This paper presents an algorithm that computes Taylor series expansions of order \( k=2, 3, 4 \) of the policy function of non-linear rational expectations models. Approximations of order \( k \geq 2 \) may be markedly more accurate than linear approximations \((k=1)\) if the variability of the exogenous shocks is high and/or the model exhibits strong curvature.

JEL classification: C63, C68, C88.

Keywords: Solution methods; Non-linear rational expectations models; Taylor expansions.

(*)Tel.: 49 228 734073; Fax: 49 228 739100; E-mail: kollmann@wiwi.uni-bonn.de
http://www.wiwi.uni-bonn.de/kollmann

I thank Ken Judd, Alan Sutherland and participants at the November 2003 CFS-JEDC conference (Frankfurt) for useful suggestions/discussions.
1. Introduction

Many widely studied stochastic rational expectations models can be expressed as:

\[
E_t G(\omega_t, \xi, \varepsilon_{t+1}) = 0, \quad t \geq 0
\]  

where \( E_t \) denotes the mathematical expectation conditional upon complete information about periods \( t \) and earlier; \( G: R^{2n+m} \rightarrow R^n \) is a function, and \( \omega_t \) is an \( n \times 1 \) vector of variables known at date \( t \); \( \xi \geq 0 \) is a scalar, and \( \varepsilon_{t+1} = (\varepsilon_{1,t+1}; \varepsilon_{2,t+1}; \ldots; \varepsilon_{m,t+1}) \) is an \( m \times 1 \) vector of date \( t+1 \) exogenous independent random variables. The following discussion assumes that \( \varepsilon_t \) has bounded support and these moments:

\[
E_t \varepsilon_{1,t+1} = 0, \quad E_t (\varepsilon_{1,t+1})^2 = 1, \quad E_t (\varepsilon_{1,t+1})^3 = 0, \quad E_t (\varepsilon_{1,t+1})^4 = 3. \quad 1
\]

The solution of (1) is a "policy function"

\[
\omega_{t+1} = f(\omega_t, \xi, \varepsilon_{t+1}, \xi),
\]

that satisfies the condition

\[
E_t G(f(\omega_t, \xi, \varepsilon_{t+1}, \xi), \omega_t, \xi, \varepsilon_{t+1}) = 0, \quad \forall \omega_t, \forall \xi \geq 0.
\]  

When \( G \) is linear, then the solution \( f \) is likewise linear and can easily be computed using well-known algorithms (e.g., Hansen and Sargent (1980), Blanchard and Kahn (1980), Anderson and Moore (1985), Klein (2000) and Sims (2002)). However, most economic models \( (G) \) are non-linear. A widely used approach (e.g., King, Plosser and Rebelo, 1988) consists in taking a linear approximation of non-linear models, around a deterministic steady state. A drawback of that approach is that it does not allow to capture the effect of the volatility of exogenous shocks on the mean values of endogenous variables, as the linearized solution exhibits certainty-equivalence; that method is thus not suited for computing welfare or for the analysis of risk premia on financial assets.

Judd and Guu (1993), Judd and Gaspar (1996) and Judd (1998) propose a general approach for approximating the policy functions of continuous time and discrete time models using Taylor expansions of order \( k>1 \), around a steady state. To date, applications of that approach to discrete time models have mainly focused on quadratic approximations, \( k=2 \). See, e.g., Sims (2000), Collard and Juillard (2001), Schmitt-Grohé and Uribe (2004), and Schauburg (2002) who have produced (and made publicly available) computer programmes for \( k=2 \); several studies have used these programmes for the analysis of medium scale macroeconomic models (e.g., Kollmann (2002, 2003, 2004), Schmitt-Grohé and Uribe (2003) and Kim (2003)). Judd and Jin (2002), Jin (2003) and Juillard (2003) recently developed computer code for discrete time approximations of order \( k>2 \) (I learnt about these contributions after completing most of the work described here).

