Fifth Order Perturbation Solution to DSGE Models

Oren Levintal*

Interdisciplinary Center Herzliya (IDC), School of Economics

oren.levintal@idc.ac.il

November 8, 2014

Abstract

This paper derives a fifth-order perturbation solution to DSGE models. Solutions of this order are required in models with rare disasters (Barro 2006), where lower order solutions fail to approximate accurately the true solution. The paper develops a new notation that is simpler than the standard notation of Schmitt-Grohé and Uribe (2004). Specifically, the perturbation parameter is treated as a state variable and the model is presented as a composite function. This yields a notational gain that rises exponentially with the perturbation order. Further notational gain is obtained by developing a compact matrix notation for high order chain rules. Finally, memory constraints are relaxed by exploiting symmetry and sparsity of the model derivatives.

Keywords: Perturbation, Fifth order, Rare disasters, High order multivariate chain rules, Compressed differentiation, Sylvester equation, DSGE, Symmetric tensors.

JEL classification: C63, C68, E0.

*I would like to thank Martin Uribe, Xavier Gabaix and Junior Maih for their helpful comments and suggestions. All errors are solely mine.

1 Introduction

Perturbation methods have become highly popular for solving DSGE models (Uhlig 1997, Gaspar and Judd 1997, Judd and Guu 1997, Judd 1998, Klein 2000, Sims 2001, Jin and Judd 2002). Recently, there has been a growing interest in high order perturbation solutions (Andreasen, Fernández-Villaverde and Rubio-Ramírez 2013). High order solutions are more accurate (Aruoba, Fernández-Villaverde and Rubio-Ramírez 2006), and they are essential for studying nonlinear issues such as welfare analysis (Kim and Kim 2003), volatility shocks (Fernández-Villaverde, Guerrón-Quintana, Rubio-Ramírez and Uribe 2011) or risk premia (Andreasen 2012).

The existing literature provides closed-form expressions for the second order solution (Schmitt-Grohé and Uribe 2004, Kim, Kim, Schaumburg and Sims 2008 and Gomme and Klein 2011) and the third order solution (Andreasen 2012, Ruge-Murcia 2012 and Binning 2013a). For standard business cycle models, these solutions are sufficiently accurate (Aruoba, Fernández-Villaverde and Rubio-Ramírez 2006). Yet there are models that require higher order solutions. Specifically, the paper shows that models with rare disasters, which are the subject of a growing body of literature,[1] cannot be solved accurately with second or even third order solutions. These models require fourth and fifth order solutions to get reasonable accuracy. Solutions of these orders are currently unavailable analytically.[2]

[1] See Rietz (1988), Barro (2006, 2009), Barro and Jin (2011), Gabaix (2011, 2012), Gourio (2012, 2013), Nakamura, Steinsson, Barro and Ursúa (2013), Wachter (2013), Farhi and Gabaix (2013).

[2] High order solutions can be obtained by pure symbolic algorithms, e.g. Aruoba, Fernández-Villaverde and Rubio-Ramírez (2006) and Swanson, Anderson and Levin (2005). These algorithms are difficult to optimize for memory and speed, because the structure of the problem is not transparent. Memory and speed are critical for high order solutions, both in forming the linear system and in solving it. These issues are discussed in detail in this paper. An alternative algorithm that

The present paper derives fourth and fifth order solutions to DSGE models, by extending the method of Schmitt-Grohé and Uribe (2004). The solutions are implemented by a MATLAB package available online.[3] The accuracy of the MATLAB package was tested on four models with closed form solutions: a neoclassical growth model with a closed form solution, Burnside (1998), Barro (2006), and an artificial model that has no economic meaning but enables to test accuracy on large models. For these models, the derivatives of the solution at the steady state can be solved analytically and compared to the numerical algorithm. In all cases the algorithm succeeded to replicate the true derivatives. All accuracy tests are available together with the MATLAB package.

Obtaining high order solutions is not trivial. The system of equations that defines the solution becomes extremely complicated and difficult to manage. This is seen most clearly in the third order solutions derived by Andreasen (2012) and Ruge-Murcia (2012). For instance, the third order system in Andreasen (2012) requires almost three pages of equations, with expressions that contain dozens of terms. One can only imagine how complicated the system would be for fourth and fifth order solutions.

This paper proposes a different notation, which is simpler than the existing literature. The new notation treats the perturbation parameter as an exogenous state variable, which is included in the vector of state variables x. As a result, the model is presented in the composite form $Ef(v(x)) = 0$, where v is a vector of the model

produces a general k-order solution is implemented by the Dynare^{++} software. However, Dynare^{++} is restricted to models with Gaussian shocks (Kamenik 2011, p. 9), so it is not applicable to models with rare-disaster shocks.

[3]The package is available at https://sites.google.com/site/orenlevintal.

variables and x is a vector that includes the state variables and the perturbation parameter. The solution is obtained by differentiating the model with respect to x only. This yields a notational gain that rises exponentially with the order of perturbation, which allows to derive high order solutions with less effort than previous studies.

Further notational gain is obtained by using high order multivariate chain rules to differentiate the model. The paper derives multivariate chain rules up to fifth order. Previous papers provide chain rules up to third order only (Magnus and Neudecker 1999 and Binning 2013a). These papers derive the chain rules by matrix notation, but their method is difficult to extend to fourth and fifth orders. By comparison, the present paper combines tensor and matrix notation interchangeably. Importantly, the proposed notation exploits permutations, which appear extensively in high order chain rules and bear significant notational gains. These tools enable to express the chain rules in a very compact matrix form, which is easy to use and code.[4]

Apart from notational complexity, high order solutions are likely to raise memory problems that do not exist in low order solutions, because the size of the problem grows exponentially with the perturbation order. This is particularly acute in the symbolic differentiation step, because the model needs to be differentiated five times with respect to all variables. To address this problem, the paper develops a compressed differentiation routine, which exploits sparsity and symmetry of high order derivatives. Specifically, the routine differentiates the model in a compressed man-

[4]MATALB codes that perform the chain rules are available at the author's homepage https://sites.google.com/site/orenlevintal.

ner, so that only the unique nonzero derivatives are differentiated and stored.[5] This feature makes it feasible to differentiate and solve medium-sized models, such as Christiano, Eichenbaum and Evans (2005), up to fifth order.

The importance of fifth order solutions is demonstrated on Barro (2006) and Gabaix (2012). These papers study an asset pricing model with rare disasters, following the work of Rietz (1988). Their model has proved useful in explaining many asset pricing puzzles (Gabaix 2012), and has become the benchmark model in this literature. The paper shows that rare-disaster models cannot be solved accurately by low order perturbation solutions. For instance, the second and third order solutions of the unlevered equity premium in Barro (2006) are 0.9 and 1.5 percent, respectively, whereas the true (closed-form) solution is 3.0 percent. Hence, the errors of low order solutions are economically significant, even at the third order. By comparison, the fifth order solution of the equity premium is 2.6 percent, which is much closer to the true solution. Similarly, the third order solution of the price-dividend ratio in Gabaix (2012) understates the true solution by 12%-19% over a small range around the steady state. The fifth order solution understates the true solution by 1%-5% only. These results suggest that the fourth and fifth order terms are economically important and should not be ignored in models with rare disasters.

Finally, the tools developed in this paper can be useful for other perturbation methods as well. As Kim, Kim, Schaumburg and Sims (2008) point out, "the use of perturbation methods ... is still in its early stages." Indeed, new methods have been proposed recently, e.g. perturbation of the impulse response function by Lan

[5] The MATLAB function `compderivs.m` performs the compressed differentiation and is available at the author's homepage.

and Meyer-Gohde (2013), or perturbation of markov-switching models by Foerster, Rubio-Ramírez, Waggoner and Zha (2013). These perturbation methods and others that are likely to emerge in the future, must handle notation complexity and memory issues similar to the ones addressed in this paper.

The paper proceeds as follows. Section 2 presents the new notation of the model. Section 3 derives the first and second order solutions and demonstrates the notational gain compared to Schmitt-Grohé and Uribe (2004). To extend the solution to higher orders, section 4 derives high order multivariate chain rules. Section 5 discusses the symmetry property and shows how to compress and uncompress derivative matrices. Section 6 describes the compressed differentiation routine, which exploits sparsity and symmetry of high order derivatives. The third, fourth and fifth order solutions are derived in section 7. Section 8 discusses the solution algorithm. Section 9 reports accuracy tests. Section 10 applies the algorithm on Barro (2006) and Gabaix (2012) and demonstrates the economic importance of the fourth and fifth order terms. Section 11 concludes. The appendix discusses some technical issues in more detail. A separate technical paper, Levintal (2014), elaborates on the implementation of the MATLAB package and some further technical issues.

2 The Model

The basic notation builds on Schmitt-Grohé and Uribe (2004). The only difference is that the perturbation parameter is treated as a "state" variable. Specifically, let σ_t denote an exogenous state variable that is constant across time:

$$\sigma_{t+1} = \sigma_t. \tag{1}$$

Define the vector of state variables by:

$$x_t = \left(x_t^1, x_t^2, \sigma_t\right), \tag{2}$$

where x_t^1 is a vector of n_x^1 pre-determined variables and x_t^2 is a vector of n_x^2 exogenous variables. The size of x_t is $n_x = n_x^1 + n_x^2 + 1$.

The model is defined by a set of n_f expectational conditions:

$$Ef\left(y_{t+1}, y_t, x_{t+1}, x_t\right) = 0, \tag{3}$$

where y_t is a vector of n_y control variables, and $f : \mathbb{R}^{2n_x+2n_y} \rightarrow \mathbb{R}^{n_x+n_y}$. Note that (1) is included in (3).

The solution has the form:

$$y_t = g\left(x_t\right), \tag{4}$$

$$x_{t+1} = h\left(x_t\right) + \sigma_t \eta \epsilon_{t+1}, \tag{5}$$

where y_t and x_t are presented as column vectors, $g : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$, $h : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$, η is a known matrix of dimensions $n_x \times n_\epsilon$, and ϵ is a $n_\epsilon \times 1$ vector of zero mean shocks. The cross moments of ϵ are denoted $M^2, M^3, \ldots$. For instance, M^2 is the variance-covariance matrix of ϵ, M^3 is a 3-dimensional tensor whose ijk element is $E\epsilon_i\epsilon_j\epsilon_k$, and so on. The model is deterministic at $\sigma_t = 0$ and stochastic for $\sigma_t > 0$.

Since σ_t is already included in x_t, it would be more efficient to represent the RHS of (5) as a function of x_t only. To do so, define a new $n_x \times n_x$ matrix ζ_t as follows:

$$\zeta_t = \left(0_{n_x \times (n_x - 1)}, \eta\epsilon_{t+1}\right). \tag{6}$$

ζ_t is a sparse matrix with all columns equal to zero except the last column, which is $\eta\epsilon_{t+1}$. With this matrix, (5) can be stated as a function of x_t only (presented as a column vector):

$$x_{t+1} = h(x_t) + \zeta_t x_t. \tag{7}$$

The model is completely defined by (3), (4) and (7). The solution is the functions $g(x_t)$ and $h(x_t)$ for which the system holds.

To reduce notation further, time subscripts are dropped and next period variables are denoted by $'$. In addition, let $v = (y', y, x', x)$ denote the vector of the model variables. Now, (3), (4) and (7) can be written as follows:

$$Ef(v) = 0, \tag{8}$$

$$v(x,\zeta) = \begin{pmatrix} g(h(x) + \zeta x) \\ g(x) \\ h(x) + \zeta x \\ x \end{pmatrix}, \tag{9}$$

where v is a column vector of size $n_v = 2(n_y + n_x)$. By this notation the model is presented as a composition of the functions $f(v)$ and $v(x,\zeta)$. Hence, f will be differentiated with respect to x only, by applying high order multivariate chain rules on the composite function $f(v(x,\zeta))$. This is the source of the notational gain.

The deterministic steady state is derived by setting $\sigma_t = 0$ and solving the steady state. Let the row vectors $\bar{x}^1$, $\bar{x}^2$ and $\bar{y}$ denote the steady state solution, and define $\bar{x} = (\bar{x}^1, \bar{x}^2, 0)$. Note that (6) implies that $\zeta\bar{x}^T = 0$. The steady state solution $\bar{x}, \bar{y}$ satisfies:

$$f(\bar{y}, \bar{y}, \bar{x}, \bar{x}) = 0.$$

A k'th order perturbation solution is the k'th order Taylor series of $g(x)$ and $h(x)$ centered at $\bar{x}$. The unknowns are the derivatives of $g(x)$ and $h(x)$ up to k order evaluated at $\bar{x}$. To solve these derivatives, (8) is differentiated k times with respect to x and evaluated at $\bar{x}$. Note that we do not need to differentiate the system with respect to the perturbation parameter, because it is already included in x. We get the following system:

$$0 = E\frac{\partial f^i}{\partial x_{j_1}}\bigg|_{x=\bar{x}} = E\frac{\partial^2 f^i}{\partial x_{j_1}\partial x_{j_2}}\bigg|_{x=\bar{x}} = \ldots = E\frac{\partial^k f^i}{\partial x_{j_1}\cdots\partial x_{j_k}}\bigg|_{x=\bar{x}} \tag{10}$$
$$\forall i \in \{1,\ldots,n_f\},\ j_1,\ldots,j_k \in \{1,\ldots,n_x\}.$$

where f^i denotes the i'th element of $f(v(x,\zeta))$, and x_{j_k} is the j_k'th element of x.

The solution is the unknown derivatives of the policy functions. As shown by Judd (1998) and Judd and Guu (1997), the solution is obtained recursively. The first equation defines the first derivatives as a solution of a quadratic system. The higher order conditions are linear in the unknown derivatives, given the solution of lower derivatives. Throughout, I assume that a first order solution is available (see Blanchard and Khan 1980, Uhlig 1997, Klein 2000, Sims 2001), and focus on the higher order solutions.[6]

3 First and Second Order Solutions

This section derives the first and second order perturbation solutions. The third, fourth and fifth order solutions are derived in section 7. High order solutions require to adopt notation that can handle multidimensional arrays. This paper uses tensor notation and matrix notation interchangeably, depending on the task at hand. Tensor

[6]The MATLAB package solves the first order solution with the function `gx_hx.m` written by Stephanie Schmitt-Grohé and Martín Uribe. I thank them for letting me use their code.

notation is more convenient for differentiation and for tensor permutations, which appear frequently in perturbations higher than second order. Matrix notation is required to form and manipulate the linear system.

