1,t+1} \partial y_{1,y,t+1}} & \cdots & \frac{\partial^3 f_1}{\partial x_{1,t+1}^2} & \cdots & \frac{\partial^3 f_n}{\partial x_{1,t+1} \partial y_{n,y,t+1}} & \cdots & \frac{\partial^3 f_n}{\partial x_{1,t+1} \partial y_{n,y,t+1}} \\
\end{bmatrix}
$$

### 3.1.2. Solution

I differentiate equation (6) with respect to $x_t$ 3 times and evaluate the third derivatives at the non-stochastic steady state (that is $x = 0$ and $\sigma = 0$). Applying Theorem 1 (see Appendix A), which is the matrix representation of the third-order chain rule, gives

\(^9\) $D$ is the same as the matrix $D$ in Gomme & Klein (2011).

\(^10\) $H$ is the same as the matrix $H$ in Gomme & Klein (2011).
\[
\left( I \otimes M'_x \otimes M'_x \right) T M_x + \left( I \otimes (M_{xx}^*)' \right) H M_x + \\
\left( I \otimes M'_x \otimes I \right) \left( H \otimes I \right) M_{xx} + \\
\left( I \otimes (M^*_x)' \right) \left( H \otimes I \right) M_{xx} + \\
\left( D \otimes I \right) \left[ \begin{array}{c} h_{xxx} \\ h_{xxx} \\ g_{xx} \\ h_x \\ h_x^* \\ h \end{array} \right] + \\
\left( D \otimes I \right) \left[ \begin{array}{c} h_{xxx} \\ h_{xxx} \\ g_{xx} \\ h_x \\ h_x^* \\ h \end{array} \right] = 0,
\]

(7)

where

\[
K = \left( I \otimes h'_x \otimes I \right) \left( g_{xx} \otimes I \right) h_{xx} \\
+ \left( I \otimes (h^*_x)' \right) \left( g_{xx} \otimes I \right) h_{xx} + \left( I \otimes (h_{xx}^*)' \right) g_{xx} h_x
\].

Following Gomme & Klein (2011) I use the partition \( D = \left[ \begin{array}{c} d_1, d_2, d_3, d_4 \end{array} \right] \) to rewrite equation (7) as

\[
\left( I \otimes M'_x \otimes M'_x \right) T M_x + \left( I \otimes (M_{xx}^*)' \right) H M_x + \\
\left( I \otimes M'_x \otimes I \right) \left( H \otimes I \right) M_{xx} + \left( I \otimes (M^*_x)' \right) \left( H \otimes I \right) M_{xx} + \\
\left[ \begin{array}{c} \left( I \otimes h'_x \otimes h'_x \right) g_{xx} h_x + \left( g_x \otimes I \right) h_{xxx} + K \end{array} \right] = 0.
\]

(8)

Rearranging equation (8) gives
\[
\begin{align*}
A_1 & = \left( I \otimes M_x' \otimes M_x' \right) TM_x + \left( I \otimes (M_x')^t \right) HM_x + \\
& \quad \left( I \otimes M_x' \otimes I \right) \left( H \otimes I \right) M_x + \\
& \quad \left( I \otimes (M_x')^t \right) \left( H \otimes I \right) M_x + \left( d_2 \otimes I \right) K.
\end{align*}
\]

where

\[
\begin{align*}
A_1 & = \left( I \otimes M_x' \otimes M_x' \right) TM_x + \left( I \otimes (M_x')^t \right) HM_x + \\
& \quad \left( I \otimes M_x' \otimes I \right) \left( H \otimes I \right) M_x + \\
& \quad \left( I \otimes (M_x')^t \right) \left( H \otimes I \right) M_x + \left( d_2 \otimes I \right) K.
\end{align*}
\]

Applying the vec operator to both sides of (9) gives

\[
\begin{align*}
\text{vec}(A_1) + \left( I \otimes B \right) \text{vec}(h_{\text{xxx}}) + \\
\left( C + I \otimes d_4 \otimes I \right) \text{vec}(g_{\text{xxx}}) = 0,
\end{align*}
\]

where

\[
B = \left( d_1 \otimes I \right) + \left( d_2 \otimes I \right) \left( g_x \otimes I \right),
\]

and

\[
C = h_{\text{xx}}' \otimes \left( \left( d_2 \otimes I \right) \left( I \otimes h_{\text{xx}}' \otimes h_{\text{xx}}' \right) \right).
\]

Equation (10) be written as the linear system

\[
\begin{bmatrix}
C + \left( I \otimes d_4 \otimes I \right), & I \otimes B
\end{bmatrix}
\begin{bmatrix}
\text{vec}(g_{\text{xxx}}) \\
\text{vec}(h_{\text{xxx}})
\end{bmatrix} = -\text{vec}(A_1)
\]

This is easily solved using matrix algebra. Alternatively equation (9) could have been written in the form of a generalised Sylvester equation and solved as explained in Gomme & Klein (2011) (see Appendix B). This second approach is computationally more efficient and uses less memory.

---

\(^{11}\)Using vec\((XYZ) = (Z' \otimes X)\text{vec}(Y)\).
3.2. Solving for \( g_{\sigma x} \) and \( h_{\sigma x} \)

I have found \( g_{xx} \) and \( h_{xx} \) which can be used, along with \( g_x, h_x, g_{xx}, h_{xx}, g_\sigma \) and \( h_\sigma \), to solve for \( g_{\sigma x} \) and \( h_{\sigma x} \).

Before I begin I define some additional matrices that will prove useful for the solution.

3.2.1. Matrix definitions

I define

\[
N_\sigma = \begin{bmatrix}
  I_{n_x \times n_x} \\
g_x \\
  0_{n \times n_x}
\end{bmatrix},
\]

I define

\[
N^*_\sigma = \begin{bmatrix}
  0_{n_x \times n_x^2} \\
g^*_xx \\
  h_x \otimes I_{n_x \times n_x} \\
  0_{n \times n_x^2}
\end{bmatrix},
\]

which is a matrix of second-derivatives of the policy functions with respect to \( \sigma \) and \( x \).

I also define the prediction error variance-covariance matrix:

\[
\Sigma = \begin{bmatrix}
  \sigma^2_1 & \cdots & \sigma_{1,n_x} \\
  \vdots & \ddots & \vdots \\
  \sigma_{n_x,1} & \cdots & \sigma^2_{n_x}
\end{bmatrix},
\]

where \( \sigma^2_i \) is the variance of the prediction error of \( i \)th predetermined variable. Likewise, \( \sigma_{i,j} \) is the covariance between the prediction errors for the \( i \)th and \( j \)th predetermined variables.