This paper presents an algorithm for computing approximations of order \( k=2, 3 \) and \( 4 \), using an approach that differs from that used by the papers that were just cited (see discussion in Sect. 2.3 below). The computational approach used here differs from that used by the papers cited above. MATLAB code that implements the present algorithm will be made available on my web page.

---

1 The first to fourth moments of \( \varepsilon_{1,t+1} \) correspond \( \xi \) thus to those of a standard normal random variable. Note that \( E_t (\varepsilon_{1,t+1})^3 = 0 \) holds for any symmetric distribution. The algorithm can easily be adapted to allow for \( E_t (\varepsilon_{1,t+1})^4 \neq 3 \).
It appears that approximations of order \( k \geq 2 \) may be markedly more accurate than linear approximations \((k=1)\). A fourth order approximation may be noticeably more accurate than a second order approximation if the variability of the exogenous shocks is high and/or the model exhibits strong curvature.

Section 2 describes the algorithm (and compares it to previous methods). Section 3 applies it to selected models.

2. The method

I begin by discussing definitions/notation. Throughout this paper, the term "steady state" refers to the deterministic steady state, i.e. to a model solution in which \( \omega_{t+1} = \omega_t = \omega \), \( \varepsilon_{t+1} = 0 \) \( \forall t \), with \( G(\omega, \omega, 0_m) = 0_n \), where \( 0_m \) is a column vector of zeros \((m\) elements). Steady state values are denoted by variables without time subscripts, and \( dz_t = z_t - z \) is the deviation of a variable \( z_t \) from its steady state value.

\( R_n \) denotes a polynomial consisting of powers of order \( n \) and higher of elements of \( \{d \omega_t, \xi \varepsilon_t\}_{t=0} \).

\( h_i^{(n)} \) denotes an \( n \)-th order accurate approximation of variable \( h_i \), in the following sense: \( h_i - h_i^{(n)} = R_{n+1} \). Let \( h_i^{(s)} = h_i^{(s)} - h_i^{(s-1)} \), for \( s > 1 \). Thus, \( h_i^{(2)} = h_i^{(1)} + h_i^{(2)} \), \( h_i^{(3)} = h_i^{(1)} + h_i^{(2)} + h_i^{(3)} \) etc.

If \( a \) and \( b \) are matrices, then \((a;b)\) denotes the matrix obtained by vertically concatenating \( a \) and \( b \) (provided \( a \) and \( b \) have the same number of columns), while \((a,b)\) denotes horizontal concatenation.

Let \( k \) be a column vector with \( N \) elements. \( P_n(k) \), for \( n = 2,3,... \) denotes a column vector consisting of all \( n \)-th order powers and cross-products of the elements of \( k \). In the computer programs, these powers/cross-products are arranged in the following order: \( P_1(k) = k \) and \( P_{s+1}(k) = (k_1 P_s(k); k_2 P_s((k_1; k_2); \ldots; k_{N-1} P_s((k_{N-1}; k_N))); k_N P_s(k_N)) \) for \( s > 1 \), where \( k_i \) is the \( i \)-th element of \( k \).

2.1. First-order approximations

The algorithm for generating second (and higher) order approximations presented here takes as its starting point a first-order accurate (linear) model solution. As discussed above, several solution methods for linear(ized) rational expectations models are available in the literature. Any of these methods could be used to generate higher order accurate solutions. Here, I use Sims' (2002) algorithm (that can be implemented using Chris Sims' computer program gensys, available at www.princeton.edu/~sims/). This section briefly reviews Sims' (2002) approach.

Following Sims (2002) note that (1) implies

\[
G(\omega_{t+1}, \omega_t, \xi \varepsilon_{t+1}) + \Pi \eta_{t+1} = 0, \quad \text{with } E \eta_{t+1} = 0, \tag{2}
\]

where \( \Pi \) is a matrix of size \( n \times p \), where \( p \) equals the number of model equations that include date \( t \) expectations of date \( t+1 \) variables. \( \eta_i \) is function of \( \varepsilon_{t+1} \) (that function is not known a priori).