Throughout the paper, high order derivatives will be denoted either by multi-dimensional tensors or by 2-dimensional matrices. The mapping from tensors to matrices (and vice versa) is implemented by the reshape operator.[7] In the mathematical literature the tensor-to-matrix mapping is called "tensor unfolding", see Ragnarsson and Van Loan (2012a). High order derivatives are unfolded in the following way. The first dimension of the tensor/matrix indexes the function that is differentiated. This dimension is identical for tensors and matrices. The difference is in the other dimensions. Specifically, the columns of the matrix correspond to the second to last dimensions of the tensor.

For example, the matrix g_{xx} contains the second derivatives of g with respect to x. The dimensions of this matrix are $n_y \times (n_x)^2$. It is obtained by reshaping (unfolding) the $n_y \times n_x \times n_x$ tensor of second derivatives into a matrix. Similarly, g_{xxx} denotes a matrix of dimensions $n_y \times (n_x)^3$ of the third derivatives of g, and so on. This matrix representation of high order derivatives is used also by Andreasen, Fernández-Villaverde and Rubio-Ramírez (2013). It is useful for presenting Taylor series in a compact way. For example, the second order Taylor series of $g(x)$ is given by:

$$g(x) = g(\bar{x}) + g_x (x - \bar{x}) + \frac{1}{2} g_{xx} (x - \bar{x})^{\otimes 2},$$

[7]The MATLAB function `reshape`.

where x and $\bar{x}$ are column vectors of size n_x, and $g(x)$ is a column vector of size n_y. The notation $^{\otimes k}$ denotes a "Kronecker power", e.g. $B^{\otimes 2} = B \otimes B$, see also Ragnarsson and Van Loan (2012b). The Kronecker power is performed before the inner product, hence $AB^{\otimes 2} \equiv A(B^{\otimes 2})$. For a different way to present Taylor series with matrix notation see Gomme and Klein (2011) and Benigno, Benigno and Nisticò (2013).

Finally, to distinguish between tensor and matrix notation, tensors are enclosed by square brackets. For example, $[g_{xx}]$ denotes the 3-dimensional tensor of the second derivatives of g with respect to x. The i, j, k element of this tensor is denoted by $[g_{xx}]^i_{jk}$. Matrices are denoted without brackets.

To get the first order solution, differentiate (8) with respect to x. Recall that x contains the perturbation variable, so this is the only differentiation that is required. The result is presented in tensor notation:

$$E\,[f_v]^i_\alpha\,[v_x]^\alpha_j = 0, \qquad \forall i = 1, \dots, n_f \quad j = 1, \dots, n_x. \tag{11}$$

$[f_v]^i_\alpha$ denotes the derivative of the i'th element of f with respect to the α'th element of v. Similarly, $[v_x]^\alpha_j$ denotes the derivative of the α'th element of v with respect to the j'th element of x. Greek letters denote summation indices, so $[f_v]^i_\alpha\,[v_x]^\alpha_j$ implies $\sum_\alpha [f_v]^i_\alpha\,[v_x]^\alpha_j$. This notation will be more useful for higher order solutions, which are derived later.

To get v_x, differentiate (9) with respect to x. Here, v_x is presented as a $n_v \times n_x$ matrix:

$$v_x = \begin{pmatrix} \frac{\partial g(h(x)+\zeta x)}{\partial x} \\ g_x \\ h_x + \zeta \\ I_{n_x} \end{pmatrix}. \tag{12}$$

The first element is a first order chain rule applied to the composite function $g\left(h\left(x\right) + \zeta x\right)$. It is presented in matrix form by:

$$\frac{\partial g\left(h\left(x\right) + \zeta x\right)}{\partial x} = g_x\left(h_x + \zeta\right). \tag{13}$$

As in Schmitt-Grohé and Uribe (2004), g_x and h_x denote the derivatives of the policy functions with respect to x, evaluated at the steady state $\bar{x}$.

The matrix v_x is stochastic, because it depends on the stochastic matrix ζ. It will be convenient for later stages to present it as a polynomial function of ζ. To do so, substitute (13) in (12) and write v_x as follows:

$$v_x = V_x^0 + V_x^1 \zeta, \tag{14}$$

where the coefficients V_x^0 and V_x^1 are defined by:

$$V_x^0 = \begin{pmatrix} g_x h_x \\ g_x \\ h_x \\ I_{n_x} \end{pmatrix}, \quad V_x^1 = \begin{pmatrix} g_x \\ 0_{n_y \times n_x} \\ I_{n_x} \\ 0_{n_x \times n_x} \end{pmatrix}. \tag{15}$$

To complete the first order system, substitute (14) in (11). This yields the condition for the first order solution, presented in matrix notation as:

$$f_v V_x^0 = 0. \tag{16}$$

It follows from (15) that this is a quadratic system in g_x and h_x.

To get the second order solution, differentiate (11) with respect to x:

$$E\left[f_{vv}\right]^i_{\alpha\beta}\left[v_x\right]^\beta_k\left[v_x\right]^\alpha_j + E\left[f_v\right]^i_\alpha\left[v_{xx}\right]^\alpha_{jk} = 0 \qquad \forall i = 1,\ldots,n_f \quad j,k = 1,\ldots,n_x. \tag{17}$$

The tensor notation $\left[f_{vv}\right]^i_{\alpha\beta}$ denotes the second derivative of the i'th element of f with respect to the α'th and β'th elements of v. In the mathematical literature the tensor products in (17) are called "tensor contractions", see Ragnarsson and Van Loan (2012a). These products can be reshaped (unfolded) into the following matrix notation:

$$E f_{vv} v_x^{\otimes 2} + E f_v v_{xx} = 0. \tag{18}$$

Here, f_{vv} is the tensor $[f_{vv}]$ reshaped into a matrix of dimensions $n_f \times (n_x)^2$, and v_{xx} is the tensor $[v_{xx}]$ reshaped into a matrix of dimensions $n_v \times (n_x)^2$. The unfolding of the tensor contractions in (17) into the matrices in (18) is explained in appendix A.1.

The term v_{xx} is obtained by differentiating (12) with respect to x:

$$v_{xx} = \begin{pmatrix} \frac{\partial^2 g(h(x)+\zeta x)}{\partial x^2} \\ g_{xx} \\ h_{xx} \\ 0_{n_x \times n_x^2} \end{pmatrix}. \tag{19}$$

Here, v_{xx} is a $n_v \times (n_x)^2$ matrix, and the dimensions of the matrices g_{xx} and h_{xx} are $n_y \times (n_x)^2$ and $n_x \times (n_x)^2$, respectively. $\frac{\partial^2 g(h(x)+\zeta x)}{\partial x^2}$ denotes a $n_y \times (n_x)^2$ matrix of the second derivatives of g with respect to x. To get these derivatives, apply a second order chain rule on the composite function $g\left(h\left(x\right) + \zeta x\right)$, which has a similar form to the left hand side of (18):

$$\frac{\partial^2 g\left(h\left(x\right) + \zeta x\right)}{\partial x^2} = g_{xx}\left(h_x + \zeta\right)^{\otimes 2} + g_x h_{xx}.$$

Substituting in (19) yields a quadratic function of ζ:

$$v_{xx} = V_{xx}^0 + V_{xx}^1 \left(h_x \otimes \zeta + \zeta \otimes h_x\right) + V_{xx}^1 \zeta^{\otimes 2}, \tag{20}$$

where the coefficients are:

$$V_{xx}^0 = \begin{pmatrix} g_{xx} h_x^{\otimes 2} + g_x h_{xx} \\ g_{xx} \\ h_{xx} \\ 0 \end{pmatrix}, \quad V_{xx}^1 = \begin{pmatrix} g_{xx} \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{21}$$

Note that the first term in V_{xx}^0 is the second order chain rule applied to $g\left(h\left(x\right)\right)$, namely $\frac{\partial^2 g(h(x))}{\partial x^2}$. Similarly, the first term in V_x^0, given in (15), was the first order chain rule $\frac{\partial g(h(x))}{\partial x}$. This pattern will show up again in higher order solutions.

To calculate (18), we need the expected values of $v_x^{\otimes 2}$ and v_{xx}, which follow from (14) and (20):

$$Ev_x^{\otimes 2} = \left(V_x^0\right)^{\otimes 2} + \left(V_x^1\right)^{\otimes 2} E\zeta^{\otimes 2}, \tag{22}$$

$$Ev_{xx} = V_{xx}^0 + V_{xx}^1 E\zeta^{\otimes 2}. \tag{23}$$

The term $E\zeta^{\otimes 2}$ can be derived from (6), as follows:

$$E\zeta^{\otimes 2} = \left(0_{(n_x)^2 \times \left((n_x)^2-1\right)}, E\left(\eta\epsilon\right)^{\otimes 2}\right) \tag{24}$$
$$= \left(0_{(n_x)^2 \times \left((n_x)^2-1\right)}, \eta^{\otimes 2}\text{vec}\left(M^2\right)\right),$$

where vec (M^2) is a vectorization of the variance-covariance matrix M^2.

Substituting (22)-(23) in (18) and using (21) yield the second order system in g_{xx} and h_{xx}:

$$A + f_{y'}g_{xx}B + f_y g_{xx} + \left(f_{x'} + f_{y'}g_x\right)h_{xx} = 0, \tag{25}$$
$$A \equiv f_{vv}Ev_x^{\otimes 2},$$
$$B \equiv \left(h_x^{\otimes 2} + E\zeta^{\otimes 2}\right).$$

$f_{y'}, f_y, f_{x'}, f_x$ denote derivatives of f with respect to y', y, x', x, respectively, hence $f_v = (f_{y'}, f_y, f_{x'}, f_x)$. Higher order systems will have a similar structure, only with different matrices A and B. The solution algorithm of this system is discussed in section 8.

The notational gain of the proposed method can be illustrated by comparing (25) with the equivalent system in Schmitt-Grohé and Uribe (2004, pp. 762-763). The inclusion of the perturbation parameter in x eliminates expressions such as $g_{\sigma\sigma}$, which are already included in g_{xx}. Moreover, since f is presented as a composite function $f\left(v\left(x,\zeta\right)\right)$, all the second derivatives of f are included in f_{vv}. This yields a significant notational gain, which rises exponentially with the order of perturbation. For instance, the derivative f_{vv} replaces 16 derivatives of f with respect to y', y, x', x that appear in the notation of Schmitt-Grohé and Uribe (2004). In a fifth order system, f_{vvvvv} replaces 1,024 derivatives of f with respect to y', y, x', x.

The next sections extend this method to higher order solutions. This requires to derive high order chain rules, which were used in the second order system in (18), (19) and (21). Higher order systems will use higher order chain rules at equivalent points.

4 High Order Multivariate Chain Rules

This section derives multivariate chain rules up to fifth order. The rules are derived by tensor notation and then transformed (unfolded) into matrix notation, which is much more compact and useful. An alternative way is to derive high order chain rules directly in matrix notation, as shown by Magnus and Neudecker (1999) for a second order rule and Binning (2013a) for a third order rule. However, this method becomes extremely tedious for higher orders. The tensor notation used here is simpler because it enables to exploit permutations very easily, yielding significant notational gains. Moreover, the symmetric structure of the problem is preserved in a transparent way, which will be used later to compress and uncompress these expressions.

The chain rules are implemented by the MATLAB functions chain2.m, chain3.m, chain4.m and chain5.m, which are available online together with all the other codes of this paper. These codes were tested by comparison with MATLAB symbolic differentiation. The test file is also available online.

For a given composite function $f(v(x))$, a k'th order multivariate chain rule generates a $k+1$ dimensional tensor, denoted $\left[\frac{\partial^k f(v(x))}{\partial x^k}\right]$. The first dimension of this tensor indexes an element of f, and the other dimensions index elements of x, with respect to which the function is differentiated. For example, $\left[\frac{\partial^3 f(v(x))}{\partial x^3}\right]^i_{jkl}$ denotes $\frac{\partial^3 f_i(v(x))}{\partial x_j \partial x_k \partial x_l}$. A single element of the tensor can be calculated by the Faa di Bruno formula for the multivariate case (Constantine and Savits 1996). However, here we are interested in calculating the entire tensor. Hence, the formulas are derived by successive differentiations.

In what follows, $[f_v], [f_{vv}], \ldots$ denote tensors of the first, second and higher derivatives of f with respect to v. Similarly, $[v_x], [v_{xx}], \ldots$ denote the derivative tensors of v with respect to x. A particular element of these tensors is denoted by its indices. For instance, $[f_{vvv}]^i_{jkl}$ denotes the third derivative of the i'th element of f with respect to the j'th, k'th and l'th elements of v. As before, tensors are enclosed by square brackets whereas matrices are denoted without brackets.