I also introduce the the matrix trace (trm). This is defined (as in Gomme & Klein 2011) so that for an \( nm \times n \) matrix:

\[
\begin{bmatrix}
  Y'_1 & Y'_2 & \cdots & Y'_m
\end{bmatrix},
\]

the matrix trace gives an \( m \times 1 \) vector:

\[
\begin{bmatrix}
  \text{tr}(Y_1) & \text{tr}(Y_2) & \cdots & \text{tr}(Y_m)
\end{bmatrix}.
\]

\(^{12}N_\sigma \) is the same as the matrix \( N \) in Gomme & Klein (2011).
3.2.2. Solution

I differentiate equation (6) with respect to $\sigma$ twice and $x_t$ once. Using Theorem 1 (see Appendix A) the matrix representation of the solution takes the form

\[
\text{tr} \left( \left( I_{n \times n} \otimes M_x' \otimes N_{\sigma}' \right) T N_{\sigma} \Sigma \right) + 2 \times \text{tr} \left( \left( I_{n \times n} \otimes (N_{\sigma}^*)' \right) H N_{\sigma} \Sigma \right)
\]

\[
+ \left( I_{n \times n} \otimes M_x' \right) H \left[ \text{tr} \left( \left( I_{ny \times ny} \otimes \Sigma \right) g_{xx} \right) + g_x h_{\sigma} + g_{\sigma} \right]
\]

\[
+ \left( D \otimes I_{nx \times nx} \right) \begin{bmatrix} h_{\sigma x} \\ n x^2 \times 1 \\ P \\ ny \times nx \times 1 \\ g_{\sigma x} \\ ny \times nx \times 1 \end{bmatrix} = 0, \quad (12)
\]

where

\[
P = \left( I_{ny \times ny} \otimes h_x' \right) g_{xx} h_{\sigma} + \left( g_x \otimes I_{ny \times nx} \right) h_{\sigma x} + \left( I_{ny \times ny} \otimes h_x' \right) g_{\sigma x} + \text{tr} \left( \left( I_{ny \times ny} \otimes \Sigma \right) \left( I_{ny \times ny} \otimes h_x' \otimes I_{nx \times nx} \right) g_{xxx} \right).
\]

Substituting $D = [d_1, d_2, d_3, d_4]$ into equation (12) and rearranging gives

\[
A_2 + \left( d_1 \otimes I_{nx \times nx} \right) h_{\sigma x} + \left( d_2 \otimes I_{nx \times nx} \right) \left( g_x \otimes I_{nx \times nx} \right) h_{\sigma x} 
\]

\[
+ \left( d_2 \otimes I_{nx \times nx} \right) \left( I_{ny \times ny} \otimes h_x' \right) g_{\sigma x} + \left( d_4 \otimes I_{nx \times nx} \right) g_{\sigma x} = 0, \quad (13)
\]

where
\[ A_2 = \text{trm} \left( \left( I_{n \times n} \otimes M'_{x} \otimes N'_{\sigma} \right) T N_\sigma \Sigma \right) + 2 \times \text{trm} \left( \left( I_{n \times n} \otimes N'_{\sigma} \right) H N_\sigma \Sigma \right) + \left( I_{n \times n} \otimes M'_{x} \right) H \left[ \begin{array}{c} h_{\sigma} \\ g_{xx} \\ 0 \\ g_{\sigma} \\ g_{\sigma \sigma} \end{array} \right] + \left( d_2 \otimes I_{n \times n \times n} \right) \left[ \begin{array}{c} h'_{x} \end{array} \right] g_{xx} h_{\sigma} + \text{trm} \left( \left( I_{n \times n \times n} \otimes \Sigma_{n \times n \times n} \right) \left( I_{n \times n} \otimes h'_{x} \otimes I_{n \times n \times n} \right) g_{xx} \right). \]

Equation (13) can be written as the linear system

\[ Q_{n \times n \times n \times n} \left[ \begin{array}{c} g_{\sigma x} \\ h_{\sigma x} \end{array} \right] = -A_2, \quad (14) \]

where

\[ Q = \left[ \left( d_2 \otimes I_{n \times n \times n} \right) \left( I_{n \times n \times n} \otimes h'_{x} \right) + \left( d_4 \otimes I_{n \times n \times n} \right), \right. \]

\[ \left. \left( d_1 \otimes I_{n \times n \times n} \right) + \left( d_2 \otimes I_{n \times n \times n} \right) \left( g_{x} \otimes I_{n \times n \times n} \right) \right] \]

which is easily solved using matrix algebra.

### 3.3. Solving for \( g_{\sigma x} \) and \( h_{\sigma x} \)

In this section I solve for \( g_{\sigma x} \) and \( h_{\sigma x} \) using some of the previous results. Before I do this, I define some additional matrices.

#### 3.3.1. Matrix definitions

I define the matrix

\[ N_{\sigma}^{*} = \left[ \begin{array}{c} 0 \\ g_{xx}^{*} \\ 0 \end{array} \right], \]

which is a matrix of the second derivatives of the policy functions with respect to \( \sigma \). I also define the skewness (co-skewness) matrix:
The skewness matrix contains the third moments of the prediction errors, where $s_i = E_t [u_{i,t}^3]$, $s_{i,j,k} = E_t [u_{i,t}u_{j,t}u_{k,t}]$, and $u_{i,t}$ is the prediction error for the $i$th predetermined variable. This follows from the definition of the variance-covariance matrix: $\Sigma = E_t [u_t \otimes u_t']$, so that $S = E_t [u_t \otimes u_t' \otimes u_t']$, 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.