Sims (2002) shows that the solution of the model can be written as:

\[
y_{t+1} = F(y_t, \xi \varepsilon_{t+1}), \quad x_{t+1} = M(y_{t+1}),
\]

\[
(y_t; x_t) = Z \omega_t, \tag{3}
\]

where \( Z \) is a non-singular \( n \times n \) matrix. \( y_t \) and \( x_t \) are column vectors with \( n_y \) and \( n_x \) elements, respectively, with \( n = n_y + n_x \); \( F \) and \( M \) are functions.
Take a first-order Taylor expansion of (2) around a steady state. This gives:
\[ G0 \partial o_{t+1} = G1 \partial o + G2 \xi \varepsilon_{t+1} + \Pi \eta_{t+1} + R_z, \]  (4)
where \( G0, G1 \) and \( G2 \) are matrices/vectors of size \( n \times n \), \( n \times n \) and \( n \times 1 \), respectively.

Using (3), we can write (4) as:
\[ K0 (dy_{t+1}, x_{t+1}) = K1 (dy_{t+1}, x_{t+1}) + G2 \xi \varepsilon_{t+1} + \Pi \eta_{t+1} + R_z, \]  (5)
where \( K0 \) is the inverse of \( G0 \), \( K1 = G1 \) and \( K\) is the inverse of \( G \).

Sims (2002) shows there exists an \( n \times n \) matrix \( T \) with the following properties:
\[
\begin{bmatrix} I_n & H1 \\ 0 & J1 \end{bmatrix}, \quad T K1 = \begin{bmatrix} F1 & H2 \\ 0 & J2 \end{bmatrix} \quad \text{and} \quad T \Pi = \begin{bmatrix} 0_{n \times p} \\ \Pi^* \end{bmatrix},
\]  (6)
where \( H1, J1 \) are matrices of size \( n \times n \), \( F1, J2 \) are matrices of size \( n \times n \), \( H2 \) is a matrix of size \( n \times n \), \( J2 \) is a matrix of size \( n \times n \), \( \Pi^* \) is a matrix of size \( (n \times p) \) (sizes shown in parentheses).

Premultiplying (5) by \( T \) thus gives a block-recursive system of equations:
\[
dy_{t+1} + H1dx_{t+1} = F1dy_{t} + H2dx_{t} + F2 \xi \varepsilon_{t+1} + R_z, \]  (7)
\[
J1dx_{t+1} = J2dx_{t} + G22 \xi \varepsilon_{t+1} + \Pi \eta_{t+1} + R_z, \]  (8)
where \( F2 \) and \( G22 \) are matrices with \( n_y \) and \( n_x \) rows respectively.

The assumed stationarity of the model solution implies that the eigenvalues of \( F1 \) and \( (J2)^{-1}J1 \) are inside the unit circle. Solving forward (8) yields:
\[
dx_i = -\left\{ \sum_{j=0}^{\infty} ((J2)^{-1}J1)^j ((J2)^{-1}(G22 \xi \varepsilon_{t+1} + \Pi^* \eta_{t+1} + R_z)) \right\} + R_z. \]  (9)
As \( dx = E(dx) \) holds, and \( E(\varepsilon_{t+1}) = E(\eta_{t+1}) = 0 \) \( \forall s \geq 0 \), (9) implies: \( dx_i = R_z \), and thus:
\[
dx_i^{(1)} = 0. \]  (10)

(9) and (7) imply:
\[
dy_{t+1}^{(1)} = F1 dy_{t} + F3 \xi \varepsilon_{t+1}. \]  (11)