The second order chain rule was introduced in the previous section. It is presented again for convenience:

$$\left[\frac{\partial^2 f(v(x))}{\partial x^2}\right]^i_{jk} = [f_{vv}]^i_{\alpha\beta} [v_x]^\beta_k [v_x]^\alpha_j + [f_v]^i_\alpha [v_{xx}]^\alpha_{jk} . \tag{26}$$

The notation of multivariate differentiation follows Schmitt-Grohé and Uribe (2004). This rule appears also in Judd (1998, p. 490). To get the third order chain rule, differentiate (26) with respect to the l'th element of x. This gives a new tensor denoted $\left[\frac{\partial^3 f(v(x))}{\partial x^3}\right]$ whose $ijkl$ element is:

$$\begin{aligned}\left[\frac{\partial^3 f\left(v\left(x\right)\right)}{\partial x^3}\right]^i_{jkl} &= \left[f_{vvv}\right]^i_{\alpha\beta\gamma}\left[v_x\right]^\gamma_l\left[v_x\right]^\beta_k\left[v_x\right]^\alpha_j + \left[f_{vv}\right]^i_{\alpha\beta}\left[v_{xx}\right]^\beta_{kl}\left[v_x\right]^\alpha_j \\ &+ \left[f_{vv}\right]^i_{\alpha\beta}\left[v_x\right]^\beta_k\left[v_{xx}\right]^\alpha_{jl} + \left[f_{vv}\right]^i_{\alpha\beta}\left[v_x\right]^\beta_l\left[v_{xx}\right]^\alpha_{jk} + \left[f_v\right]^i_\alpha\left[v_{xxx}\right]^\alpha_{jkl}.\end{aligned} \tag{27}$$

This expression contains five terms, where each term is an element of a tensor with dimensions $n_f \times n_x \times n_x \times n_x$. However, the three middle tensors are three different permutations of the same tensor, because f_{vv} is symmetric in the indices α and β.[8]

Hence, the third order multivariate chain rule can be stated more compactly as:

$$\begin{aligned}\left[\frac{\partial^3 f\left(v\left(x\right)\right)}{\partial x^3}\right]^i_{jkl} &= \left[f_{vvv}\right]^i_{\alpha\beta\gamma}\left[v_x\right]^\gamma_l\left[v_x\right]^\beta_k\left[v_x\right]^\alpha_j + \sum_{qrst\in\Omega_1}\left(\left[f_{vv}\right]^q_{\alpha\beta}\left[v_{xx}\right]^\beta_{rs}\left[v_x\right]^\alpha_t\right) \\ &+ \left[f_v\right]^i_\alpha\left[v_{xxx}\right]^\alpha_{jkl},\end{aligned} \tag{28}$$

where Ω_1 is a set of three permutations, defined by:

$$\Omega_1 = \left\{iklj, ijlk, ijkl\right\}. \tag{29}$$

Proceeding similarly, the fourth order multivariate chain rule is:

[8]Due to this symmetry, the third term can be restated as $\left[f_{vv}\right]^i_{\beta\alpha}\left[v_{xx}\right]^\alpha_{jl}\left[v_x\right]^\beta_k$, and the fourth term as $\left[f_{vv}\right]^i_{\beta\alpha}\left[v_{xx}\right]^\alpha_{jk}\left[v_x\right]^\beta_l$. These are two permutations of the second term.

$$\left[\frac{\partial^4 f(v(x))}{\partial x^4}\right]^i_{jklm} = [f_{vvvv}]^i_{\alpha\beta\gamma\delta}[v_x]^\delta_m[v_x]^\gamma_l[v_x]^\beta_k[v_x]^\alpha_j \tag{30}$$

$$+\sum_{qrstu\in\Omega_2}\left([f_{vvv}]^q_{\alpha\beta\gamma}[v_{xx}]^\gamma_{rs}[v_x]^\beta_t[v_x]^\alpha_u\right)+\sum_{qrstu\in\Omega_3}\left([f_{vv}]^q_{\alpha\beta}[v_{xxx}]^\beta_{rst}[v_x]^\alpha_u\right)$$

$$+\sum_{qrstu\in\Omega_4}\left([f_{vv}]^q_{\alpha\beta}[v_{xx}]^\beta_{rs}[v_{xx}]^\alpha_{tu}\right)+[f_v]^i_\alpha[v_{xxxx}]^\alpha_{jklm}$$

where

$$\Omega_2 = \{ilmkj, ikmlj, ijmlk, iklmj, ijlmk, ijkml\}$$

$$\Omega_3 = \{iklmj, ijlmk, ijkml, ijklm\}$$

$$\Omega_4 = \{ikljm, ijlkm, ijklm\}$$

The permutation sets $\Omega_2, \Omega_3, \Omega_4$ are obtained by counting the indices of permuted tensors. Take for example the middle term in (28). Differentiating $[f_{vv}]^q_{\alpha\beta}[v_{xx}]^\beta_{rs}[v_x]^\alpha_t$ with respect to x_m (the m'th element of x) requires to apply the derivative of a product. One of the expressions included in this derivative is:

$$[f_{vv}]^q_{\alpha\beta}[v_{xx}]^\beta_{rs}[v_{xx}]^\alpha_{tm}.$$

The permutation set Ω_1 implies that $qrst \in \{iklj, ijlk, ijkl\}$. Summing these permutations together gives:

$$\sum_{qrstu\in\{ikljm,ijlkm,ijklm\}}\left([f_{vv}]^q_{\alpha\beta}\,[v_{xx}]^\beta_{rs}\,[v_{xx}]^\alpha_{tu}\right).$$

These are the permutations in Ω_4.

The fifth order multivariate chain rule is:

$$\begin{aligned}\left[\frac{\partial^5 f(v(x))}{\partial x^5}\right]^i_{jklmn} &= [f_{vvvvv}]^i_{\alpha\beta\gamma\delta\epsilon}\,[v_x]^\epsilon_n\,[v_x]^\delta_m\,[v_x]^\gamma_l\,[v_x]^\beta_k\,[v_x]^\alpha_j \qquad (31)\\ &+\sum_{qrstuv\in\Omega_5}\left([f_{vvvv}]^q_{\alpha\beta\gamma\delta}\,[v_{xx}]^\delta_{rs}\,[v_x]^\gamma_t\,[v_x]^\beta_u\,[v_x]^\alpha_v\right)+\sum_{qrstuv\in\Omega_6}\left([f_{vvv}]^q_{\alpha\beta\gamma}\,[v_{xxx}]^\gamma_{rst}\,[v_x]^\beta_u\,[v_x]^\alpha_v\right)\\ &+\sum_{qrstuv\in\Omega_7}\left([f_{vvv}]^q_{\alpha\beta\gamma}\,[v_{xx}]^\gamma_{rs}\,[v_{xx}]^\beta_{tv}\,[v_x]^\alpha_n\right)+\sum_{qrstuv\in\Omega_8}\left([f_{vv}]^q_{\alpha\beta}\,[v_{xxxx}]^\beta_{rstu}\,[v_x]^\alpha_v\right)\\ &+\sum_{qrstuv\in\Omega_9}\left([f_{vv}]^q_{\alpha\beta}\,[v_{xxx}]^\beta_{rst}\,[v_{xx}]^\alpha_{uv}\right)+[f_v]^i_\alpha\,[v_{xxxxx}]^\alpha_{jklmn}\end{aligned}$$

where:

$$\Omega_5=\{imnlkj, ilnmkj, iknmlj, ijnmlk, ilmnkj, ikmnlj, ijmnlk, iklnmj, ijlnmk, ijknml\}$$

$$\Omega_6=\{ilmnkj, ikmnlj, ijmnlk, iklnmj, ijlnmk, ijknml, iklmnj, ijlmnk, ijkmnl, ijklnm\}$$

$$\begin{aligned}\Omega_7=\{&ilmknj, ikmlnj, ijmlnk, iklmnj, ijlmnk, ijkmnl, ilmjnk, ikmjnl, ijmknl, ikljnm,\\ &ijlknm, ijklnm, ikljmn, ijlkmn, ijklmn\}\end{aligned}$$

$$\Omega_8=\{iklmnj, ijlmnk, ijkmnl, ijklnm, ijklmn\}$$

$$\Omega_9=\{iklmjn, ijlmkn, ijkmln, ijklmn, iklnjm, ijlnkm, ijknlm, ijmnkl, ikmnjl, ilmnjk\}$$

Matrix notation provides an easier way to present high order chain rules. To this end, we need to unfold the tensor products ("tensor contractions"), namely, expressions such as $[f_{vv}]^q_{\alpha\beta}[v_{xx}]^\beta_{rs}[v_x]^\alpha_t$. Then, the permutations Ω should be implemented by matrices. These issues are explained in Appendix A.2. The final result is the following third, fourth and fifth order chain rules in matrix notation:

$$\frac{\partial^3 f(v(x))}{\partial x^3} = f_{vvv}v_x^{\otimes 3} + f_{vv}(v_x \otimes v_{xx})\Omega_1 + f_v v_{xxx} \tag{32}$$

$$\frac{\partial^4 f(v(x))}{\partial x^4} = f_{vvvv}v_x^{\otimes 4} + f_{vvv}\left(v_x^{\otimes 2} \otimes v_{xx}\right)\Omega_2 + f_{vv}(v_x \otimes v_{xxx})\Omega_3 + f_{vv}\left(v_{xx}^{\otimes 2}\right)\Omega_4 + f_v v_{xxxx} \tag{33}$$

$$\frac{\partial^5 f(v(x))}{\partial x^5} = f_{vvvvv}v_x^{\otimes 5} + f_{vvvv}\left(v_x^{\otimes 3} \otimes v_{xx}\right)\Omega_5 + f_{vvv}\left(v_x^{\otimes 2} \otimes v_{xxx}\right)\Omega_6 + f_{vvv}\left(v_x \otimes v_{xx}^{\otimes 2}\right)\Omega_7 + f_{vv}(v_x \otimes v_{xxxx})\Omega_8 + f_{vv}(v_{xx} \otimes v_{xxx})\Omega_9 + f_v v_{xxxxx} \tag{34}$$

In this notation, $\Omega_1, \ldots, \Omega_9$ denote matrices that perform the sum of permutations specified in the tensor notation. These are coefficient matrices that are functions of n_x only. They can be calculated in advance, and then used whenever high order chain rules are applied. Note that these matrices are sparse so their memory consumption is low. Further details are provided in Appendix A.2. The MATLAB function `create_OMEGA` available online calculates the Ω matrices.

5 Symmetry

Symmetry is a strong property of high order solutions. It follows from the symmetry of mixed derivatives. For instance, the tensor $[v_{xxx}]$ is of dimensions $n_v \times n_x \times n_x \times n_x$. Its $ijkl$ element is the third derivative of the i'th row of v with respect to the j'th, k'th and l'th elements of x. Hence, the second to last dimensions of $[v_{xxx}]$ are symmetric in the sense that permuting these dimensions does not change the tensor.[9] The $n_y \times (n_x)^3$ matrix v_{xxx} is a reshaped form of the corresponding tensor. The symmetry of the tensor is reflected by the repetitive columns of v_{xxx}.

Since the matrix v_{xxx} has repetitive columns, it can be compressed into its unique columns. For a general k order derivative matrix v_{x^k}, the number of unique columns is $\binom{n_x+k-1}{k}$, see Ballard, Kolda and Plantenga (2011) for a proof. Note that the number of unique derivatives is smaller by a factor of approximately $k!$ compared to the full number of derivatives. This factor is very large at high orders. Hence, compressing derivatives by extracting their unique elements entails high memory gains.

To compress v_{x^k}, let U_k be a $(n_x)^k \times \binom{n_x+k-1}{k}$ matrix that extracts the unique columns from the derivative matrix v_{x^k} by the inner products $v_{x^k}U_k$. For example, $v_{xxx}U_3$ is a matrix of dimensions $n_v \times \frac{n_x(n_x+1)(n_x+2)}{6}$ whose columns are the unique third derivatives of v. In addition, define the matrix W_k with dimensions $\binom{n_x+k-1}{k} \times (n_x)^k$ that performs the reverse process. Namely, W_k uncompresses the compressed derivative matrix. It follows that $v_{x^k}U_kW_k = v_{x^k}$. Note that this holds only for derivative matrices such as v_{x^k}, where the columns are repetitive in a very specific way, hence $U_kW_k \neq I$.[10]

The high order multivariate chain rules used in this paper preserve the symmetry property. Consequently, it is very easy to compress these rules into unique terms, and vice versa. For instance, the third order chain rule (32) is a matrix of dimensions $n_f \times (n_x)^3$. To extract the unique derivatives of f, postmultiply by U_3:

[9] For a formal definition of a symmetric tensor see Ballard, Kolda and Plantenga (2011).

[10] The MATLAB function `create_UW.m` written by the author calculates the compress and uncompress matrices U and W, given n_x and k. It is available at the author's homepage as part of the perturbation package.

$$f_{vvv}v_x^{\otimes 3}U_3 + f_{vv}(v_x \otimes v_{xx})\Omega_1 U_3 + f_v v_{xxx}U_3.$$

This is the unique third derivatives of $f(v(x))$ with respect to x. Note that $v_{xxx}U_3$ is the unique third derivatives of $v(x)$ with respect to x. Moreover, each derivative matrix can be expressed in terms of its compressed form. For instance, substituting $v_{xx} = v_{xx}U_2W_2$ gives an expression in terms of $v_{xx}U_2$, which is the unique second derivatives of $v(x)$. This type of manipulations can be done easily due to the symmetric structure of the chain rule notation.

6 Compressed differentiation

High order solutions require to differentiate $f(v)$ with respect to v several times. This creates matrices that can be very large, because $n_v = 2(n_y + n_x)$. While the differentiation is performed automatically by symbolic software, the memory requirement may be too large. For example, the New Keynsian model of Christiano, Eichenbaum and Evans (2005) has 11 state variables, 11 control variables and 22 equations.[11] Differentiating the model symbolically with respect to all state and control variables for current and next period gives an incredible number of more than 3.6 billion fifth order derivatives (the matrix f_{vvvvv}). Symbolic differentiation of this size requires an amount of memory that is well beyond the capacity of a standard desktop computer.

This section develops a compressed differentiation routine that allows to differen-

[11] I use the formulation of Schmitt-Grohé and Uribe (2004b) for this model.

tiate large models, by exploiting sparsity and symmetry of high order derivatives.[12] Specifically, many derivatives of f with respect to v are likely to be zero. For example, the budget constraint of the neoclassical growth model is $A_t k_t^{\alpha} + k_t (1-\delta) - c_t - k_{t+1} = 0$. The derivatives of this equation with respect to c_t and k_{t+1} are all zero, except for the first derivatives. Furthermore, mixed derivatives are symmetric. Hence, when we differentiate this equation, we need to store only the nonzero unique derivatives. The nonzero unique derivatives are differentiated to get higher order derivatives, and again only the nonzero unique elements are stored.

It is shown in Levintal (2014) that this procedure enables to differentiate Christiano, Eichenbaum and Evans (2005) up to fifth order in about 85 seconds. Once the model has been differentiated symbolically, the numerical fifth order solution is obtained in about 30 seconds on a standard desktop computer.

The compressed differentiation is performed recursively. Each step produces higher order nonzero unique derivatives and matrices that transform these unique derivatives into the full matrix of derivatives. These matrices are functions of the previous step. Importantly, the compressed differentiation is performed separately for each element of f, denoted f^i. This allows us to fully exploit sparsity, because the sparse structure of each element of f is different.

To present the algorithm, suppose we have already produced the third order derivatives f_{vvv} and we wish to proceed to the fourth derivatives. Consider row i of the matrix of third derivatives f_{vvv} and denote it by f^i_{vvv}. This is a row vector of length $(n_v)^3$ that contains all the third derivatives of the i'th row of f with respect

[12]The compressed differentiation routine is performed by the MATLAB function `compderivs.m` available at the author's homepage.

to v. Suppose that N_3^i is a matrix that extracts the nonzero entries from f_{vvv}^i by the inner product $f_{vvv}^i N_3^i$, and M_3^i is a matrix that inserts back the zero entries.[13] We can express f_{vvv}^i as follows:

$$f_{vvv}^i = f_{vvv}^i N_3^i M_3^i.$$

The row vector $f_{vvv}^i N_3^i$ contains all the nonzero third derivatives of f^i. Due to symmetry of mixed derivatives, we can further compress $f_{vvv}^i N_3^i$ by extracting the unique nonzero derivatives. Let U_3^i denote the matrix that extracts the unique nonzero derivatives from $f_{vvv}^i N_3^i$ and let W_3^i denote the reverse matrix. It follows that:

$$f_{vvv}^i = f_{vvv}^i N_3^i U_3^i W_3^i M_3^i.$$

The i superscript is required because each row of f has a different sparse structure, and so different compression matrices.