### 3.3.2. Solution

I differentiate equation (6) with respect to $\sigma$ 3 times. Using Theorem 1 (see Appendix A) I can write the matrix representation of this solution in the following way:

$$
\text{trm} \left( \left( \begin{array}{c}
I \otimes N'_{\sigma} \otimes N'_{\sigma} \\
\end{array} \right) T \Sigma^{-1} \right) + 3 \times \text{trm} \left( \left( \begin{array}{c}
I \otimes (N'_{\sigma})' \\
\end{array} \right) H \Sigma^{-1} \right) \\
+ D \left[ \begin{array}{c}
h_{\sigma\sigma} \\
0 \\
g_{\sigma\sigma} \\
\end{array} \right] = 0. 
$$

(15)

Substituting $D = [d_1, d_2, d_3, d_4]$ into equation (15) and rearranging gives the linear system

$$
[d_2 + d_4, d_1 + d_2g_x] \left[ \begin{array}{c}
g_{\sigma\sigma} \\
h_{\sigma\sigma} \\
\end{array} \right] = -A_3, 
$$

(16)

where

$$
A_3 = \text{trm} \left( \left( \begin{array}{c}
I \otimes N'_{\sigma} \otimes N'_{\sigma} \\
\end{array} \right) T \Sigma^{-1} \right) + 3 \times \text{trm} \left( \left( \begin{array}{c}
I \otimes (N'_{\sigma})' \\
\end{array} \right) H \Sigma^{-1} \right) + d_2 \times \text{trm} \left( \left( \begin{array}{c}
I \otimes S \\
\end{array} \right) g_{xxx} \right). 
$$

Equation (16) can then be solved using matrix algebra.
4. A simple example

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

\[
0 = c_t - \gamma - \beta E_t \left\{ \left( 1 + \alpha a_{t+1} k_{t+1}^{\alpha - 1} - \delta \right) c_{t+1} \right\}
\]
\[
0 = k_t + c_t - a_t k_{t-1} - (1 - \delta) k_{t-1}
\]
\[
0 = a_t - a_r^{t-1} \exp(\sigma \varepsilon_t),
\]

with the non-stochastic steady states: \( a = 1, k = \left[ \frac{a_0}{1 - \beta (1 - \delta)} \right]^{\frac{1}{1 - \alpha}}, c = a k^\alpha - \delta k, \varepsilon = 0. \)

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: \( \hat{a}_t = \log(a_t), \hat{k}_t = \log(k_t), \hat{c}_t = \log(c_t), \hat{\varepsilon}_t = 0, \hat{a}^*_t = \log(a_t).^{13} \) In addition, I include an auxiliary variable for technology 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:

\[
0 = \exp(\hat{c}_{t}) - \gamma - \beta \left( 1 + \alpha \exp(\hat{a}^{t+1}_{t}) \exp(\hat{k}_{t})^{\alpha - 1} - \delta \right) \exp(\hat{c}_{t+1})^{-\gamma}
\]
\[
0 = \exp(\hat{k}_{t}) + \exp(\hat{c}_{t}) - \exp(\hat{a}_{t}) \exp(\hat{k}_{t-1})^{\alpha} - (1 - \delta) \exp(\hat{k}_{t-1})
\]
\[
0 = \hat{a}_{t} - \rho \hat{a}_{t-1} - \sigma \hat{\varepsilon}_{t}
\]
\[
0 = \hat{a}^*_{t} - \hat{a}_{t}
\]
\[
0 = \hat{\varepsilon}_{t+1},
\]

with the non-stochastic steady states: \( \hat{a} = \log(a), \hat{k} = \log(k), \hat{c} = \log(c), \hat{\varepsilon} = 0, \hat{a}^* = \log(a). \)

The vector of predetermined variables takes the form

\[
x_t = \left[ \begin{array}{c} \hat{k}_{t-1} \hat{a}_{t-1} \hat{\varepsilon}_{t} \end{array} \right]'.
\]

The vector of non-predetermined variables takes the form

\[
y_t = \left[ \begin{array}{c} \hat{c}_{t} \hat{a}^*_{t} \end{array} \right]'.
\]

I also define the variance-covariance matrix,

\[
\Sigma = \left[ \begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma^2 \end{array} \right],
\]

\(^{13}\)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.
and a skewness matrix,

\[
S = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2
\end{bmatrix}
\]

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.\(^{14}\)

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 method outlined in this paper. The results are presented in Appendix C 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.\(^{15}\)

I also 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 in total when auxiliary variables are included) New Keynesian DSGE model. The times (in seconds) from both experiments are recorded in the table below:

<table>
<thead>
<tr>
<th>Model</th>
<th>Tensors</th>
<th>Without Tensors</th>
<th>Without Tensors (Opt)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBC</td>
<td>0.1211</td>
<td>0.0040</td>
<td>0.0037</td>
</tr>
<tr>
<td>NK DSGE</td>
<td>1.1288</td>
<td>0.0447</td>
<td>0.0137</td>
</tr>
</tbody>
</table>

Table 1: Computation Times

My solution method took 0.0040 seconds to find the third-order terms: \(g_{xxx}, h_{xxx}, g_{\sigma x}, h_{\sigma x}, g_{\sigma \sigma} \) and \(h_{\sigma \sigma}\), 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

\(^{14}\)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.

\(^{15}\)I have also checked my solution method against Dynare and Andreasen’s code using other small DSGE models.
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.\footnote{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 $t+1$).} This results in further performance improvements. 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 for my method the extra time required to shrink the matrices is more than the time saved in the matrix division.\footnote{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.} 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.

5. Conclusion

In this paper I demonstrate 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.

Appendix A.

In this appendix I show that my matrix representation is indeed a solution to the third-order matrix chain rule.

I begin by defining some function $g$ that is an n-ary function of $f$. $f$ is an m-ary function of $x$ so that

$$y = g\left(f^1(x), \cdots, f^n(x)\right)$$

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

$$x = [x_1, \cdots, x_m]$$

By Fáa di Bruno’s formula, the third derivative of $y$ with respect to the $i$th, $j$th and $k$th elements in $x$ is given by
\[
\frac{\partial^3 y}{\partial x_i \partial x_j \partial x_k} = \sum_{a=1}^n \sum_{b=1}^n \sum_{c=1}^n \frac{\partial^3 g}{\partial f^a \partial f^b \partial f^c} \left( \frac{\partial f^a}{\partial x_i} \right) \left( \frac{\partial f^b}{\partial x_j} \right) \left( \frac{\partial f^c}{\partial x_k} \right) + \\
\sum_{a=1}^n \sum_{b=1}^n \sum_{c=1}^n \frac{\partial^2 g}{\partial f^a \partial f^b} \left( \frac{\partial^2 f^a}{\partial x_i \partial x_j} \right) \left( \frac{\partial f^b}{\partial x_k} \right) + \\
\sum_{a=1}^n \sum_{b=1}^n \sum_{c=1}^n \frac{\partial^2 g}{\partial f^a \partial f^b} \left( \frac{\partial^2 f^a}{\partial x_i \partial x_k} \right) \left( \frac{\partial f^b}{\partial x_j} \right) + \\
\sum_{a=1}^n \sum_{b=1}^n \sum_{c=1}^n \frac{\partial g}{\partial f^a} \left( \frac{\partial^3 f^a}{\partial x_i \partial x_j \partial x_k} \right)
\]

for any \( i, j, k \in \{1, \ldots, m\} \) and \( a, b, c \in \{1, \ldots, n\} \).