2.2. Higher order approximations

To \( n - th \) order accurate solutions \( (n \geq 2) \), take a \( n - th \) order Taylor expansion of (2):
\[ G0 \partial o_{t+1} = G1 \partial o + G2 \xi \varepsilon_{t+1} + \sum_{i=2}^{n} \Theta_i P(d\Psi_{t+1}) + \Pi \eta_{t+1} + R_{n+1}, \]  (12)
where \( \Psi_{t+1} = (\omega_t; \xi; \varepsilon_{t+1}) \), while \( \Theta_\), \..., \( \Theta_n \) are matrices. Using (3), we can transform (12) into:
\[ K0 (dy_{t+1}, x_{t+1}) = K1 (dy_{t+1}, x_{t+1}) + G2 \xi \varepsilon_{t+1} + \sum_{i=2}^{n} \Omega_i P(d\Lambda_{t+1}) + \Pi \eta_{t+1} + R_{n+1}, \]  (13)
where \( \Lambda_{t+1} = (y_{t+1}, x_{t+1}, y_t; x_t; \xi \varepsilon_{t+1}) \), while \( \Omega_\), \..., \( \Omega_n \) are matrices. Using (3), we can transform (13) into:
\[
K0 (dy_{t+1}, x_{t+1}) = K1 (dy_{t+1}, x_{t+1}) + G2 \xi \varepsilon_{t+1} + \sum_{i=2}^{n} \Omega_i P(d\Lambda_{t+1}) + \Pi \eta_{t+1} + R_{n+1}, \]  (13)

Premultiplying (13) by \( T \) (see (6)) yields:
\[
dy_{t+1} + H1dx_{t+1} = F1dy_{t} + H2dx_{t} + F2 \xi \varepsilon_{t+1} + \sum_{i=2}^{n} \Phi_i P(d\Lambda_{t+1}) + R_{n+1}, \]  (14)
\[
J1dx_{t+1} = J2dx_{t} + G22 \xi \varepsilon_{t+1} + \Pi \eta_{t+1} + \sum_{i=2}^{n} \Phi_i P(d\Lambda_{t+1}) + R_{n+1}, \]  (15)
where \( \Phi_1 \) and \( \Phi_2 \) are matrices: \( \Phi(1; \Phi(2) = T \Omega_i \text{ for } i = 2, ..., n \).

Solving (15) forward and taking conditional expectations gives:

---

2 In Chris Sims’ gensys program, \( T \) corresponds to the product of the matrices \( tmat \) and \( q^* \): \( T = tmat * q \).

3 \( \Omega_i \) (for \( i = 1, ..., n \)) is a function of \( \Theta_\) and of \( Z \). Determining \( \Omega_i \) is simple but tedious. Interested readers may consult the computer code for the details. Analogous remarks apply to many of the coefficients in the rest of this paper.
\[ dx_i^{(n)} = -\sum_{s=0}^{\infty} ((J2)^{-1} J_1)^s (J2)^{-1} E_i \{ \sum_{i=2}^{\infty} \Phi 2 (P_i(d\Lambda_{r+1}))^{(n)} \}. \]  

(16)

Thus, to determine \( dx_i^{(n)} \) we have to compute the path of the 2nd to \( n-\text{th} \) order powers and cross-products of the state variables. Given such a path, and a solution for \( dx_i^{(n)} \), the time path for \( (dy_{r+1})^{(n)} \) can be determined recursively using (14). An \( n-\text{th} \) order accurate solution for \( \omega_i \) can then be computed using (3): \( (d\omega_i)^{(n)} = Z^{-1}((dy_i)^{(n)}; (dx_i)^{(n)}) \).

The algorithm described below is based on the fact that \( P_i(d\Lambda_{r+1})^{(n)} \) (for \( i = 2, \ldots, n \)) can be determined from \( (d\Lambda_{r+1})^{(1)}, (d\Lambda_{r+1})^{(2)}, \ldots, (d\Lambda_{r+1})^{(n-1)} \). For example, a second order accurate model solution requires knowledge of \( P_2(d\Lambda_r)^{(2)} \). \( P_2(d\Lambda_r) \) is a vector consisting of the products of the elements of \( d\Lambda_r \). Let \( dk_i \) and \( dq_i \) be two elements of \( d\Lambda_r \). Note that

\[ (dk_i dq_i)^{(2)} = (dk_i)^{(1)} dq_i^{(1)}. \]  