Transposing this equation yields:

$$\text{vec}\left(f_{vvv}^i\right) = \left(W_3^i M_3^i\right)^T \text{vec}\left(f_{vvv}^i N_3^i U_3^i\right). \tag{35}$$

This result shows how to obtain the full vector of third derivatives from the compressed vector $f_{vvv}^i N_3^i U_3^i$ that contains only the unique nonzero derivatives. Conse-

[13]It can be shown that M_3^i is the transpose of N_3^i.

quently, storage requirements are much smaller, because we store only the unique nonzero derivatives and the compression matrices, which are sparse.

To differentiate (35) with respect to v, let $J\left(f^i_{vvv}\right)$ denote the Jacobian matrix of the column vector $\operatorname{vec}\left(f^i_{vvv}\right)$ and let $J\left(f^i_{vvv}N^i_3U^i_3\right)$ denote the Jacobian matrix of the column vector $\operatorname{vec}\left(f^i_{vvv}N^i_3U^i_3\right)$. It follows that:

$$J\left(f^i_{vvv}\right) = \left(W^i_3 M^i_3\right)^T J\left(f^i_{vvv}N^i_3U^i_3\right). \tag{36}$$

The left-hand-side is the matrix of fourth derivatives f^i_{vvvv}, reshaped into dimensions $(n_v)^3 \times n_v$. These are also the dimensions of the right-hand-side. Hence, we can express (36) in a vectorized form:

$$\operatorname{vec}\left(f^i_{vvvv}\right) = \left(I_{n_v} \otimes \left(W^i_3 M^i_3\right)^T\right) \operatorname{vec}\left(J\left(f^i_{vvv}N^i_3U^i_3\right)\right). \tag{37}$$

Here, we used the property $\operatorname{vec}(ABC) = \left(C^T \otimes A\right)\operatorname{vec}(B)$. Transposing and post-multiplying by $N^i_4U^i_4$ yields a row vector of the unique nonzero fourth derivatives:

$$f^i_{vvvv}N^i_4U^i_4 = \left(\operatorname{vec}\left(J\left(f^i_{vvv}N^i_3U^i_3\right)\right)\right)^T \left(I_{n_v} \otimes \left(W^i_3 M^i_3\right)\right) N^i_4U^i_4, \tag{38}$$

where f^i_{vvvv} is a row vector of size $(n_v)^4$ that contains all fourth derivatives of f^i.

Equation (38) is the main recursive formula. It provides the vector of unique nonzero fourth derivatives $(f^i_{vvvv} N^i_4 U^i_4)$ as a function of the Jacobian of the unique nonzero third derivatives $(f^i_{vvv} N^i_3 U^i_3)$. This formula enables to speed up symbolic differentiation by differentiating only the unique nonzero third derivatives, and using the matrices $N^i_3, M^i_3, U^i_3, W^i_3, N^i_4, U^i_4$ to construct the unique nonzero fourth derivatives. Then, we can obtain the full matrix of fourth derivatives by a fourth-order version of (35).

The recursive procedure is implemented in the following order, assuming we already have the unique nonzero third order derivatives and the matrices $N^i_3, M^i_3, U^i_3, W^i_3$ from the previous step. First, differentiate the unique nonzero third order derivatives to get the Jacobian matrix $J\,(f^i_{vvv} N^i_3 U^i_3)$. Second, use (36) to obtain the matrix of fourth derivatives and detect the nonzero elements of this matrix.[14] Third, construct N^i_4 and M^i_4 from the information on the nonzero fourth derivatives. Five, construct U^i_4 and W^i_4 to compress and uncompress the nonzero fourth derivatives. Finally, obtain the unique nonzero fourth derivatives through (38). Having the unique nonzero fourth derivatives and the matrices $N^i_4, M^i_4, U^i_4, W^i_4$ we can proceed to the fifth order by the same recursion.

7 Third, Fourth and Fifth Order Solutions

The high order multivariate chain rules derived previously are used in this section to derive the third, fourth and fifth order solutions. To do so, it is useful to introduce

[14]To do so, we do not need to retrieve the entire matrix of fourth derivatives. It is sufficient to get only its sparse structure. This is done by replacing the Jacobian $J\left(f^i_{vvv} N^i_3 U^i_3\right)$ in (36) with a sparse matrix of the same dimensions, where nonzero entries of the Jacobian are replaced with ones, and all other entries are zeros.

a new notation that simplifies sums of permutations of Kronecker products. For example, $A^{\otimes 2} \otimes B + A \otimes B \otimes A + B \otimes A^{\otimes 2}$ is a sum of all the combinations of Kronecker products that can be created by the matrices A, A, B. To simplify this expression I use the notation $P\left(A^{\otimes 2} \otimes B\right)$, where P is an operator that permutes $A^{\otimes 2} \otimes B$ in all possible combinations of Kronecker products, and sums all the permutations together.

Other expressions that will be necessary are the derivatives of v with respect to x. These derivatives are obtained by differentiating (9). The second derivative is given in (19) in matrix form. The third, fourth and fifth derivatives have similar matrix forms:

$$v_{xxx} = \begin{pmatrix} \frac{\partial^3 g(h(x)+\zeta x)}{\partial x^3} \\ g_{xxx} \\ h_{xxx} \\ 0_{n_x \times n_x^3} \end{pmatrix} \quad v_{xxxx} = \begin{pmatrix} \frac{\partial^4 g(h(x)+\zeta x)}{\partial x^4} \\ g_{xxxx} \\ h_{xxxx} \\ 0_{n_x \times n_x^4} \end{pmatrix} \quad v_{xxxxx} = \begin{pmatrix} \frac{\partial^5 g(h(x)+\zeta x)}{\partial x^5} \\ g_{xxxxx} \\ h_{xxxxx} \\ 0_{n_x \times n_x^5} \end{pmatrix} \tag{39}$$

Each of these matrices has n_v rows. The number of columns is $(n_x)^k$ for the k order matrix. Note that the first term in each matrix applies a k order multivariate chain rule on the composite function $g\left(h\left(x\right) + \zeta x\right)$.

7.1 Third order solution

A third order system is defined by:

$$f_{vvv}Ev_x^{\otimes 3} + f_{vv}E\left(v_x \otimes v_{xx}\right)\Omega_1 + f_v E v_{xxx} = 0, \tag{40}$$

where (32) provides the matrix notation of the third order chain rule. The stochastic matrices are v_x, v_{xx}, v_{xxx}. To derive $Ev_x^{\otimes 3}$ and $Ev_x \otimes v_{xx}$, use (14) and (20):

$$Ev_x^{\otimes 3} = \left(V_x^0\right)^{\otimes 3} + EP\left(V_x^0 \otimes \left(V_x^1\zeta\right)^{\otimes 2}\right) + E\left(V_x^1\zeta\right)^{\otimes 3}, \tag{41}$$

$$\begin{aligned} Ev_x \otimes v_{xx} = {} & V_x^0 \otimes V_{xx}^0 + E\left(V_x^1\zeta\right) \otimes \left(V_{xx}^1 P\left(\zeta \otimes h_x\right)\right) + EV_x^0 \otimes \left(V_{xx}^1 \zeta^{\otimes 2}\right) \\ & + E\left(V_x^1\zeta\right) \otimes \left(V_{xx}^1 \zeta^{\otimes 2}\right). \end{aligned} \tag{42}$$

Expressions that are linear in ζ drop because $E\zeta = 0$. The remaining stochastic components are quadratic and cubic functions of ζ. Appendix A.3 shows how to calculate the expected value of these terms.

To calculate Ev_{xxx}, apply a third order chain rule on the composite function $g\left(h\left(x\right) + \zeta x\right)$ in order to get the first element of v_{xxx}, which is defined in (39):

$$\begin{aligned} \frac{\partial^3 g\left(h\left(x\right) + \zeta x\right)}{\partial x^3} = {} & g_{xxx}\left(h_x + \zeta\right)^{\otimes 3} + g_{xx}\left(\left(h_x + \zeta\right) \otimes h_{xx}\right)\Omega_1 + g_x h_{xxx} \\ = {} & g_{xxx}\left(h_x^{\otimes 3} + P\left(h_x^{\otimes 2} \otimes \zeta\right) + P\left(h_x \otimes \zeta^{\otimes 2}\right) + \zeta^{\otimes 3}\right) \\ & + g_{xx}\left(h_x \otimes h_{xx} + \zeta \otimes h_{xx}\right)\Omega_1 + g_x h_{xxx}. \end{aligned} \tag{43}$$

Substituting in (39) yields a third order polynomial in ζ:

$$v_{xxx} = V^0_{xxx} + V^1_{xxx} P\left(h_x^{\otimes 2} \otimes \zeta\right) + V^1_{xxx} P\left(h_x \otimes \zeta^{\otimes 2}\right) + V^1_{xxx}\left(\zeta^{\otimes 3}\right) + V^1_{xx}\left(\zeta \otimes h_{xx}\right)\Omega_1, \tag{44}$$

where the new coefficients are:

$$V^0_{xxx} = \begin{pmatrix} \frac{\partial^3 g(h(x))}{\partial x^3} \\ g_{xxx} \\ h_{xxx} \\ 0 \end{pmatrix}, \qquad V^1_{xxx} = \begin{pmatrix} g_{xxx} \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{45}$$

Taking expectation yields:

$$Ev_{xxx} = V^0_{xxx} + V^1_{xxx} EP\left(h_x \otimes \zeta^{\otimes 2}\right) + V^1_{xxx} E\left(\zeta^{\otimes 3}\right). \tag{46}$$

Appendix A.3 shows how to calculate the expected values of the quadratic and cubic functions of ζ.

Having (41), (42) and (46), we can substitute in (40) and form a linear system in g_{xxx}, h_{xxx}, with the same structure as the second order system (25):

$$A + f_{y'} g_{xxx} B + f_y g_{xxx} + (f_{x'} + f_{y'} g_x) h_{xxx} = 0 \tag{47}$$

$$A \equiv f_{vvv} E v_x^{\otimes 3} + f_{vv} E (v_x \otimes v_{xx}) \Omega_1 + f_{y'} g_{xx} (h_x \otimes h_{xx}) \Omega_1$$

$$B \equiv h_x^{\otimes 3} + EP \left(h_x \otimes \zeta^{\otimes 2}\right) + E \left(\zeta^{\otimes 3}\right).$$

7.2 Fourth Order Solution

A fourth order system follows from (33):

$$f_{vvvv} E v_x^{\otimes 4} + f_{vvv} E \left(v_x^{\otimes 2} \otimes v_{xx}\right) \Omega_2 + f_{vv} E (v_x \otimes v_{xxx}) \Omega_3 \tag{48}$$

$$+ f_{vv} E \left(v_{xx}^{\otimes 2}\right) \Omega_4 + f_v E v_{xxxx} = 0.$$

The new expression is v_{xxxx}. As before, apply a fourth order chain rule on $g(h(x) + \zeta x)$:

$$\frac{\partial^4 g(h(x) + \zeta x)}{\partial x^4} = g_{xxxx} \left(h_x^{\otimes 4} + P\left(h_x^{\otimes 3} \otimes \zeta\right) + P\left(h_x^{\otimes 2} \otimes \zeta^{\otimes 2}\right) + P\left(h_x \otimes \zeta^{\otimes 3}\right) + \zeta^{\otimes 4}\right)$$

$$+ g_{xxx} \left(\left(h_x^{\otimes 2} + P(h_x \otimes \zeta) + \zeta^{\otimes 2}\right) \otimes h_{xx}\right) \Omega_2 + g_{xx} ((h_x + \zeta) \otimes h_{xxx}) \Omega_3$$

$$+ g_{xx} \left(h_{xx}^{\otimes 2}\right) \Omega_4 + g_x h_{xxxx}.$$

Substitute in (39) to express v_{xxxx} as a fourth order Polynomial in ζ:

$$v_{xxxx} = V^0_{xxxx} + V^1_{xxxx}\left(P\left(h_x^{\otimes 3}\otimes\zeta\right) + P\left(h_x^{\otimes 2}\otimes\zeta^{\otimes 2}\right) + P\left(h_x\otimes\zeta^{\otimes 3}\right) + \zeta^{\otimes 4}\right) \quad (49)$$
$$+ V^1_{xxx}\left(\left(P\left(h_x\otimes\zeta\right)+\zeta^{\otimes 2}\right)\otimes h_{xx}\right)\Omega_2 + V^1_{xx}\left(\zeta\otimes h_{xxx}\right)\Omega_3,$$

where the new coefficients are:

$$V^0_{xxxx} = \begin{pmatrix} \frac{\partial^4 g(h(x))}{\partial x^4} \\ g_{xxxx} \\ h_{xxxx} \\ 0 \end{pmatrix}, \qquad V^1_{xxxx} = \begin{pmatrix} g_{xxxx} \\ 0 \\ 0 \\ 0 \end{pmatrix}. \quad (50)$$

Substitute in (48) to get a linear system in g_{xxxx}, h_{xxxx}:

$$A + f_{y'}g_{xxxx}B + f_y g_{xxxx} + \left(f_{x'} + f_{y'}g_x\right)h_{xxxx} = 0 \quad (51)$$
$$A \equiv f_{vvvv}Ev_x^{\otimes 4} + f_{vvv}E\left(v_x^{\otimes 2}\otimes v_{xx}\right)\Omega_2 + f_{vv}E\left(v_x\otimes v_{xxx}\right)\Omega_3 + f_{vv}E\left(v_{xx}^{\otimes 2}\right)\Omega_4$$
$$+ f_v V^1_{xxx}E\left(\zeta^{\otimes 2}\otimes h_{xx}\right)\Omega_2 + f_{y'}\left(\frac{\partial^4 g\left(h\left(x\right)\right)}{\partial x^4} - g_{xxxx}h_x^{\otimes 4} - g_x h_{xxxx}\right)$$
$$B \equiv h_x^{\otimes 4} + EP\left(h_x^{\otimes 2}\otimes\zeta^{\otimes 2}\right) + EP\left(h_x\otimes\zeta^{\otimes 3}\right) + E\zeta^{\otimes 4},$$

where the expectations $Ev_x^{\otimes 4}$, $Ev_x^{\otimes 2}\otimes v_{xx}$, $Ev_x\otimes v_{xxx}$ and $Ev_{xx}^{\otimes 2}$ are provided in the appendix. The term $\frac{\partial^4 g(h(x))}{\partial x^4} - g_{xxxx}h_x^{\otimes 4} - g_x h_{xxxx}$ is the fourth order chain rule of $g\left(h\left(x\right)\right)$, excluding the first and last terms.