This can be written more compactly as

\[
y_{i,j,k} = \sum_{a=1}^n \sum_{b=1}^n \sum_{c=1}^n g_{a,b,c} f_i^a f_j^b f_k^c + \sum_{a=1}^n \sum_{b=1}^n g_{a,b} f_{i,j,k}^a f_j^b + \sum_{a=1}^n \sum_{b=1}^n g_{a,b} f_{i,k}^a f_k^b + \sum_{a=1}^n \sum_{b=1}^n g_{a,b} f_{j,k}^a f_j^b + \sum_{a=1}^n g_{a} f_{i,j,k}^a
\]

where

\[
y_{i,j,k} = \frac{\partial^3 y}{\partial x_i \partial x_j \partial x_k},
\]

\[
\sum_{a=1}^n \sum_{b=1}^n \sum_{c=1}^n g_{a,b,c} f_i^a f_j^b f_k^c = \sum_{a=1}^n \sum_{b=1}^n \sum_{c=1}^n \frac{\partial^3 g}{\partial f^a \partial f^b \partial f^c} \left( \frac{\partial f^a}{\partial x_i} \right) \left( \frac{\partial f^b}{\partial x_j} \right) \left( \frac{\partial f^c}{\partial x_k} \right),
\]

\[
\sum_{a=1}^n \sum_{b=1}^n \sum_{c=1}^n g_{a,b} f_{i,j,k}^a f_j^b = \sum_{a=1}^n \sum_{b=1}^n \frac{\partial^2 g}{\partial f^a \partial f^b} \left( \frac{\partial^2 f^a}{\partial x_i \partial x_j} \right) \left( \frac{\partial f^b}{\partial x_k} \right),
\]

\[
\sum_{a=1}^n \sum_{b=1}^n \sum_{c=1}^n g_{a,b} f_{i,k}^a f_k^b = \sum_{a=1}^n \sum_{b=1}^n \frac{\partial^2 g}{\partial f^a \partial f^b} \left( \frac{\partial^2 f^a}{\partial x_i \partial x_k} \right) \left( \frac{\partial f^b}{\partial x_j} \right),
\]

\[
\sum_{a=1}^n \sum_{b=1}^n g_{a} f_{i,j,k}^a = \sum_{a=1}^n \frac{\partial g}{\partial f^a} \left( \frac{\partial^3 f^a}{\partial x_i \partial x_j \partial x_k} \right).
\]

I define the matrix \( S \) as the matrix representation of all possible combinations of the third derivative of \( y \) with respect to each element \( x_i \) in \( \mathbf{x} \) so that

16
$$S = \begin{bmatrix}
\tilde{S}_1 \\
\vdots \\
\tilde{S}_k \\
\vdots \\
\tilde{S}_m
\end{bmatrix}$$

Where

$$\tilde{S}_k = \begin{bmatrix}
y_{1,1,k} & \cdots & y_{m,1,k} \\
\vdots & \ddots & \vdots \\
y_{1,m,k} & \cdots & y_{m,m,k}
\end{bmatrix}$$

Let $s_{r,c}$ represent the element in row $r$ and column $c$ in $S$. Alternatively let $s_{j+m(k-1),i}$ refer to the element in the $j+m(k-1)$th row and the $i$th column of $S$ where as before $i,j,k \in \{1, \cdots, m\}$. This alternative indexation allows me to match the location of each element in $S$ to the derivative in that position. For example

$$y_{i,j,k} = s_{j+m(k-1),i}$$

**Theorem 1.** The third order matrix chain rule for $y$ takes the form

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

where the matrices are defined in the following section.

**Appendix A.1. Matrix definitions**

I define $D$ such that

$$D = \begin{bmatrix}
f_1^1 & \cdots & f_1^m \\
\vdots & \ddots & \vdots \\
f_n^1 & \cdots & f_n^m
\end{bmatrix} \tag{A.2}$$

so that

$$f_i^a = d_{a,i}$$

for $i = 1, \cdots, m$ and $a = 1, \cdots, n$, where $d_{a,i}$ represents the element in the $a$th row and the $i$th column of $D$.

I define $Z$:
\[
Z_{n^2 \times n} = \begin{bmatrix}
\tilde{Z}_1 \\
\vdots \\
\tilde{Z}_c \\
\vdots \\
\tilde{Z}_n
\end{bmatrix}
\]

Where

\[
\tilde{Z}_c = \begin{bmatrix}
g_{1,1,c} & \cdots & g_{n,1,c} \\
\vdots & & \vdots \\
g_{1,n,c} & \cdots & g_{n,n,c}
\end{bmatrix}
\]

so that

\[
g_{a,b,c} = z_{b+n(c-1),a}
\]

for \(a, b, c \in \{1, \cdots, n\}\), where \(z_{b+n(c-1),a}\) is the element in the \(b+n(c-1)\)th row and the \(a\)th column of the matrix \(Z\).

I define \(P\) such that

\[
P_{n \times m^2} = \begin{bmatrix}
\tilde{P}_1 & \cdots & \tilde{P}_j & \cdots & \tilde{P}_m
\end{bmatrix}
\]

Where

\[
\tilde{P}_j = \begin{bmatrix}
f_{1,j}^1 & \cdots & f_{m,j}^1 \\
\vdots & & \vdots \\
f_{1,j}^n & \cdots & f_{m,j}^n
\end{bmatrix}
\]

so that

\[
f_{i,j}^a = p_{a,i+m(j-1)}
\]

for \(a = 1, \cdots, n\) and \(i, j = 1, \cdots, m\), where \(p_{a,i+m(j-1)}\) is the element in the \(a\)th row and the \(i+m(j-1)\)th column of the matrix \(P\).

I define \(W\) so that

\[
W_{n \times n} = \begin{bmatrix}
g_{1,1} & \cdots & g_{n,1} \\
\vdots & & \vdots \\
g_{1,n} & \cdots & g_{n,n}
\end{bmatrix}
\]

which implies

\[
g_{a,b} = w_{a,b}
\]

for \(a, b = 1, \cdots, n\), where \(w_{a,b}\) is the element in the \(a\)th row and the \(b\)th column of \(W\).
I define $Q$ so that

\[
Q_{n \times m^2} = \begin{bmatrix}
I_{m \times m} \otimes D[1,:]
\vdots
I_{m \times m} \otimes D[n,:]
\end{bmatrix}
= \begin{bmatrix}
f_1^1 & f_1^2 & \cdots & f_1^m & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & f_1^1 & f_1^2 & \cdots & f_1^m & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & f_2^m & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & f_2^1 & f_2^2 & \cdots & f_2^m & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & f_3^m & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & f_3^1 & f_3^2 & \cdots & f_3^m & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & f_n^m & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & f_n^1 & f_n^2 & \cdots & f_n^m & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & f_n^m & 0 & 0 & \cdots & 0 & 0 & \cdots & 0
\end{bmatrix}
\]

which implies

\[
q_{i+n(a-1),j+m(k-1)} = \begin{cases} 
 f_a^i & \text{if } i = k \\
 0 & \text{if } i \neq k 
\end{cases}
\]

for $i, j, k = 1, \cdots, m$ and $a = 1, \cdots, n$, where $q_{i+n(a-1),j+m(k-1)}$ is the element in the $i + n(a - 1)$th row and the $j + m(k - 1)$th column of $Q$.