(17)

To generate a third order accurate model solution, we need a third order accurate evaluation of products of pairs and triplets of elements of the vector \( d\Lambda_r \). Such an evaluation can be obtained from first- and second order accurate model solution, as the product of two variables \( dk_i \) and \( dq_i \) can be expressed as:

\[ (dk_i dq_i)^{(3)} = (dk_i)^{(1)} (dq_i)^{(1)} + (dk_i)^{(1)} (dq_i)^{(2)} + (dk_i)^{(2)} (dq_i)^{(1)}, \]  

(18)

while a third order accurate approximation of the product of three variables \( dk_i, dq_i, dr_i \) is given by:

\[ (dk_i dq_i dr_i)^{(3)} = (dk_i)^{(1)} (dq_i)^{(1)} (dr_i)^{(1)} + (dk_i)^{(1)} (dq_i)^{(2)} (dr_i)^{(1)} + (dk_i)^{(2)} (dq_i)^{(1)} (dr_i)^{(1)}. \]  

(19)

A fourth order accurate model solution requires a fourth order accurate evaluation of products of pairs, triplets, and quadruplets of elements of the vector \( d\Lambda_r \). This can be obtained from first order, second order and third order accurate solutions. Note that

\[ (dk_i dq_i dr_i)^{(4)} = (dk_i)^{(1)} (dq_i)^{(1)} (dr_i)^{(1)} + (dk_i)^{(1)} (dq_i)^{(2)} (dr_i)^{(1)} + (dk_i)^{(2)} (dq_i)^{(1)} (dr_i)^{(1)} + (dk_i)^{(1)} (dq_i)^{(1)} (dr_i)^{(2)} + (dk_i)^{(1)} (dq_i)^{(2)} (dr_i)^{(2)} + (dk_i)^{(2)} (dq_i)^{(1)} (dr_i)^{(2)}, \]  

(20)

\[ (dk_i dq_i dr_i ds_i)^{(4)} = (dk_i)^{(1)} (dq_i)^{(1)} (dr_i)^{(1)} (ds_i)^{(1)} + (dk_i)^{(1)} (dq_i)^{(2)} (dr_i)^{(1)} (ds_i)^{(1)} + (dk_i)^{(2)} (dq_i)^{(1)} (dr_i)^{(1)} (ds_i)^{(1)} + (dk_i)^{(1)} (dq_i)^{(1)} (dr_i)^{(2)} (ds_i)^{(1)} + (dk_i)^{(1)} (dq_i)^{(2)} (dr_i)^{(2)} (ds_i)^{(1)} + (dk_i)^{(2)} (dq_i)^{(1)} (dr_i)^{(2)} (ds_i)^{(1)}. \]  

(21)

2.2.1. Second order accurate solution

For \( n = 2 \), (14) and (16) are given by:

\[ dy_{r+1} + H1 dx_{r+1} = F1 dy_i + H2 dx_i + F2 \xi \epsilon_{r+1} + \Phi 2l P_2(d\Lambda_{r+1}) + R_1, \]  

(23)

\[ dx_i^{(2)} = -\sum_{s=0}^{\infty} ((J2)^{-1} J_1)^s (J2)^{-1} E_i \{ \Phi 22 (P_i(d\Lambda_{r+1}))^{(2)} \}. \]  

(24)

Let

\[ Z2_{r+1} \equiv (\xi^2; P_2((dy_i; \xi \epsilon_{r+1}))), \]  

(25)

Note that \( dk_i dq_i = (dk_i ^{(1)} + R_2)(dq_i ^{(1)} + R_2) = dk_i ^{(1)} dq_i ^{(1)} + R_2. \)
\[ h_{2i} \equiv (\xi^2; P_2(dy_i)). \]  
(26)