7.3 Fifth Order Solution

Using the fifth order chain rule (34), a fifth order system is given by:

$$f_{vvvvv}Ev_x^{\otimes 5}+f_{vvvv}E\left(v_x^{\otimes 3}\otimes v_{xx}\right)\Omega_5+f_{vvv}E\left(v_x^{\otimes 2}\otimes v_{xxx}\right)\Omega_6 \tag{52}$$
$$+f_{vvv}E\left(v_x\otimes v_{xx}^{\otimes 2}\right)\Omega_7+f_{vv}E\left(v_x\otimes v_{xxxx}\right)\Omega_8+f_{vv}E\left(v_{xx}\otimes v_{xxx}\right)\Omega_9+f_vEv_{xxxxx}=0.$$

The expression v_{xxxxx} is derived from (39) by applying a fifth-order chain rule on the composite function $g\left(h\left(x\right)+\zeta x\right)$:

$$\frac{\partial^5 g\left(h\left(x\right)+\zeta x\right)}{\partial x^5}=g_{xxxxx}\left(h_x^{\otimes 5}+P\left(h_x^{\otimes 4}\otimes\zeta\right)+P\left(h_x^{\otimes 3}\otimes\zeta^{\otimes 2}\right)+P\left(h_x^{\otimes 2}\otimes\zeta^{\otimes 3}\right)+P\left(h_x\otimes\zeta^{\otimes 4}\right)+\zeta^{\otimes 5}\right)$$
$$+g_{xxxx}\left(\left[h_x^{\otimes 3}+P\left(h_x^{\otimes 2}\otimes\zeta\right)+P\left(h_x\otimes\zeta^{\otimes 2}\right)+\zeta^{\otimes 3}\right]\otimes h_{xx}\right)\Omega_5+g_{xxx}\left(\left[h_x^{\otimes 2}+P\left(h_x\otimes\zeta\right)+\zeta^{\otimes 2}\right]\otimes h_{xxx}\right)\Omega_6$$
$$+g_{xxx}\left(\left(h_x+\zeta\right)\otimes h_{xx}^{\otimes 2}\right)\Omega_7+g_{xx}\left(\left(h_x+\zeta\right)\otimes h_{xxxx}\right)\Omega_8+g_{xx}\left(h_{xx}\otimes h_{xxx}\right)\Omega_9+g_xh_{xxxxx}.$$

Substituting in (39) yields a fifth-order Polynomial in ζ:

$$v_{xxxxx}=V_{xxxxx}^0+V_{xxxxx}^1\left(P\left(h_x^{\otimes 4}\otimes\zeta\right)+P\left(h_x^{\otimes 3}\otimes\zeta^{\otimes 2}\right)+P\left(h_x^{\otimes 2}\otimes\zeta^{\otimes 3}\right)+P\left(h_x\otimes\zeta^{\otimes 4}\right)+\zeta^{\otimes 5}\right)$$
$$+V_{xxxx}^1\left(\left[P\left(h_x^{\otimes 2}\otimes\zeta\right)+P\left(h_x\otimes\zeta^{\otimes 2}\right)+\zeta^{\otimes 3}\right]\otimes h_{xx}\right)\Omega_5+V_{xxx}^1\left(\left[P\left(h_x\otimes\zeta\right)+\zeta^{\otimes 2}\right]\otimes h_{xxx}\right)\Omega_6$$
$$+V_{xxx}^1\left(\zeta\otimes h_{xx}^{\otimes 2}\right)\Omega_7+V_{xx}^1\left(\zeta\otimes h_{xxxx}\right)\Omega_8,$$

where the new coefficients are:

$$V^0_{xxxxx} = \begin{pmatrix} \frac{\partial^5 g(h(x))}{\partial x^5} \\ g_{xxxxx} \\ h_{xxxxx} \\ 0 \end{pmatrix}, \qquad V^1_{xxxxx} = \begin{pmatrix} g_{xxxxx} \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{53}$$

The linear system in g_{xxxxx}, h_{xxxxx} follows from (52):

$$A + f_{y'} g_{xxxxx} B + f_y g_{xxxxx} + \left(f_{x'} + f_{y'} g_x\right) h_{xxxxx} = 0 \tag{54}$$

$$\begin{aligned} A \equiv\ & f_{vvvvv} E v_x^{\otimes 5} + f_{vvvv} E \left(v_x^{\otimes 3} \otimes v_{xx}\right) \Omega_5 + f_{vvv} E \left(v_x^{\otimes 2} \otimes v_{xxx}\right) \Omega_6 + f_{vvv} E \left(v_x \otimes v_{xx}^{\otimes 2}\right) \Omega_7 \\ & + f_{vv} E \left(v_x \otimes v_{xxxx}\right) \Omega_8 + f_{vv} E \left(v_{xx} \otimes v_{xxx}\right) \Omega_9 + f_v V^1_{xxxx} \left(E \left[P \left(h_x \otimes \zeta^{\otimes 2}\right) + \zeta^{\otimes 3}\right] \otimes h_{xx}\right) \Omega_5 \\ & + f_v V^1_{xxx} \left(E \zeta^{\otimes 2} \otimes h_{xxx}\right) \Omega_6 + f_{y'} \left(\frac{\partial^5 g\left(h\left(x\right)\right)}{\partial x^5} - g_{xxxxx} h_x^{\otimes 5} - g_x h_{xxxxx}\right) \end{aligned}$$

$$B \equiv h_x^{\otimes 5} + EP \left(h_x^{\otimes 3} \otimes \zeta^{\otimes 2}\right) + EP \left(h_x^{\otimes 2} \otimes \zeta^{\otimes 3}\right) + EP \left(h_x \otimes \zeta^{\otimes 4}\right) + E\zeta^{\otimes 5}.$$

where $Ev_x^{\otimes 5}, Ev_x^{\otimes 3} \otimes v_{xx}, Ev_x^{\otimes 2} \otimes v_{xxx}, Ev_x \otimes v_{xx}^{\otimes 2}, Ev_x \otimes v_{xxxx}, Ev_{xx} \otimes v_{xxx}$ are derived in the appendix.

8 Solution Algorithm

This section discusses the solution algorithm. The discussion concentrates on the second order system (25), but the same principles apply to higher order solutions, which have the same structure. The second order system can be presented as a Sylvester equation:

$$A + D \begin{pmatrix} g_{xx} \\ h_{xx} \end{pmatrix} B + G \begin{pmatrix} g_{xx} \\ h_{xx} \end{pmatrix} = 0 \tag{55}$$

$$D \equiv \left(f_{y'} \, , \, 0_{n_f \times n_x} \right) \tag{56}$$

$$G \equiv \left(f_y \, , \, f_{x'} + f_{y'} g_x \right) \tag{57}$$

where D and G are $n_f \times n_f$, and A and B are defined by (25).

The number of conditions in this system is $n_f \left(n_x\right)^2$. In general, a k order system has $n_f \left(n_x\right)^k$ conditions and variables. In higher order systems this can be a very large number. To reduce the dimensions of the problem, I follow three procedures. First, the policy functions of exogenous state variables are known. Their derivatives can be calculated directly from (5) and substituted into (55), and the corresponding conditions should be dropped from the system. This reduces the number of unknown derivatives to $\left(n_y + n_x^1\right) \left(n_x\right)^2$. The technical details are provided in the appendix. Here I proceed with the original system (55).

The second procedure, which is more substantial, is to exploit the symmetry of mixed derivatives by solving only the unique derivatives. To do so, postmultiply the second order system by the compression matrix U_2 that extracts the unique derivatives:

$$AU_2 + DXW_2BU_2 + GX = 0 \tag{58}$$

$$X = \begin{pmatrix} g_{xx} \\ h_{xx} \end{pmatrix} U_2 \tag{59}$$

The number of conditions in this system is $n_f \frac{n_x(n_x+1)}{2}$, and X is a matrix of the unique second derivatives. This system is about half the size of the full system. At the fifth order, the system of unique derivatives is approximately 120 (5!) times smaller than the full system.

The term W_2BU_2 deserves some attention. It follows from (25) that this term equals:

$$W_2 \left(h_x^{\otimes 2}\right) U_2 + W_2 \left(E\zeta^{\otimes 2}\right) U_2.$$

The size of $h_x^{\otimes 2}$ and $E\zeta^{\otimes 2}$ is $(n_x)^4$. In the general k order system, the corresponding matrices will have a size of $(n_x)^{2k}$, which is likely to exceed memory limits. However, these matrices do not need to be stored at full size. The second matrix is a sparse matrix with only one nonzero column. Hence, it can be stored as a sparse matrix. The first matrix is part of the expression $W_2 h_x^{\otimes 2} U_2$. This expression can be calculated without creating the big inner matrix, as shown by Moravitz Martin and Van Loan (2007).

Finally, the size of the system can be further reduced by exploiting known results on derivatives with respect to the perturbation parameter. Theorem 7 in Jin and Judd (2002) shows that the perturbation parameter has no first order effect on a k order solution. Schmitt-Grohé and Uribe (2004) prove a similar result for first and second order solutions. Hence, all the elements of $g_x, g_{xx}, \ldots, h_x, h_{xx}, \ldots$ that represent derivatives (or mixed derivatives) with respect to σ once, are equal to zero. To exploit this analytic result, we can redefine the matrix U_k so that it extracts the unique nonzero derivatives. For example, $g_{xxx}U_3$ is a matrix of all unique third derivatives of g, except for $\frac{\partial^3 g(x)}{\partial x_j \partial x_k \partial \sigma} \forall x_j, x_k \in \{x^1, x^2\}$. Similarly, the matrix W_k performs the reverse procedure. This reduces the k order system to size $n_f \left(\binom{n_x+k-1}{k} - \binom{n_x+k-3}{k-1}\right)$.

The Sylsvester equation (58) is solved by the Hessenberg-Schur method of Golub, Nash and Van Loan (1979). This is an extension of Bartels and Stewart (1972) to Sylvester equations of the form $AXB - X + C = 0$. It is performed by the MATLAB function dlyap.m. To implement this function, (58) should be premultiplied by $-G^{-1}$. For the existence and stability of G^{-1} see Kameník (2005). Alternatively, (58) can be solved by standard vectorization, which is more memory consuming. The MATLAB code provided by the paper allows the two options (dlyap.m requires to install the MATLAB Control System Toolbox). Other methods for solving the Sylvester equation are discussed in Binning (2013b) and Kameník (2005).[15]

[15] These methods assume that the matrix B in $AXB - X + C = 0$ is a Kronecker product of another matrix. They cannot be applied on (58) or (55) directly, but (55) can be partitioned into blocks that satisfy the required structure. For example, the block of derivatives with respect to the state variables only (excluding the perturbation parameter) satisfies the condition.

9 Accuracy Tests

The algorithm is tested on four models. The first model is a simple neoclassical growth model with full depreciation, logarithmic utility and Cobb-Douglass production function. The second model is the asset pricing model of Burnside (1998). The third model is Barro (2006), which has non-Gaussian cross moments $M^2, \ldots, M^5$. These three models have closed form solutions, so the perturbation solution can be derived analytically. The numerical algorithm succeeds to replicate the analytical perturbation solution in all three models. The results are given in Levintal (2014).

This section reports a fourth test which uses an artificial model to test the accuracy of the algorithm. The artificial model has no economic meaning, but it can test accuracy for any arbitrary number of variables and shocks. The artificial model is built as follows. Let z denote a column vector of n_z exogenous variables that follow the evolution law:

$$z' = \sigma \tilde{\eta} \epsilon',$$

where $'$ denotes next period value, $\epsilon \sim N(0, I)$ is a vector of n_ϵ iid standard normal shocks, and $\tilde{\eta}$ is a $n_z \times n_\epsilon$ matrix.

Let w denote an endogenous state variable and y denote a control variable. For simplicity, w and y are assumed to be scalars, but the extension of the model to vectors is straightforward (with some notational cost). The model conditions are given by:

$$w' = e^{H_0 z} + Ee^{H_1 w + \frac{\sqrt{2H_2} H_3 z'}{\sqrt{(H_3\tilde{\eta})(H_3\tilde{\eta})^T}}} - 2, \tag{60}$$

$$Ez' = 0,$$

$$Ey' + y = Ee^{G_0 z'} + Ee^{G_1 w' + \frac{\sqrt{2G_2} G_3 z'}{\sqrt{(G_3\tilde{\eta})(G_3\tilde{\eta})^T}}} + e^{G_0 z} + Ee^{G_1 w + \frac{\sqrt{2G_2} G_3 z'}{\sqrt{(G_3\tilde{\eta})(G_3\tilde{\eta})^T}}} - 4,$$

where H_0, G_0, H_3, G_3 are $1 \times n_z$ row vectors, and H_1, G_1, H_2, G_2 are scalars. This model has the required form $E(y', y, x', x) = 0$, where $x = (w, z)$.

The closed form solution of the model is:

$$w' = e^{H_0 z} + e^{H_1 w + H_2 \sigma^2} - 2, \tag{61}$$

$$y = e^{G_0 z} + e^{G_1 w + G_2 \sigma^2} - 2.$$

This is verified by substituting (61) in (60), noting that the expected value of expressions such as $e^{\frac{\sqrt{2H_2} H_3 z'}{\sqrt{(H_3\tilde{\eta})(H_3\tilde{\eta})^T}}}$ is $e^{H_2 \sigma^2}$, because $H_3 z'$ is a scalar that is normally distributed with zero mean and variance $(H_3\tilde{\eta})(H_3\tilde{\eta})^T$. The closed form solution has a deterministic steady state, which is $\bar{z} = 0, \bar{w} = 0, \bar{y} = 0, \sigma = 0$. In addition, the first order effect of σ is zero at $\sigma = 0$, as required by Theorem 7 in Jinn and Judd (2002). These are necessary conditions to apply the algorithm.

To test the accuracy of the algorithm, I solve the derivatives of the solution at the steady state up to fifth order with the MATLAB algorithm and compare the results to the derivatives of the closed form solution with respect to w, z and σ.

The coefficients $H_0, H_1, H_2, G_0, G_1, G_2, \tilde{\eta}$ are drawn randomly. Note that (61) is not necessarily the only solution of the system, nor is it stable. These conditions are relevant for a first order solution but not for higher order solutions. Hence, the first derivatives are taken directly from (61) and used as inputs for higher order solutions which are solved recursively by the MATLAB algorithm.

Table 1 shows results for three models of different size. The smallest model has 2 state variables, 1 control and 5 shocks. The largest model has 7 state variables, 2 controls and 5 shocks. The table presents the log10 of the maximum absolute difference between the derivatives of the closed form solution and the perturbation solution. The table confirms the accuracy of the algorithm, as errors across all derivatives are less than 10^{-10}. Accuracy declines to some extent with the size of the model and the derivative order, due to the accumulation of round-off errors.