I define $R$ as

\[
R = [g_1, \cdots, g_n]
\]

so that

\[
g_a = r_{1,a}
\]

for $a = 1, \cdots, n$, where $r_{1,a}$ is the $a$th entry in the row vector $R$.

I define $T$ such that

\[
T_{n \times m^2} = \begin{bmatrix}
\tilde{T}_1 \\
\vdots \\
\tilde{T}_a \\
\vdots \\
\tilde{T}_n
\end{bmatrix}
\]

where
and

\[ \hat{T}_a^{m} = \begin{bmatrix} \hat{T}_1^a \\ \vdots \\ \hat{T}_k^a \\ \vdots \\ \hat{T}_m^a \end{bmatrix} \]

so that

\[ f_{i,j,k}^a = t_{j+m(k-1)+m^2(a-1),i} \]

for \( a = 1, \ldots, n \) and \( i, j, k = 1, \ldots, m \), where \( t_{j+m(k-1)+m^2(a-1),i} \) is the element in \( j + m(k-1) + m^2(a-1) \)th row and the \( i \)th column of the matrix \( T \).

I define \( V \) so that

\[ V^{n\times m} = \begin{bmatrix} \tilde{V}_1 \\ \vdots \\ \tilde{V}_a \\ \vdots \\ \tilde{V}_n \end{bmatrix} \]

where

\[ \tilde{V}_a^{m} = \begin{bmatrix} f_{1,1}^a & \cdots & f_{m,1}^a \\ \vdots & \ddots & \vdots \\ f_{1,m}^a & \cdots & f_{m,m}^a \end{bmatrix} \]

so that

\[ f_{i,j}^a = v_{j+m(a-1),i} \]

for \( a = 1, \ldots, n \) and \( i, j = 1, \ldots, m \), where \( v_{j+m(a-1),i} \) is the element in the \( j + m(a-1) \)th row and the \( i \)th column of the matrix \( V \).

**Proof** The proof proceeds as follows, I define the following matrices
\[
S_1 = (D' \otimes D') ZD \\
S_2 = P' WD \\
S_3 = \left( D' \otimes I_{m \times m} \right) \left( W \otimes I_{m \times m} \right) V \\
S_4 = Q' \left( W \otimes I_{m \times m} \right) V \\
S_5 = \left( R \otimes I_{m^2 \times m^2} \right) T
\]

so that I can rewrite equation (A.1) as

\[
S = S_1 + S_2 + S_3 + S_4 + S_5
\]

To prove the proposition I need to show that for each element in \( S \), the following holds

\[
y_{i,j,k} = s_{1j,m(k-1),i} + s_{2j,m(k-1),i} + s_{3j,m(k-1),i} + s_{4j,m(k-1),i} + s_{5j,m(k-1),i}.
\]

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

\[
s_{1j,m(k-1),i} = \sum_{a=1}^{n} \sum_{b=1}^{n} \sum_{c=1}^{n} g_{a,b,c} f_i^a f_j^b f_k^c \\
s_{2j,m(k-1),i} = \sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_i^a f_j^b f_k^c \\
s_{3j,m(k-1),i} = \sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{i,j}^a f_k^b \\
s_{4j,m(k-1),i} = \sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{i,k}^a f_j^b \\
s_{5j,m(k-1),i} = \sum_{a=1}^{n} g_{a} f_{i,j,k}^a
\]

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 sub matrix the indexation matches up with the appropriate derivatives.
Step 1

From equation (A.4) $S_1 = (D’ \otimes D’) ZD$. I need to show that $s_{1+m(k-1),i} = \sum_{a=1}^{n} \sum_{b=1}^{n} \sum_{c=1}^{n} g_{a,b,c} f_{a,i}^{b} f_{k,j}^{c} f_{a}^{1} f_{b}^{1} f_{c}^{1}$

I define $\Theta_1$ so that

$$\Theta_1 = D’ \otimes D’ = \begin{bmatrix}
    f_1^1 f_1^1 & f_1^1 f_2^1 & \cdots & f_1^1 f_n^1 \\
    f_1^2 f_2^1 & f_2^2 f_1^1 & \cdots & f_1^2 f_n^1 \\
    \vdots & \vdots & \ddots & \vdots \\
    f_m^1 f_m^1 & f_m^2 f_m^1 & \cdots & f_m^n f_m^1 \\
    f_m^1 f_m^2 & f_m^2 f_m^2 & \cdots & f_m^n f_m^2 \\
    \vdots & \vdots & \ddots & \vdots \\
    f_m^1 f_m^n & f_m^2 f_m^n & \cdots & f_m^n f_m^n
\end{bmatrix}$$

where

$$\theta_{1,j+m(k-1),b+n(c-1)} = f_{k,j}^{b} f_{j}^{c}$$

is the element in the $j + m(k-1)$th row and the $b + n(c-1)$th column of $\Theta_1$ for $j, k = 1, \cdots, m$ and $b, c = 1, \cdots, n$.