(17) implies that \( (P_2(d\Lambda_{r+1}))^{(2)} = P_2((d\Lambda_{r+1})^{(1)}) \). It follows from (10),(11) that
\[ (d\Lambda_{r+1})^{(1)} = (F_1 dy_i + F_2 \xi e_{r+1}; 0_n; dy_i; 0_n, \xi e_{r+1}). \]  
(27)

Hence, \( (P_2(d\Lambda_{r+1}))^{(2)} \) is a linear function of the squares and cross-products of the vector \( (dy_i; \xi e_{r+1}) \), and thus of the elements of \( P_2((dy_i; \xi e_{r+1})) \). Thus we can write
\[ \Phi_{2l} (P_2(d\Lambda_{r+1}))^{(2)} = \Psi_{2l} Z_{2r+1}, \quad \Phi_{22} (P_2(d\Lambda_{r+1}))^{(2)} = \Psi_{22} Z_{2r+1}, \]  
(28)

for some matrices \( \Psi_{2l}, \Psi_{22} \).

As \( dy_i \) and \( e_{r+1} \) are independent, \( E_i Z_{2r+1} \) is a linear function of \( \xi^2 \) and of \( P_2(dy_i) \), and thus:
\[ E_i Z_{2r+1} = \Xi_2 h_{2i}, \]  
(29)

for some matrix \( \Xi_2 \).

The logic that underlies (28) also implies that
\[ (h_{2r+1})^{(2)} = \Psi_{23} Z_{2r+1}, \]  
(30)

for some matrix \( \Psi_{23} \).

Hence,
\[ E_i (h_{2r+1})^{(2)} = \Psi_{23} \Xi_2 h_{2i}, \]  
(31)

(28)-(31) imply that \( \Phi_{22} E_i (P_2(d\Lambda_{r+1}))^{(2)} = \Psi_{22} \Xi_2 (\Phi_{23} \Xi_2)^t h_{2i} \). Substituting this into (24) gives:
\[ dx_i^{(2)} = S_2 h_{2i}, \text{ with } S_2 \equiv -\sum_{s=0}^{\infty} ((J_2)^{-1} J_1)^s (J_2)^{-1} \Psi_{22} \Xi_2 (\Phi_{23} \Xi_2)^t. \]  
(32)

Using (28), (30) and (32), we can write (23) as:
\[ (dy_{r+1})^{(2)} = F_1 dy_i + F_2 \xi e_{r+1} + P_2 Z_{2r+1}, \]  
(33)

for some matrix \( P_2 \).

(32), (33) can also be expressed as:
\[ dx_i^{(2)} = Q_1 \xi^2 + Q_2 P_2(dy_i), \]  
(34)
\[ (dy_{r+1})^{(2)} = F_1 dy_i + F_2 \xi e_{r+1} + F_3 \xi^2 + F_4 P_2((dy_i; \xi e_{r+1}), \]  
(35)

where \( Q_1, Q_2, F_3 \) and \( F_4 \) are matrices/vectors.

(34),(35) have the same form as the second order accurate solutions derived by Sims (2000) and by Schmitt-Grohé and Uribe (2004). Application of the second-order accurate algorithm presented here to several models yielded coefficients \( F_1, F_2, F_3, F_4 \) and \( M_1, M_2 \) that are numerically indistinguishable from coefficients implied by the Sims (2000) algorithm.

2.2.2. Third order accurate solution

For \( n=3 \), (14) and (16) are given by:
\[ dy_{r+1} + H_1 dx_{r+1} = F_1 dy_i + H_2 dx_i + F_2 \xi e_{r+1} + \Phi_{21} P_2(d\Lambda_{r+1}) + \Phi_{31} P_3(d\Lambda_{r+1}) + R_4, \]  
(36)
\[ dx_i^{(3)} = -\sum_{s=0}^{\infty} ((J_2)^{-1} J_1)^s (J_2)^{-1} E_i \{ \Phi_{22} (P_2(d\Lambda_{r+1}))^{(3)} + \Phi_{32} (P_3(d\Lambda_{r+1}))^{(3)} \}. \]  
(37)