10 Application to Models with Rare Disasters

Higher order solutions are necessary in models that exhibit strong nonlinearities. One particular example is the model of Rietz (1988) and Barro (2006), which introduces rare disasters into an otherwise standard asset pricing model. This model has proved useful in matching asset pricing moments, which standard dynamic models fail to explain (Mehra and Prescott 1985). Consequently, a growing body of literature has started to study the effects of disaster risk in stochastic dynamic models, e.g. Gabaix (2011, 2012), Barro (2009), Barro and Jin (2011), Gourio (2012, 2013), Nakamura, Steinsson, Barro and Ursúa (2013), Wachter (2013), Farhi and Gabaix

(2013).

Most of this literature builds on Lucas (1979) model, which admits a closed form solution. Extensions beyond this basic setup require numerical solutions, such as perturbation (Andreasen 2012) or projection (Gourio 2012). This section shows that perturbation solutions to models with disaster risk are sensitive to the order of perturbation. Specifically, the fourth and fifth order terms of these solutions are economically important and should not be ignored. This is demonstrated on two models with rare disasters: Barro (2006) and Gabaix (2012). The models are solved with perturbation up to fifth order, and the solutions are compared to the closed form solution, which is available for these models. The results show that the approximation errors of perturbation solutions are large, even at a third order, but decline to a reasonable level at the fifth order.

10.1 Barro (2006)

Barro's (2006) model consists of a representative agent and a physical asset (tree) that produces A_t goods each period. The supply of the physical asset is fixed at 1 (Lucas 1979). Asset returns are determined by a standard Euler equation:

$$A_t^{-\theta} = e^{-\rho} E_t R_{t+1} A_{t+1}^{-\theta},$$

where A_t is output in period t, $e^{-\rho}$ is the time discount factor, θ is the relative risk aversion coefficient and R_{t+1} denotes asset returns.

The evolution of output is a random walk with drift:

$$\Delta \log A_{t+1} = \gamma + u_{t+1} + v_{t+1}, \tag{62}$$

where $u_{t+1} \sim N(0, \sigma^2)$ is a regular technology shock, and v_{t+1} is an independent disaster shock defined as follows:

$$v_{t+1} = \left\{ \begin{array}{c|c} 0 & 1-p \\ \log(1-b) & p \end{array} \right. \tag{63}$$

The parameter b denotes the economic contraction in a disaster. Following Barro (2006), I assume that the economic contraction in disasters is a random variable with mean Eb. The distribution of b is calibrated as in Barro (2006).

There are two types of securities in the economy: one-period equity and one-period government bond. One-period equity is a claim to get A_{t+1} in $t+1$. It is traded at market value P_t^e, which is determined by the Euler condition:

$$\frac{P_t^e}{A_t} = e^{-\rho} E_t \left(\frac{A_{t+1}}{A_t} \right)^{1-\theta}. \tag{64}$$

One-period government bond is a claim to get the payout x_{t+1} in $t+1$. It is traded at market value P_t^b determined by the Euler condition:

$$P_t^b = e^{-\rho} E_t x_{t+1} \left(\frac{A_{t+1}}{A_t} \right)^{-\theta}. \tag{65}$$

The payout x_{t+1} is defined by:

$$\log x_{t+1} = w_{t+1}, \tag{66}$$

where w_{t+1} is a shock that depends on v_{t+1}. The conditional distribution of w_{t+1} is:

$$w_{t+1}|_{v_{t+1}=0} = 0$$

$$w_{t+1}|_{v_{t+1}=\log(1-b)} = \left\{ \begin{array}{c|c} 0 & 1-q \\ \log(1-b) & q. \end{array} \right.$$

Namely, the government never defaults in normal periods, and it defaults with probability q in disaster periods. Following Barro (2006), I assume that the loss on default in disaster periods equals the economic contraction b. Recall that b is a random variable, so the loss is larger when the economic contraction is larger.

Finally, we are interested in three asset pricing moments: the expected return on equity, the expected return on government bonds and the ratio between the two, which is defined as the equity premium. Denoting the log of these variables by r^e, r^b and τ, respectively, we get three more equations:

$$e^{r_t^e} = E_t\left(\frac{A_{t+1}}{P_t^e}\right) \tag{67}$$

$$e^{r_t^b} = E_t\left(\frac{x_{t+1}}{P_t^b}\right) \tag{68}$$

$$\tau_t = r_t^e - r_t^b. \tag{69}$$

The model is defined by (62)-(69). The state variables are $\Delta \log A_t$ and x_t. The control variables are P_t^e/A_t, P_t^b, r_t^e, r_t^b and τ_t. There are three shocks: u_t, v_t and w_t. Note that the means of v_t and w_t are not zero. Hence, to apply the perturbation algorithm, we need to redefine these shocks as zero mean shocks. This requires to transform (62) and (66) to:

$$\Delta \log A_{t+1} = \gamma + \mu_v + u_{t+1} + (v_{t+1} - \mu_v), \tag{70}$$

$$\log x_{t+1} = \mu_w + (w_{t+1} - \mu_w), \tag{71}$$

where μ_v and μ_w are the means of v and w, respectively. By this formulation, the zero mean shocks are u_t, $v_t - \mu_v$ and $w_t - \mu_w$. Note that u_t is independent of v_t and w_t, but v_t and w_t are not independent. This property will be represented by non-diagonal variance-covariance matrix M^2. This is permissible because the algorithm makes no assumption on the moments $M^2, \ldots, M^5$.

The closed form solutions are given by equations (9), (12) and (13) in Barro (2006) for asymptotically short periods. Since the perturbation algorithm assumes discrete time, I reproduce the solutions for discrete time:

$$r^e = \rho + \theta\gamma - (1/2)\,\theta^2\sigma^2 + \theta\sigma^2 + \log\frac{1 - pEb}{1 - p + pE\,(1-b)^{(1-\theta)}}$$

$$r^b = \rho + \theta\gamma - (1/2)\,\theta^2\sigma^2 + \log\frac{1 - pqEb}{1 - p + p\cdot\left[(1-q)\,E\,(1-b)^{-\theta} + qE\,(1-b)^{1-\theta}\right]}$$

$$\tau = \theta\sigma^2 + \log\frac{1 - pEb}{1 - pqEb}\cdot\frac{1 - p + p\cdot\left[(1-q)\,E\,(1-b)^{-\theta} + qE\,(1-b)^{1-\theta}\right]}{1 - p + pE\,(1-b)^{(1-\theta)}}$$

For a very small p, these solutions are equivalent to equations (9), (12) and (13) in Barro (2006).

Table 2 compares the perturbation solutions of orders 1st, 2nd, 3rd, 4th and 5th to the true (closed-form) solution. The table replicates Table V in Barro (2006). The true solution slightly differs from Barro (2006), because I use discrete time solutions and not continuous time. For a model with no disasters ($p = q = 0$) the low order solutions are very accurate. In this parametrization the precautionary saving effect is very weak so the equity premium is close to zero. This is the well known equity premium puzzle (Mehra and Prescott 1985).

When disaster risk is included, the equity premium rises to 3% in the baseline parametrization. In this case, low order solutions fail to approximate accurately the true solution. Interestingly, accuracy is low even for a third order solution, which yields an equity premium of 1.5%, about half of the true premium. For this case, the fifth order solution provides an equity premium of 2.6%, which is much more accurate. This result holds also for the other parametrizations used in Barro (2006).

Note that when the disaster probability is high ($p = .025$), the equity premium rises from 3.0% to 4.1%. One would expect that the approximation error would also rise. It turns out that the error of the fifth order solution stays at the magnitude of 0.5 percentage points.

10.2 Gabaix (2012)

Another example of a rare-disaster model is Gabaix (2012). The interesting feature of this model is that the effect of the disaster on asset returns is time varying. For instance, in some periods dividends can be highly sensitive to the disaster shock, while in other periods they can be more resilient. The resilience of an equity share is denoted $\hat{H}_t$ and assumed to follow a certain exogenous process. The process is designed in such a way that the model can still be solved analytically. Consequently, the price-dividend ratio of an equity share can be expressed as an increasing function of the resilience $\hat{H}_t$, given by Eq. (13) in Gabaix (2012). Thus, equity shares with high resilience values are traded at higher prices, reflecting lower risk premia.

The technical paper, Levintal (2014), shows how to solve Gabaix (2012) with perturbation. For the full details of the model and the derivation of the closed-form solution see Gabaix (2012). Table 3 compares the perturbation solutions with the closed-form solution. The special property of this model, compared to Barro (2006), is that the price-dividend ratio is a function of the state variable $\hat{H}_t$. This gives an opportunity to study the approximation error of the perturbation solution on a non-constant policy function.

Table 3 presents the closed-form solution of the price-dividend ratio for different values of $\hat{H}_t$, and compare it to perturbation solutions of various orders. The table confirms the previous findings that approximation errors of low order solutions can be economically large, while fifth order solutions can improve accuracy significantly. For instance, the errors of the third order solution range from 12% to 19%, depending on the resilience $\hat{H}_t$. By comparison, the errors of the fifth order solution range from 1% to 5%.

These results suggest that the fourth and fifth order terms are economically important in models with rare disasters. The nonlinearity in these models driven by the precautionary saving motive is not well approximated by low order solutions when the precautionary saving effect is strong. On the other hand, fifth order solutions seem to approximate the true solution relatively well. This implies that the fourth and fifth order terms of perturbation solutions to models with rare disasters should not be ignored.

11 conclusion

This paper provides fourth and fifth order perturbation solutions to DSGE models, by extending Schmitt-Grohé and Uribe (2004). The paper uses a different notation which is more compact and enables to derive high order solutions with less effort. The solution is implemented by a MATLAB algorithm that is available online. The package includes several features that reduce memory consumption. This enables to solve models of the size of Christiano, Eichenbaum and Evans (2005) up to a fifth order on a regular desktop computer.

The paper shows that the fourth and fifth order terms of perturbation solutions are economically important in models with disaster risk such as Barro (2006) and Gabaix (2012). These models embed strong nonlinearities due to the precautionary saving effect. This makes the accuracy of the solution sensitive to the perturbation order. It is shown that fifth order solutions are able to approximate the true solution with a reasonable error, whereas lower order solutions such as second and third orders contain errors that are economically large. This result suggests that fifth order perturbation solutions can be useful for solving models with rare disasters, which are the subject of a growing literature.

A Appendix

A.1 Unfolding Tensor Contractions

Tensor contraction is a general term that refers to tensor products. High order differentiations produce a certain type of tensor contraction, e.g. $[f_{vv}]^q_{\alpha\beta}\,[v_{xx}]^\beta_{rs}\,[v_x]^\alpha_t$. In this tensor contraction, the tensor f_{vv} is symmetric in its second and third dimensions. This appendix shows how to unfold (reshape) this expression into a matrix. The appendix follows the previous notation that encloses tensors by square brackets and denote matrices without brackets. For a general treatment of this subject and further examples of tensor contractions and their unfoldings see Ragnarsson and Van Loan (2012a, 2012b).

Proposition 1 *Consider a $N+1$ dimensional tensor $[A]$ with dimensions $m \times n \times n \times \cdots \times n$. Define a new $N+1$ dimensional tensor $[T]$ whose $i, j_1, \ldots, j_N$ element* *is:*

$$[T]^i_{j_1 \dots j_N} = [A]^i_{\alpha_1 \dots \alpha_N} [B_N]^{\alpha_N}_{j_1} \dots [B_1]^{\alpha_1}_{j_N}, \tag{72}$$

where $[B_1], \dots, [B_N]$ *are 2-dimensional tensors (matrices) whose first dimension is* n.[16] *If* $[A]$ *is symmetric in its second to last dimensions, then tensor* $[T]$ *is a reshaped form of the following matrix:*

$$T = A (B_1 \otimes \cdots \otimes B_N), \tag{73}$$

where A *is the tensor* $[A]$ *reshaped into a matrix of dimensions* $m \times n^N$. *Note that the order of the matrices* $B_1, \dots, B_N$ *is reversed from tensor to matrix notation.*

Proof. The symmetry of A implies that $[A]^i_{\alpha_1 \dots \alpha_N} = [A]^i_{\alpha_N \dots \alpha_1}$. Substitute in (72):

$$[T]^i_{j_1 \dots j_N} = [A]^i_{\alpha_N \dots \alpha_1} [B_N]^{\alpha_N}_{j_1} \dots [B_1]^{\alpha_1}_{j_N}. \tag{74}$$

The unfolding of (74) into (73) is proved by Ragnarsson and Van Loan (2012a), examples (4.10)-(4.11), for a transposed form of (73) assuming $m = 1$. The extension to $m > 1$ is straightforward. ■

[16] Expression (72) can be stated by the more common notation $T(i, j_1, \dots, j_N) = \sum_{\alpha_1=1}^{n} \dots \sum_{\alpha_N=1}^{n} A(i, \alpha_1, \dots, \alpha_N) B_N(\alpha_N, j_1) \cdots B_1(\alpha_1, j_N)$.

A.2 Chain Rules in Matrix Form

This appendix explains how to derive the matrix form of the high order chain rules. As an example, take the third order chain rule (28). This rule consists of five tensors of dimensions $n_f \times n_x \times n_x \times n_x$, but three of them are permutations of the same tensor as listed in (29). Denote the three unique tensors by T_1, T_2, T_3. The third order chain rule is of the form:

$$\frac{\partial^3 f_i\left(v\left(x\right)\right)}{\partial x_j \partial x_k \partial x_l} = T_1\left(i, l, k, j\right) + T_2\left(i, k, l, j\right) + T_2\left(i, j, l, k\right) + T_2\left(i, j, k, l\right) + T_3\left(i, j, k, l\right).$$

Calculating the entire tensor of third order derivatives requires to permute T_1, T_2, T_3, so that the i, j, k, l elements of each of the five permuted tensors been summed correspond to the correct elements of the original (unpermuted) tensors. This type of permutation is performed in MATLAB by the `ipermute` function. Note that the i index is always the first index, so the permutations apply only to the second to last indices. Moreover, the first and last tensors need not be permuted, because both are symmetric in their second to last indices. Only the three middle terms need to be permuted. This holds for all high order chain rules.

To get the chain rule in matrix form, the tensors T_1, T_2, T_3 are reshaped (unfolded) into $n_f \times n_x^3$ matrices by proposition 1. For instance, the tensor $\left[T_2\right]^q_{rst} = \left[f_{vv}\right]^q_{\alpha\beta}\left[v_{xx}\right]^\beta_{rs}\left[v_x\right]^\alpha_t$ is unfolded into $f_{vv}\left(v_x \otimes v_{xx}\right)$.