I can then define $\Theta_2$ to be

$$\Theta_2 = \Theta_1 Z = \begin{bmatrix}
    \sum_{b=1}^{n} \sum_{c=1}^{n} g_{1,b,c} f_{1}^{b} f_{1}^{c} & \sum_{b=1}^{n} \sum_{c=1}^{n} g_{2,b,c} f_{1}^{b} f_{2}^{c} & \cdots & \sum_{b=1}^{n} \sum_{c=1}^{n} g_{n,b,c} f_{1}^{b} f_{1}^{c} \\
    \sum_{b=1}^{n} \sum_{c=1}^{n} g_{1,b,c} f_{2}^{b} f_{1}^{c} & \sum_{b=1}^{n} \sum_{c=1}^{n} g_{2,b,c} f_{2}^{b} f_{2}^{c} & \cdots & \sum_{b=1}^{n} \sum_{c=1}^{n} g_{n,b,c} f_{2}^{b} f_{2}^{c} \\
    \vdots & \vdots & \ddots & \vdots \\
    \sum_{b=1}^{n} \sum_{c=1}^{n} g_{1,b,c} f_{m}^{b} f_{m}^{c} & \sum_{b=1}^{n} \sum_{c=1}^{n} g_{2,b,c} f_{m}^{b} f_{m}^{c} & \cdots & \sum_{b=1}^{n} \sum_{c=1}^{n} g_{n,b,c} f_{m}^{b} f_{m}^{c} \\
    \vdots & \vdots & \ddots & \vdots \\
    \sum_{b=1}^{n} \sum_{c=1}^{n} g_{1,b,c} f_{m}^{b} f_{m}^{c} & \sum_{b=1}^{n} \sum_{c=1}^{n} g_{2,b,c} f_{m}^{1} f_{m}^{1} & \cdots & \sum_{b=1}^{n} \sum_{c=1}^{n} g_{n,b,c} f_{m}^{b} f_{m}^{c}
\end{bmatrix}$$
where

\[ \theta_{2j+m(k-1),a} = \sum_{b=1}^{n} \sum_{c=1}^{n} g_{a,b,c} f_k^b f_j^c \]

is the element in the \( j + m(k - 1) \)th row and the \( a \)th column of \( \Theta_2 \) for \( j, k = 1, \cdots, m \) and \( a = 1, \cdots, n \).

The matrix \( D \) as defined in (A.2)

\[
D_{n \times m} = \begin{bmatrix}
  f_1^1 & \cdots & f_i^1 & \cdots & f_m^1 \\
  \vdots & \ddots & \vdots & \ddots & \vdots \\
  f_1^a & \cdots & f_i^a & \cdots & f_m^a \\
  \vdots & \ddots & \vdots & \ddots & \vdots \\
  f_1^n & \cdots & f_i^n & \cdots & f_m^n 
\end{bmatrix}
\]

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

\[ f_i^a = d_{a,i} \]

Multiplying \( \Theta_2 \) by \( D \) gives \( S_1 \)

\[
S_1 = \begin{bmatrix}
  \sum_{a=1}^{n} \sum_{b=1}^{n} \sum_{c=1}^{n} g_{a,b,c} f_k^b f_j^c f_1^a & \sum_{a=1}^{n} \sum_{b=1}^{n} \sum_{c=1}^{n} g_{a,b,c} f_k^b f_1^c f_2^a & \cdots & \sum_{a=1}^{n} \sum_{b=1}^{n} \sum_{c=1}^{n} g_{a,b,c} f_k^b f_j^c f_m^a \\
  \vdots & \ddots & \vdots & \ddots & \vdots \\
  \sum_{a=1}^{n} \sum_{b=1}^{n} \sum_{c=1}^{n} g_{a,b,c} f_k^b f_1^c f_2^a & \sum_{a=1}^{n} \sum_{b=1}^{n} \sum_{c=1}^{n} g_{a,b,c} f_k^b f_2^c f_2^a & \cdots & \sum_{a=1}^{n} \sum_{b=1}^{n} \sum_{c=1}^{n} g_{a,b,c} f_k^b f_j^c f_m^a \\
  \vdots & \ddots & \vdots & \ddots & \vdots \\
  \sum_{a=1}^{n} \sum_{b=1}^{n} \sum_{c=1}^{n} g_{a,b,c} f_k^b f_m^c f_1^a & \sum_{a=1}^{n} \sum_{b=1}^{n} \sum_{c=1}^{n} g_{a,b,c} f_k^b f_m^c f_2^a & \cdots & \sum_{a=1}^{n} \sum_{b=1}^{n} \sum_{c=1}^{n} g_{a,b,c} f_k^b f_m^c f_m^a 
\end{bmatrix}
\]

From the indexation in equation (A.8) it can be verified that\(^{18}\)

\[ S_{1j+m(k-1),i} = \sum_{a=1}^{n} \sum_{b=1}^{n} \sum_{c=1}^{n} g_{a,b,c} f_i^a f_k^b f_j^c \]

\(^{18}\)The ordering of the derivatives of the \( f \) functions does not matter because these are scalars.
Step 2

From equation (A.5) $S^2 = P'WD$. I need to show that $s_{j+m(k-1),i} = \sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{i}^{a} f_{j,k}^{b}$.

I define $\Theta_3 = P'W$, so that

$$\Theta_1 = P'W = \begin{bmatrix}
\sum_{b=1}^{n} g_{1,b} f_{1,1}^{b} & \sum_{b=1}^{n} g_{2,b} f_{1,1}^{b} & \cdots & \sum_{b=1}^{n} g_{n,b} f_{1,1}^{b} \\
\vdots & \vdots & \ddots & \vdots \\
\sum_{b=1}^{n} g_{1,b} f_{1,2}^{b} & \sum_{b=1}^{n} g_{2,b} f_{1,2}^{b} & \cdots & \sum_{b=1}^{n} g_{n,b} f_{1,2}^{b} \\
\vdots & \vdots & \ddots & \vdots \\
\sum_{b=1}^{n} g_{1,b} f_{j,k}^{b} & \sum_{b=1}^{n} g_{2,b} f_{j,k}^{b} & \cdots & \sum_{b=1}^{n} g_{n,b} f_{j,k}^{b} \\
\vdots & \vdots & \ddots & \vdots \\
\sum_{b=1}^{n} g_{1,b} f_{m,m}^{b} & \sum_{b=1}^{n} g_{2,b} f_{m,m}^{b} & \cdots & \sum_{b=1}^{n} g_{n,b} f_{m,m}^{b}
\end{bmatrix}$$

where

$$\theta_{3j+m(k-1),a} = \sum_{b=1}^{n} g_{a,b} f_{j,k}^{b}$$

is the element in the $j + m(k-1)$th row and the $a$th column of $\Theta_3$, for $j, k = 1, \cdots, m$ and $a = 1, \cdots, n$.

$$S^2 = \Theta_3 D$$
so that

\[ s_{2j+m(k-1),i} = \sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{i,j}^{a} f_{j,k}^{b} \]

as required.

**Step 3**

From equation (A.6) \[ S_3 = \left( D' \otimes I_{m \times m} \right) \left( W \otimes I_{m \times m} \right) V \]. I need to show that \[ s_{3j+m(k-1),i} = \sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{i,j}^{a} f_{j,k}^{b} \].