Let
\[ Z_{3r+1} = (\xi^2; P_2(dy_i; \xi e_{r+1}); \xi^2(dy_i; \xi e_{r+1}); P_2((dy_i; \xi e_{r+1})), \]  
(38)
\[ \text{and } h_{3i} = (\xi^2; P_2(dy_i); \xi^2(dy_i); P_3(dy_i)). \]  
(39)
(30), (32), (34), (35) imply:

\[ (d\Lambda_{r+1})^{(2)} = (F3\xi^2 + F4P_1(\xi_e; \xi_{e_r})) + S2\Psi23(\xi^2; P_1(\xi_e; \xi_{e_r})) + Q_1\xi^2 + Q_2P_y(y_t); 0_n), \tag{40} \]

i.e. \((d\Lambda_{r+1})^{(2)}\) is a linear function of \(Z_{2_{r+1}}\).

(17), (18), (19), (27), (40) imply that \((P_1(d\Lambda_{r+1}))^{(3)}\) can be expressed as linear functions of \(Z_{3_{r+1}}\). Thus:

\[
\Phi21P_2(d\Lambda_{r+1})^{(3)} + \Phi31P_3(d\Lambda_{r+1})^{(3)} = \Psi31Z_{3_{r+1}}, \tag{41}
\]

\[
\Phi22P_2(d\Lambda_{r+1})^{(3)} + \Phi32P_3(d\Lambda_{r+1})^{(3)} = \Psi32Z_{3_{r+1}}, \tag{42}
\]

for some matrices \(\Psi31\) and \(\Psi32\).

As \(dy_t\) and \(\xi_{r+1}\) are independent, and as the third moments of the elements of the vector \(\xi_{r+1}\) are zero, \(E_tZ_{3_{r+1}}\) is a linear function of \(h_3\):

\[ E_tZ_{3_{r+1}} = \Xi3 \ h_3, \tag{43} \]

for some matrix \(\Xi3\).

The logic that underlies (41),(42) also implies that

\[(h_3)_{r+1}^{(3)} = \Psi33Z_{3_{r+1}}, \tag{44}\]

for some matrix \(\Psi33\). Hence,

\[ E_t(h_3_{r+1})^{(3)} = \Psi33 \Xi3 \ h_3, \tag{45}\]

(42), (43), (45) imply

\[ E_t(\Phi22P_2(d\Lambda_{r+1})^{(3)} + \Phi32P_3(d\Lambda_{r+1})^{(3)}) = \Psi32\Xi3(\Psi33\Xi3)^3h_3, \tag{46}\]

Therefore,

\[ dx_t^{(3)} = S3 \ h_3, \text{ with } S3 = - \sum_{i=0}^{\infty} ((J2)^{-1}J1)^i \left( (J2)^{-1} \psi32 \Xi3 (\psi33 \Xi3)^3 \right), \tag{46} \]

Using (41), (44) and (46), we can write (36) as:

\[ (dy_t^{r+1})^{(3)} = F1 \ dy_t + F2 \ \xi_e + P3 \ Z_{3_{r+1}}, \tag{47} \]

for some matrix \(P3\).

(46), (47) imply:

\[ dx_t^{(3)} = Q1\xi^2 + Q2P_y(dy_t) + Q3\xi^2 dy_t + Q4P_3(dy_t), \tag{48} \]

\[ (dy_t^{r+1})^{(3)} = F1 \ dy_t + F2 \ \xi_e + F3\xi^2 + F4 \ P_1(dy_t; \xi_e) + F5\xi^2(dy_t; \xi_e) + F6 \ P_3(dy_t; \xi_e), \tag{49} \]

where \(Q1, Q2, Q3, Q4, F3, F4, F5\) and \(F6\) are matrices/vectors.

The coefficients of the first order terms in (49) (i.e. \(F1, F2\)) are, by construction identical to the corresponding coefficients in the first- and second-order accurate solutions (11), (35). It appears that the coefficients of the second order terms \(Q1, Q2, F3, F4\) are also identical across the second- and third order accurate solutions (34)-(35) and (48)-(49). A proof of this is provided in the Appendix.