Implementing the permutations in matrix form requires to reorder the columns of these matrices, since the columns correspond to the second to last indices of the original tensors. To do so, define a sparse matrix Ω_1^1 such that $T_2\Omega_1^1$ implements the first permutation in (29). Similarly, let Ω_1^2, Ω_1^3 denote the permutation matrices that correspond to the second and third permutations in (29). Then, the sum of the three permutations is implemented by $T_2\left(\Omega_1^1 + \Omega_2^1 + \Omega_1^3\right)$. Define the matrix $\Omega_1 = \Omega_1^1 + \Omega_2^1 + \Omega_1^3$, so that the sum of permutations is $T_2\Omega_1 = f_{vv}\left(v_x \otimes v_{xx}\right)\Omega_1$, which is the second term in (32). All other expressions are derived in a similar way.

The MATLAB package contains the function `create_OMEGA` that calculates all the Ω matrices. This is done by reshaping the identity matrix into a tensor, performing the sum of permutations, and reshaping back into a matrix. The functions `chain3`, `chain4` and `chain5` take the Ω matrices as inputs and calculate the third, fourth and fifth order chain rules. If the Ω matrices are not supplied, the functions perform the permutations on the full tensors by the `ipermute` function.

A.3 Expectation of Kronecker Products

This appendix shows how to calculate the expected value of Polynomials in ζ that appear in high order solutions. Expressions such as $(A\zeta)^{\otimes k}$, where A is a $n_A \times n_x$ matrix, are calculated using the sparse structure of ζ defined in (6):

$$\begin{aligned} E\left(A\zeta\right)^{\otimes k} &= \left(0_{(n_A)^k \times \left((n_x)^k - 1\right)}, E\left(A\eta\epsilon\right)^{\otimes k}\right) \\ &= \left(0_{(n_A)^k \times \left((n_x)^k - 1\right)}, \left((A\eta)^{\otimes k} E\epsilon^{\otimes k}\right)\right) \\ &= \left(0_{(n_A)^k \times \left((n_x)^k - 1\right)}, (A\eta)^{\otimes k} \operatorname{vec}\left(M^k\right)\right) \end{aligned} \tag{75}$$

where $\text{vec}\left(M^k\right)$ denotes vectorization of the k'th cross moment of ϵ. The following is a more general form:

$$E\left(A\zeta\right)\otimes\left(B\zeta\right)\otimes\left(C\zeta\right)=\left(0_{n_An_Bn_C\times\left((n_x)^3-1\right)},\left(\left(A\eta\right)\otimes\left(B\eta\right)\otimes\left(C\eta\right)\right)\text{vec}\left(M^3\right)\right).$$

The expression $\zeta\otimes A\otimes\zeta$ can be stated as follows $\zeta\otimes A\otimes\zeta=Q\left(\zeta^{\otimes 2}\otimes A\right)P$, where Q and P are permutation matrices. Then, the expected value of the middle term is calculated by the previous tools.

Finally, to calculate expressions such as $E\left(A\zeta\right)\otimes\left(B\left(C\otimes\zeta\right)\right)$ use the Kronecker property $\left(AB\right)\otimes\left(CD\right)=\left(A\otimes C\right)\left(B\otimes D\right)$ to get $E\left(A\otimes B\right)\left(\zeta\otimes C\otimes\zeta\right)$.

A.4 Fourth and fifth order expectations

The fourth order system contains the following expectation terms:

$$
\begin{aligned}
Ev_x^{\otimes 4} &= \left(V_x^0\right)^{\otimes 4} + E\left\{P\left(V_x^0 \otimes \left(V_x^1\zeta\right)^{\otimes 3}\right) + P\left(\left(V_x^0\right)^{\otimes 2} \otimes \left(V_x^1\zeta\right)^{\otimes 2}\right) + \left(V_x^1\zeta\right)^{\otimes 4}\right\} \\
Ev_x^{\otimes 2} \otimes v_{xx} &= \left(V_x^0\right)^{\otimes 2} \otimes V_{xx}^0 + E\left\{\left(V_x^1\zeta\right)^{\otimes 2} \otimes V_{xx}^0 + \left[P\left(V_x^0 \otimes V_x^1\zeta\right) + \left(V_x^1\zeta\right)^{\otimes 2}\right] \otimes \left[V_{xx}^1 P\left(\zeta \otimes h_x\right)\right]\right. \\
&\quad \left. + \left[\left(V_x^0\right)^{\otimes 2} + P\left(V_x^0 \otimes V_x^1\zeta\right) + \left(V_x^1\zeta\right)^{\otimes 2}\right] \otimes \left(V_{xx}^1\zeta^{\otimes 2}\right)\right\} \\
Ev_x \otimes v_{xxx} &= V_x^0 \otimes V_{xxx}^0 + E\Big\{V_x^0 \otimes \left[V_{xxx}^1 P\left(\zeta^{\otimes 2} \otimes h_x\right) + V_{xxx}^1\left(\zeta^{\otimes 3}\right)\right] \\
&\quad + \left(V_x^1\zeta\right) \otimes \left[V_{xxx}^1 P\left(\zeta \otimes h_x^{\otimes 2}\right) + V_{xxx}^1 P\left(\zeta^{\otimes 2} \otimes h_x\right) + V_{xxx}^1\left(\zeta^{\otimes 3}\right) + V_{xx}^1\left(\zeta \otimes h_{xx}\right)\Omega_1\right]\Big\} \\
Ev_{xx}^{\otimes 2} &= \left(V_{xx}^0\right)^{\otimes 2} + E\Big\{P\left(V_{xx}^0 \otimes \left(V_{xx}^1\zeta^{\otimes 2}\right)\right) + \left(V_{xx}^1 P\left(h_x \otimes \zeta\right)\right)^{\otimes 2} + P\left(\left(V_{xx}^1 P\left(h_x \otimes \zeta\right)\right) \otimes \left(V_{xx}^1\zeta^{\otimes 2}\right)\right) \\
&\quad + \left(V_{xx}^1\zeta^{\otimes 2}\right) \otimes \left(V_{xx}^1\zeta^{\otimes 2}\right)\Big\}
\end{aligned}
$$

The fifth order system contains the following expectation terms:

$$Ev_x^{\otimes 5} = \left(V_x^0\right)^{\otimes 5} + E\left\{P\left(V_x^0 \otimes \left(V_x^1\zeta\right)^{\otimes 4}\right) + P\left(\left(V_x^0\right)^{\otimes 2} \otimes \left(V_x^1\zeta\right)^{\otimes 3}\right) + P\left(\left(V_x^0\right)^{\otimes 3} \otimes \left(V_x^1\zeta\right)^{\otimes 2}\right) + \left(V_x^1\zeta\right)^{\otimes 5}\right\}$$

$$Ev_x^{\otimes 3} \otimes v_{xx} = \left(V_x^0\right)^{\otimes 3} \otimes V_{xx}^0 + E\Bigg\{ \left(V_x^0\right)^{\otimes 3} \otimes V_{xx}^1\zeta^{\otimes 2} + P\left(\left(V_x^0\right)^{\otimes 2} \otimes \left(V_x^1\zeta\right)\right) \otimes \left[V_{xx}^1 P\left(\zeta \otimes h_x\right) + V_{xx}^1\zeta^{\otimes 2}\right]$$
$$+ \left(P\left(V_x^0 \otimes \left(V_x^1\zeta\right)^{\otimes 2}\right) + \left(V_x^1\zeta\right)^{\otimes 3}\right) \otimes \left(V_{xx}^0 + V_{xx}^1 P\left(\zeta \otimes h_x\right) + V_{xx}^1\zeta^{\otimes 2}\right) \Bigg\}$$

$$Ev_x^{\otimes 2} \otimes v_{xxx} = \left(V_x^0\right)^{\otimes 2} \otimes V_{xxx}^0 + \left(V_x^0\right)^{\otimes 2} \otimes \left[V_{xxx}^1 P\left(\zeta^{\otimes 2} \otimes h_x\right) + V_{xxx}^1\left(\zeta^{\otimes 3}\right)\right]$$
$$+ P\left(V_x^0 \otimes \left(V_x^1\zeta\right)\right) \otimes \left[V_{xxx}^1 P\left(\zeta \otimes h_x^{\otimes 2}\right) + V_{xxx}^1 P\left(\zeta^{\otimes 2} \otimes h_x\right) + V_{xxx}^1\left(\zeta^{\otimes 3}\right) + V_{xx}^1\left(\zeta \otimes h_{xx}\right)\Omega_1\right]$$
$$+ \left(V_x^1\zeta\right)^{\otimes 2} \otimes \left[V_{xxx}^0 + V_{xxx}^1 P\left(\zeta \otimes h_x^{\otimes 2}\right) + V_{xxx}^1 P\left(\zeta^{\otimes 2} \otimes h_x\right) + V_{xxx}^1\left(\zeta^{\otimes 3}\right) + V_{xx}^1\left(\zeta \otimes h_{xx}\right)\Omega_1\right]$$

$$Ev_x \otimes v_{xx}^{\otimes 2} = V_x^0 \otimes \left(V_{xx}^0\right)^{\otimes 2} + E\Bigg\{V_x^0 \otimes \Bigg[P\left(V_{xx}^0 \otimes \left(V_{xx}^1\zeta^{\otimes 2}\right)\right) + P\left(\left(V_{xx}^1 P\left(h_x \otimes \zeta\right)\right) \otimes \left(V_{xx}^1\zeta^{\otimes 2}\right)\right)$$
$$+ \left(V_{xx}^1 P\left(h_x \otimes \zeta\right)\right) \otimes \left(V_{xx}^1 P\left(\zeta \otimes h_x\right)\right) + \left(V_{xx}^1\zeta^{\otimes 2}\right) \otimes \left(V_{xx}^1\zeta^{\otimes 2}\right)\Bigg] + \left(V_x^1\zeta\right) \otimes \Bigg[P\left(\left(V_{xx}^1 P\left(\zeta \otimes h_x\right)\right) \otimes V_{xx}^0\right)$$
$$+ P\left(\left(V_{xx}^1\zeta^{\otimes 2}\right) \otimes V_{xx}^0\right) + P\left(\left(V_{xx}^1\zeta^{\otimes 2}\right) \otimes \left(V_{xx}^1 P\left(\zeta \otimes h_x\right)\right)\right) + \left(V_{xx}^1 P\left(\zeta \otimes h_x\right)\right)^{\otimes 2} + \left(V_{xx}^1\zeta^{\otimes 2}\right)^{\otimes 2}\Bigg]\Bigg\}$$

$$Ev_x \otimes v_{xxxx} = V_x^0 \otimes V_{xxxx}^0 + E\Bigg\{V_x^0 \otimes \Bigg[V_{xxxx}^1\left(P\left(h_x^{\otimes 2} \otimes \zeta^{\otimes 2}\right) + P\left(h_x \otimes \zeta^{\otimes 3}\right) + \zeta^{\otimes 4}\right) + V_{xxx}^1\left(\zeta^{\otimes 2} \otimes h_{xx}\right)\Omega_2\Bigg]$$
$$+ \left(V_x^1\zeta\right) \otimes \Bigg[V_{xxxx}^1\left(P\left(\zeta \otimes h_x^{\otimes 3}\right) + P\left(\zeta^{\otimes 2} \otimes h_x^{\otimes 2}\right) + P\left(\zeta^{\otimes 3} \otimes h_x\right) + \zeta^{\otimes 4}\right)$$
$$+ V_{xxx}^1\left(\left(P\left(h_x \otimes \zeta\right) + \zeta^{\otimes 2}\right) \otimes h_{xx}\right)\Omega_2 + V_{xx}^1\left(\zeta \otimes h_{xxx}\right)\Omega_3\Bigg]\Bigg\}$$

$$Ev_{xx} \otimes v_{xxx} = V_{xx}^0 \otimes V_{xxx}^0 + E\Bigg\{V_{xx}^0 \otimes \Bigg[V_{xxx}^1 P\left(\zeta^{\otimes 2} \otimes h_x\right) + V_{xxx}^1\left(\zeta^{\otimes 3}\right)\Bigg]$$
$$+ \left(V_{xx}^1 P\left(h_x \otimes \zeta\right)\right) \otimes \Bigg[V_{xxx}^1 P\left(\zeta \otimes h_x^{\otimes 2}\right) + V_{xxx}^1 P\left(\zeta^{\otimes 2} \otimes h_x\right) + V_{xxx}^1\left(\zeta^{\otimes 3}\right) + V_{xx}^1\left(\zeta \otimes h_{xx}\right)\Omega_1\Bigg]$$
$$+ \left(V_{xx}^1\zeta^{\otimes 2}\right) \otimes \Bigg[V_{xxx}^0 + V_{xxx}^1 P\left(\zeta \otimes h_x^{\otimes 2}\right) + V_{xxx}^1 P\left(\zeta^{\otimes 2} \otimes h_x\right) + V_{xxx}^1\left(\zeta^{\otimes 3}\right) + V_{xx}^1\left(\zeta \otimes h_{xx}\right)\Omega_1\Bigg]\Bigg\}$$

The tools developed in section A.3 enable to calculate these terms. Most expressions follow a certain pattern. As an example, take the expression $\left(V_{xx}^1 P\left(h_x \otimes \zeta\right)\right) \otimes \left(V_{xxx}^1 P\left(\zeta^{\otimes 2} \otimes h_x\right)\right)$, which is part of $Ev_{xx} \otimes v_{xxx}$ in the fifth order system. For the moment, ignore the P operator (explained in section 7) and focus on $\left(V_{xx}^1\left(h_x \otimes \zeta\right)\right) \otimes$

$(V^1_{xxx}(\zeta^{\otimes 2}\otimes h_x))$. This expression can be written also as:

$$\left(V^1_{xx}\otimes V^1_{xxx}\right)\left(h_x\otimes\zeta^{\otimes 3}\otimes h_x\right).$$

When this expression enters the fifth order system it is premultiplied by f_{vv}. Hence, we are ultimately interested in:

$$f_{vv}\left(V^1_{xx}\otimes V^1_{xxx}\right)\left(h_x\otimes\zeta^{\otimes 3}\otimes h_x\right).$$

The MATLAB package contains the function `permutekron.m` which calculates the expected value of terms of this structure. The stochastic part is the second matrix. Its expected value is calculated by the tools of Appendix A.3. The first product is performed as described in Moravitz Martin and Van Loan (2007), so that large Kronecker products are not created. The function `permutekron.m` also calculates a sum of permutations of this expression, which is required when the P operator is taken into account.