I define \( \Theta_4 \) so that
\[ \Theta_{4}^{m \times n, m} = \left( D' \otimes I_{m \times m} \right) \left( W \otimes I_{m \times m} \right) = \]

\[
\begin{bmatrix}
\sum_{b=1}^{n} g_{1,b} f_{b} & 0 & 0 & \cdots & 0 & \cdots & \sum_{b=1}^{n} g_{n,b} f_{b} & 0 & 0 & \cdots & 0 \\
0 & \sum_{b=1}^{n} g_{1,b} f_{b} & 0 & \cdots & 0 & \cdots & \sum_{b=1}^{n} g_{n,b} f_{b} & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & \cdots & \sum_{b=1}^{n} g_{1,b} f_{b} & \cdots & 0 & 0 & \cdots & \cdots & \sum_{b=1}^{n} g_{n,b} f_{b} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & \cdots & \sum_{b=1}^{n} g_{1,b} f_{b} & \cdots & 0 & 0 & \cdots & \cdots & \sum_{b=1}^{n} g_{n,b} f_{b} \\
0 & 0 & \cdots & \cdots & \sum_{b=1}^{n} g_{1,b} f_{b} & \cdots & 0 & 0 & \cdots & \cdots & \sum_{b=1}^{n} g_{n,b} f_{b} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & \cdots & \sum_{b=1}^{n} g_{1,b} f_{b} & \cdots & 0 & 0 & \cdots & \cdots & \sum_{b=1}^{n} g_{n,b} f_{b} \\
\end{bmatrix}
\]

where

\[ \theta_{4,j+m(k-1),i+n(a-1)} = \begin{cases} 
\sum_{b=1}^{n} g_{a,b} f_{b} & \text{if } i = j \\
0 & \text{if } i \neq j 
\end{cases} \]

is the element in the \( j + m(k - 1) \)th row and the \( i + n(a - 1) \)th column of \( \Theta_{4} \), for \( i, j, k = 1, \cdots, m \) and \( a = 1, \cdots, n \).

Using the definition of \( S3 \) I can write

\[ S3 = \Theta_{4} V \]
\[
S3 = \begin{bmatrix}
\sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{1,1}^{a} & \sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{1,2}^{a} & \cdots & \sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{1,m,1} \\
\sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{1,1}^{a} & \sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{1,2}^{a} & \cdots & \sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{1,m,2} \\
\vdots & \vdots & \ddots & \vdots \\
\sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{m,1}^{a} & \sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{m,2}^{a} & \cdots & \sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{m,m}^{a}
\end{bmatrix}
\]

where

\[
s_{3j+m(k-1),i} = \sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{i,j}^{a} f_{j,k}^{b}
\]
as required.

**Step 4**

From equation (A.7) \( S4 = Q' \left( W \otimes I_{m \times m} \right) V \). I need to show \( s_{4j+m(k-1),i} = \sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{i,k}^{a} f_{j}^{b} \).

From the definition of \( Q' \)

\[
Q' = \begin{bmatrix}
f_{1}^{1} & 0 & \cdots & 0 & f_{1}^{2} & 0 & \cdots & f_{1}^{n} & 0 & \cdots & 0 \\
f_{2}^{1} & 0 & \cdots & 0 & f_{2}^{2} & 0 & \cdots & f_{2}^{n} & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
f_{m}^{1} & 0 & \cdots & 0 & f_{m}^{2} & 0 & \cdots & f_{m}^{n} & 0 & \cdots & 0 \\
0 & f_{1}^{1} & \cdots & 0 & 0 & f_{1}^{2} & \cdots & 0 & f_{1}^{n} & \cdots & 0 \\
0 & f_{2}^{1} & \cdots & 0 & 0 & f_{2}^{2} & \cdots & 0 & f_{2}^{n} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & f_{m}^{1} & \cdots & 0 & 0 & f_{m}^{2} & \cdots & 0 & f_{m}^{n} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & f_{m}^{1} & 0 & 0 & \cdots & f_{m}^{2} & 0 & \cdots & f_{m}^{n}
\end{bmatrix}
\]

where

\[
q_{j+m(k-1),i+n(a-1)} = \begin{cases}
f_{j}^{a} & \text{if } i = k \\
0 & \text{if } i \neq k
\end{cases}
\]
is the element in the \( j + m(k - 1) \)th row and the \( i + n(a - 1) \)th column of \( Q' \), for 
\( i, j, k = 1, \cdots, m \) and \( a = 1, \cdots, n \).

The Kronecker product of \( W \) and the \( m \times m \) identity matrix is given by

\[
W \otimes I_{m \times m} = \begin{bmatrix}
  g_{1,1} & 0 & \cdots & 0 & \cdots & 0 & g_{n,1} & 0 & \cdots & 0 \\
  0 & g_{1,1} & \cdots & 0 & \cdots & 0 & g_{n,1} & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & \cdots & g_{1,1} & \cdots & 0 & 0 & \cdots & g_{n,1} \\
  \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
  g_{1,n} & 0 & \cdots & 0 & \cdots & g_{n,n} & 0 & \cdots & 0 \\
  0 & g_{1,n} & \cdots & 0 & \cdots & g_{n,n} & 0 & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & \cdots & g_{1,n} & \cdots & 0 & 0 & \cdots & g_{n,n}
\end{bmatrix}
\]

where

\[
w_{i+n(a-1),j+n(b-1)} = \begin{cases} 
g_{a,b} & \text{if } i = j \\
0 & \text{if } i \neq j
\end{cases}
\]

is the element in the \( i + n(a - 1) \)th row and the \( j + n(b - 1) \)th column of \( W \otimes I_{m \times m} \), for 
\( i, j = 1, \cdots, m \) and \( a, b = 1, \cdots, n \).

I define \( \Theta_5 \) so that
\[ \Theta_5 = Q' \left( W \otimes I_{m \times m} \right) = \]

\[
\begin{bmatrix}
\sum_{b=1}^{n} g_{1,b} f_{1}^b & 0 & \cdots & 0 & \cdots & \sum_{b=1}^{n} g_{n,b} f_{1}^b & 0 & \cdots & 0 \\
\sum_{b=1}^{n} g_{1,b} f_{2}^b & 0 & \cdots & 0 & \cdots & \sum_{b=1}^{n} g_{n,b} f_{2}^b & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\
\sum_{b=1}^{n} g_{1,b} f_{m}^b & 0 & \cdots & 0 & \cdots & \sum_{b=1}^{n} g_{n,b} f_{m}^b & 0 & \cdots & 0 \\
0 & \sum_{b=1}^{n} g_{1,b} f_{1}^b & \cdots & 0 & \cdots & 0 & \sum_{b=1}^{n} g_{n,b} f_{1}^b & \cdots & 0 \\
0 & \sum_{b=1}^{n} g_{1,b} f_{2}^b & \cdots & 0 & \cdots & 0 & \sum_{b=1}^{n} g_{n,b} f_{2}^b & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & \sum_{b=1}^{n} g_{1,b} f_{m}^b & \cdots & 0 & 0 & \cdots & \sum_{b=1}^{n} g_{n,b} f_{m}^b \\
0 & 0 & \cdots & \sum_{b=1}^{n} g_{1,b} f_{1}^b & \cdots & 0 & 0 & \cdots & \sum_{b=1}^{n} g_{n,b} f_{1}^b \\
0 & 0 & \cdots & \sum_{b=1}^{n} g_{1,b} f_{2}^b & \cdots & 0 & 0 & \cdots & \sum_{b=1}^{n} g_{n,b} f_{2}^b \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & \sum_{b=1}^{n} g_{1,b} f_{m}^b & \cdots & 0 & 0 & \cdots & \sum_{b=1}^{n} g_{n,b} f_{m}^b \\
\end{bmatrix}
\]

where

\[
\theta_{5,j+m(k-1),i+n(a-1)} = \begin{cases} 
\sum_{b=1}^{n} g_{a,b} f_{m}^b & \text{if } i = k \\
0 & \text{if } i \neq k 
\end{cases}
\]

is the element in the \( j + m(k-1) \)th row and the \( i + n(a-1) \)th column of \( \Theta_5 \), for \( i, j, k = 1, \ldots, m \) and \( a = 1, \ldots, n \).

Using the definition of \( S4 \), I can write

\[ S4 = \Theta_5 V \]
\[
S_4 = \begin{bmatrix}
\sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{1,1}^b & \sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{1,2}^b & \cdots & \sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{m,1}^b \\
\sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{2,1}^b & \sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{2,2}^b & \cdots & \sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{2,m}^b \\
\sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{m,1}^b & \sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{m,2}^b & \cdots & \sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{m,m}^b
\end{bmatrix}
\]

where

\[
s_{4,j+m(k-1),i} = \sum_{a=1}^{n} \sum_{b=1}^{n} g_{a,b} f_{i,k}^a f_{j}^b
\]
as required.

**Step 5**

From equation (A.7) \( S_5 = (R \otimes I_{m^2 \times m^2}) T \). I need to show that \( s_{5,j+m(k-1),i} = \sum_{a=1}^{n} g_{a,f_{i,j,k}}^a \)

\[
\Theta_6 = R \otimes I_{m^2 \times m^2} = \begin{bmatrix}
g_1 & 0 & \cdots & 0 & g_2 & 0 & \cdots & 0 & g_n & 0 & \cdots & 0 \\
0 & g_1 & \cdots & 0 & 0 & g_2 & \cdots & 0 & g_n & 0 & \cdots & 0 \\
\vdots & & & & & & & & & & & \\
0 & 0 & \cdots & g_1 & 0 & 0 & \cdots & 0 & 0 & \cdots & g_n
\end{bmatrix}
\]

where

\[
\theta_{6,i,j+n(a-1)} = \begin{cases} 
g_a & \text{if } i = j \\
0 & \text{if } i \neq j
\end{cases}
\]
is the element in the \( i \)th row and the \( j + n(a-1) \)th column in \( \Theta_6 \), for \( i,j = 1, \cdots, m \) and \( a = 1, \cdots, n \).

Using the definition of \( S_5 \), I can write

\[
S_5 = \Theta_6 T
\]
where

\[ s_{5j+m(k-1),i} = \sum_{a=1}^{n} g_{a} f_{i,j,k} \]

as required.

\[ \square \]

Appendix B. Sylvester Equations

To be included.

Appendix C. Model Solution

The solved matrices from section 4 are presented below.

\[ g_{x} = \begin{bmatrix} 0.538516074338190 & 0.128222800563108 & 0.160278500703885 \\ 0 & 0.800000000000000 & 1.000000000000000 \\ \end{bmatrix}, \]

\[ h_{x} = \begin{bmatrix} 0.960555718076461 & 0.081805764224287 & 0.102257205280358 \\ 0 & 0.800000000000000 & 1.000000000000000 \\ 0 & 0 & 0 \end{bmatrix}, \]
\[ g_{xx} = \begin{bmatrix}
0.050410880298460 & -0.056379980258910 & -0.070474975323637 \\
-0.056379980258910 & 0.048554933367482 & 0.060693666709352 \\
-0.070474975323637 & 0.060693666709352 & 0.075867083386690 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}, \]

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

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

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

\[ g_{xxx} = \begin{bmatrix}
0.000886224176982 & 0.018042424368051 & 0.022553030460064 \\
0.018042424368051 & -0.016047638262395 & -0.020059547827995 \\
0.022553030460064 & -0.020059547827994 & -0.025074434784993 \\
0.018042424368051 & -0.016047638262395 & -0.020059547827995 \\
-0.016047638262396 & 0.019412734848266 & 0.024265918560332 \\
-0.020059547827995 & 0.024265918560332 & 0.030332398200415 \\
-0.022553030460064 & -0.020059547827995 & -0.025074434784993 \\
-0.020059547827995 & 0.024265918560332 & 0.030332398200415 \\
-0.025074434784993 & 0.030332398200415 & 0.037915497750519 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}, \]
\[
\begin{bmatrix}
-0.020956383687171 & 0.029527273689885 & 0.036909092112356 \\
0.029527273689885 & -0.035680637163452 & -0.044600796454315 \\
0.036909092112356 & -0.044600796454315 & -0.055750995567894
\end{bmatrix}
\]

\[
\sigma x = \begin{bmatrix}
0.199558292329446 \times 10^{-4} \\
0.059796933577375 \times 10^{-4} \\
0.074746166971719 \times 10^{-4}
\end{bmatrix}
\]

\[
h_{xxx} = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix},
\]

\[
g_{\sigma x} = \begin{bmatrix}
0 & 0 & 0 \frac{0.208394896512764 \times 10^{-6}} \\
0 & 0 & -0.775000263651503 \times 10^{-6} \\
0 & 0 & -0.968750329564378 \times 10^{-6}
\end{bmatrix}
\]

\[
h_{\sigma x} = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix},
\]

33
\[ g_{\sigma\sigma} = \begin{bmatrix} -0.138593020922434 \times 10^{-6} \\ 0 \end{bmatrix}, \]
\[ h_{\sigma\sigma} = \begin{bmatrix} 0.127510245680320 \times 10^{-7} \\ 0 \\ 0 \end{bmatrix}. \]

References