There are few cases that have a different structure, which are calculated separately. These include expressions that are multiplied by the Ω matrices, for instance $V_x\otimes(V^1_{xxx}(\zeta^{\otimes 2}\otimes h_{xx})\Omega_2)$, which is part of $Ev_x\otimes v_{xxxx}$ from the fifth order system. This expression can be written as:

$$\left(V_x\otimes V^1_{xxx}\right)\left(I_x\otimes\zeta^{\otimes 2}\otimes h_{xx}\right)\left(I_x\otimes\Omega_2\right).$$

From here we proceed with the previous tools.

A.5 Reducing the size of the system

To reduce the size of the system, the MATLAB code enables to exclude the policy functions of exogenous state variables from the model conditions, so that the number of conditions falls to $n_f = n_y + n_x^1$. To do so, write the second order system (55) as:

$$A + D \begin{pmatrix} g_{xx} \\ h_{xx}^1 \\ h_{xx}^2 \\ 0 \end{pmatrix} B + G \begin{pmatrix} g_{xx} \\ h_{xx}^1 \\ h_{xx}^2 \\ 0 \end{pmatrix} = 0$$

where h_{xx}^1, h_{xx}^2 and 0 are the second derivatives of the policy functions of x^1, x^2 and σ. By partitioning D and G, the system can be presented as a linear system of g_{xx} and h_{xx}^1.

References

[1] Andreasen, Martin M., "On the effects of rare disasters and uncertainty shocks for risk premia in non-linear DSGE models," *Review of Economic Dynamics*, 15 (2012), 295-316.

[2] Andreasen, Martin M., Jesús Fernández-Villaverde and Juan F. Rubio-Ramírez, "The Pruned State-Space System for Non-Linear DSGE Models: Theory and Empirical Applications," NBER Working Paper 18983, April 2013.

[3] Aruoba, Borağan S., Jesús Fernández-Villaverde and Juan F. Rubio-Ramírez, "Comparing solution methods for dynamic equilibrium economies," *Journal of Economic Dynamics and Control*, 30 (2006), 2477-2508.

[4] Ballard, Grey, Tamara Kolda and Todd Plantenga, "Efficiently Computing Tensor Eigenvalues on a GPU," *IEEE International Parallel and Distributed Processing Symposium*, 2011, 1340-1348.

[5] Barro, Robert J., "Rare Disasters and Asset Markets in the Twentieth Century," *Quarterly Journal of Economics*, 121 (2006), 823-866.

[6] Barro, Robert J., "Rare Disasters, Asset Prices, and Welfare Costs," *The American Economic Review*, 99 (2009), 243-264.

[7] Barro, Robert J. and Tao Jin, "On the Size Distribution of Macroeconomic Disasters," *Econometrica*, 79 (2011), 15671589.

[8] Bartels, R. H. and G. W. Stewart, "Solution of the matrix equation AX+XB=C", *Communications of the ACM*, 15 (1972), 820826.

[9] Benigno, Gianluca, Pierpaolo Benigno and Salvatore Nisticò, "Second-order approximation of dynamic models with time-varying risk," *Journal of Economic Dynamics and Control*, 37 (2013), 1231-1247.

[10] Binning, Andrew, "Third-order approximation of dynamic models without the use of tensors," Working Paper Monetary Policy, Norges Bank, 2013a.

[11] Binning, Andrew, "Solving second and third-order approximations to DSGE models: A recursive Sylvester equation solution," Working Paper Monetary Policy, Norges Bank, 2013b.

[12] Blanchard, Olivier J, and Charles M. Kahn, "The Solution of Linear Difference Models under Rational Expectations," *Econometrica*, 48 (1980), 1305-1311.

[13] Burnside, Craig, "Solving asset pricing models with Gaussian shocks," *Journal of Economic Dynamics and Control*, 22 (1998), 329-340.

[14] Christiano, Lawrence J., Martin Eichenbaum and Charles L. Evans, "Nominal Rigidities and the Dynamic Effects of a Shock to Monetary Policy," *Journal of Political Economy*, 113 (2005), 1-45.

[15] Constantine, G. M. and T. H. Savits, "A Multivariate Faa di Bruno Formula with Applications," *Transactions of the American Mathematical Society*, 348 (1996), 503-520.

[16] Farhi, Emmanuel and Xavier Gabaix, "Rare Disasters and Exchange Rates," Working Paper, 2013.

[17] Fernández-Villaverde, Jesús, Pablo Guerrón-Quintana, Juan F. Rubio-Ramírez and Martin Uribe,"Risk Matters: The Real Effects of Volatility Shocks," *The American Economic Review*, 101 (2011), 2530-2561.

[18] Foerster, Andrew, Juan Rubio-Ramírez, Daniel F. Waggoner, and Tao Zha, "Perturbation Methods for Markov-Switching DSGE Models," 2013.

[19] Gabaix, Xavier, "Variable Rare Disasters: An Exactly Solved Framework for Ten Puzzles in Macro-Finance", *Quarterly Journal of Economics*, 127 (2012), 645-700.

[20] Gabaix, Xavier, "Disasterization: A Tractable Way to Fix the Asset Pricing Properties of Macroeconomic Models", *American Economic Review, Papers and Proceedings*, 101 (2011), 406-409.

[21] Gaspar, Jess and Kenneth L. Judd, "Solving Large-Scale Rational-Expectations Models," NBER Technical Working Papers 207, 1997.

[22] Golub, G., S. Nash and C. Van Loan, "A Hessenberg-Schur method for the problem AX+XB=C", IEEE Transactions on Automatic Control, 24 (1979), 909-913.

[23] Gomme, Paul and Paul Klein, "Second-order approximation of dynamic models without the use of tensors," *Journal of Economic Dynamics and Control* 35 (2011), 604-615.

[24] Gourio, François, "Disaster Risk and the Business Cycle," *American Economic Review*, 102 (2012), 2734-2766.

[25] Gourio, François, "Credit risk and Disaster risk," *AEJ: Macroeconomics*, 5 (2013), 1-34.

[26] Jinn, He-Hui and Kenneth L. Judd, "Perturbation methods for general dynamic stochastic models", Working Paper, 2002.

[27] Judd, Kenneth L., *Numerical Methods in Economics*, (The MIT Press, Cambridge, Massachusetts, 1998).

[28] Judd, Kenneth L. and Sy-Ming Guu, "Asymptotic methods for aggregate growth models," *Journal of Economic Dynamics and Control*, 21 (1997) 1025-1042.

[29] Kameník, Ondřej, "Solving SDGE Models: A New Algorithm for the Sylvester Equation," *Computational Economics*, 25 (2005), 167187.

[30] Kameník, Ondřej, "DSGE Models with Dynare++. A Tutorial", January 2011.

[31] Kim, Jinill and Sunghyun Kim, "Spurious welfare reversals in international business cycle models," *Journal of International Economics*, 60 (2003), 471-500.

[32] Kim, Jinill, Sunghyun Kim, Ernst Schaumburg and Christopher A. Sims, "Calculating and using second-order accurate solutions of discrete time dynamic equilibrium models," *Journal of Economic Dynamics and Control* 32 (2008), 33973414.

[33] Klein, Paul, "Using the generalized Schur form to solve a multivariate linear rational expectations model," *Journal of Economic Dynamics and Control* 24 (2000), 1405-1423.

[34] Lan, Hong and Alexander Meyer-Gohde, "Solving DSGE models with a nonlinear moving average," Journal of Economic Dynamics and Control, 37 (2013), 26432667.

[35] levintal, Oren, "MATLAB Package for a Fifth Order Perturbation Solution to DSGE Models", technical paper, 2014.

[36] Lucas, Robert E. Jr., "Asset Prices in an Exchange Economy," *Econometrica* 46 (1978), 1429-1445.

[37] Magnus, Jan R. and Heinz Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*, (John Wiley & Sons 1999).

[38] Mehra, Rajnish and Edward C.Prescott, "The Equity Premium: A Puzzle," *Journal of Monetary Economics*, 15 (1985), 145-161.

[39] Moravitz Martin, Carla D. and Charles F. Van Loan, "Shifted Kronecker Product Systems," *SIAM Journal on Matrix Analysis and Applications*, 29 (2007), 184-198.

[40] Nakamura, Emi, Jón Steinsson, Robert Barro and José Ursúa, "Crises and Recoveries in an Empirical Model of Consumption Disasters," *American Economic Journal: Macroeconomics*, 5 (2013), 3574.

[41] Ragnarsson, Stefan and Charles F. Van Loan (2012a), "Block Tensor Unfoldings," *SIAM Journal of Matrix Analysis and Applications*, 33 (2012), 149169.

[42] Ragnarsson, Stefan and Charles F. Van Loan (2012b), "Block Tensors and Symmetric Embeddings," *Linear Algebra and its Applications*, 438 (2012), 853-874.

[43] Rietz, Thomas A., "The Equity Risk Premium," *Journal of Monetary Economics*, 22 (1988), 117-131.

[44] Ruge-Murcia, Francisco, "Estimating nonlinear DSGE models by the simulated method of moments: With an application to business cycles," *Journal of Economic Dynamics and Control* 36 (2012), 914-938.

[45] Schmitt-Grohé, Stephanie and Martín Uribe (2004), "Solving dynamic general equilibrium models using a second-order approximation to the policy function," *Journal of Economic Dynamics and Control*, 28 (2004), 755-775.

[46] Schmitt-Grohé, Stephanie and Martín Uribe (2004), "Optimal Operational Monetary Policy in the Christiano-Eichenbaum-Evans Model of the U.S. Business Cycle", Working Paper, 2004b.

[47] Sims, Christopher A., "Solving Linear Rational Expectations Models," *Computational Economics*, 20 (2001), 1-20.

[48] Swanson, Eric, Gary Anderson and Andrew Levin, "Higher-Order Perturbation Solutions to Dynamic, Discrete-Time Rational Expectations Models," Working Paper, 2005.

[49] Uhlig, Harald, "A toolkit for analysing nonlinear dynamic stochastic models easily," in *Computational Methods for the Study of Dynamic Economies*, Ramon Marimon and Andrew Scott, editors, (Oxford University Press 1999).

[50] Wachter, Jessica A., "Can time-varying risk of rare disasters explain aggregate stock market volatility?," *Journal of Finance*, 68 (2013), 987-1035.

Table 1: Accuracy test on artificial models

derivative order	2	3	4	5
2 states, 1 controls, 5 shocks	-15.4	-16.3	-14.8	-15.4
4 states, 1 controls, 5 shocks	-15.2	-13.4	-12.6	-11.6
7 states, 2 controls, 5 shocks	-14.2	-12.8	-11.8	-10.3

The table presents accuracy tests on three artificial models, constructed as described in Section 9. The first row shows results on a model with 1 endogenous state variable, 1 exogenous state variable, 1 control variable and 5 shocks. The second row has 2 endogenous state variables, 2 exogenous state variables, 1 control and 5 shocks. The third row has 4 endogenous state variables, 3 exogenous state variables, 2 controls and 5 shocks. The table presents the log10 of the maximum absolute difference between the derivatives of the closed form solution and the perturbation solution.

Table 2: Perturbation Solution of Barro (2006)

	1st	2nd	3rd	4th	5th	True solution
			No disasters			
Expected equity rate	0.130	0.128	0.128	0.128	0.128	0.128
Expected bill rate	0.130	0.127	0.127	0.127	0.127	0.127
Equity premium	0.000	0.002	0.002	0.002	0.002	0.002
			Baseline			
Expected equity rate	0.105	0.092	0.083	0.079	0.077	0.076
Expected bill rate	0.105	0.083	0.068	0.057	0.051	0.046
Equity premium	0.000	0.009	0.015	0.021	0.026	0.030
			Low θ (=3)			
Expected equity rate	0.087	0.082	0.079	0.078	0.078	0.078
Expected bill rate	0.087	0.075	0.069	0.066	0.064	0.064
Equity premium	0.000	0.007	0.010	0.012	0.013	0.014
			High p (=.025)			
Expected equity rate	0.094	0.075	0.062	0.056	0.053	0.051
Expected bill rate	0.094	0.063	0.041	0.026	0.017	0.010
Equity premium	0.000	0.012	0.022	0.030	0.036	0.041
			Low q (=.3)			
Expected equity rate	0.105	0.092	0.083	0.079	0.077	0.076
Expected bill rate	0.105	0.082	0.066	0.054	0.047	0.041
Equity premium	0.000	0.010	0.018	0.025	0.029	0.034
			Low γ (=.02)			
Expected equity rate	0.085	0.072	0.063	0.059	0.057	0.056
Expected bill rate	0.085	0.063	0.048	0.037	0.031	0.026
Equity premium	0.000	0.009	0.015	0.021	0.026	0.030
			Low ρ (=.02)			
Expected equity rate	0.095	0.082	0.073	0.069	0.067	0.066
Expected bill rate	0.095	0.073	0.058	0.047	0.041	0.036
Equity premium	0.000	0.009	0.015	0.021	0.026	0.030

This table presents perturbation solutions of Barro (2006) at 1st, 2nd, 3rd, 4th, and 5th orders, and the true (closed-form) solution. Parameter values correspond to Table V in Barro (2006). The baseline parameters are $\theta = 4$, $\sigma = .02$, $\rho = .03$, $\gamma = .025$, $p = .017$ and $q = .4$. The no disaster parametrization assumes $p = q = 0$. Other parametrizations change one parameter at a time (compared to the baseline parametrization).

Table 3: Perturbation Solution of Gabaix (2012)

$\hat{H}_t$	1st	2nd	3rd	4th	5th	True solution
-0.05	8.6	12.3	12.3	13.5	13.9	14.0
-0.04	9.1	12.7	13.0	14.3	14.7	15.0
-0.03	9.6	13.2	13.7	15.1	15.6	15.9
-0.02	10.0	13.7	14.4	15.8	16.5	16.9
-0.01	10.5	14.2	15.1	16.6	17.3	17.9
0.00	11.0	14.6	15.8	17.4	18.2	18.8
0.01	11.5	15.1	16.5	18.1	19.1	19.8
0.02	11.9	15.6	17.2	18.9	19.9	20.8
0.03	12.4	16.0	17.9	19.6	20.8	21.7
0.04	12.9	16.5	18.6	20.4	21.7	22.7
0.05	13.3	17.0	19.3	21.2	22.5	23.7

This table presents perturbation solutions for the price-dividend ratio of Gabaix (2012) at 1st, 2nd, 3rd, 4th, and 5th orders, and the true (closed-form) solution. The solutions are given for different values of resilience $\hat{H}_t$. For technical details see Levintal (2014).