### 2.2.3. Fourth order accurate solution

The derivation of a fourth order accurate solution follows the same logic as the previous discussions. For \(n=4\), (14) and (16) are given by:

\[ dy_{t+1} + HI \ dx_{t+1} = F1 \ dy_t + H2 \ dx_t + F2 \ \xi_e + \Phi21P_2(d\Lambda_{r+1}) + \Phi31P_3(d\Lambda_{r+1}) + \Phi41P_4(d\Lambda_{r+1}) + R_5, \tag{50}\]

\[ dx_t^{(4)} = - \sum_{i=0}^{\infty} (J2)^{-1}J1^i (J2)^{-1}E_t(\Phi22P_2(d\Lambda_{r+1})^{(4)}) + \Phi32P_3(d\Lambda_{r+1})^{(4)} + \Phi42P_4(d\Lambda_{r+1})^{(4)}), \tag{51}\]

(44), (46), (47) and (48), imply:

\[ (d\Lambda_{r+1})^{(3)} = (F5\xi^2(dy_t; \xi_e) + F6P_3(dy_t; \xi_e); S3 \Psi33Z_{3_{r+1}}; \]
Let
\[ Z_{4,t+1} = (\xi^2; P_2(dy_t; \xi \epsilon_{t+1}); \xi^2(dy_t; \xi \epsilon_{t+1}); P_3((dy_t; \xi \epsilon_{t+1}); \xi^4; \xi^2P_2((y_t; \epsilon_{t+1}); P_4((y_t; \epsilon_{t+1}))), \]
\[ h_{4,t} = (\xi^2; P_2(dy_t); \xi^2(dy_t); P_3(dy_t); \xi^4; \xi^2P_2(dy_t); P_4(dy_t)). \]

Using (17)-(22), (27), (40) and (43), we obtain:
\[ \Phi 21(P_2(dA_{t+1}))^{(4)} + \Phi 31(P_3(dA_{t+1}))^{(4)} + \Phi 41(P_4(dA_{t+1}))^{(4)} = \Psi 41 Z_{4,t+1}, \]
\[ \Phi 22(P_2(dA_{t+1}))^{(4)} + \Phi 32(P_3(dA_{t+1}))^{(4)} + \Phi 42(P_4(dA_{t+1}))^{(4)} = \Psi 42 Z_{4,t+1}, \]
for some matrices \( \Psi 41 \) and \( \Psi 42 \). Also,
\[ E(Z_{4,t+1}) = \Xi 4 h_{4,t}, \]
and \( (h_{4,t+1})^{(4)} = \Psi 43 Z_{4,t}, E((h_{4,t+1})^{(4)}) = \Psi 43 \Xi 4 h_{4,t} \), for some matrices \( \Xi 4 \) and \( \Psi 43 \). We have
\[ (dx_t)^{(4)} = S4 h_{4,t}, \]
with \( S4 = \sum_{i=0}^{\infty} ((J2)^{-1}J1)^i (J2)^{-1} \Psi 32 \Xi 4 (\Psi 42 \Xi 4)^i, \]
and
\[ (dy_t)^{(3)} = F1 dy_t + F2 \xi \epsilon_{t+1} + P4 Z_{4,t+1}, \]
for some matrix \( P4 \).

(52), (53) can be written as:
\[ (dx_t)^{(4)} = Q1 \xi^2 + Q2 P_2(dy_t) + Q3 \xi^2 dy_t + Q4 P_3(dy_t) + Q5 \xi^2 + Q6 \xi^2 P_2(dy_t) + Q7 P_4(dy_t), \]
\[ (dy_t)^{(4)} = F1 dy_t + F2 \xi \epsilon_{t+1} + F3 \xi^2 + F4 P_2(dy_t; \xi \epsilon_{t+1}) + F5 \xi^2(dy_t; \xi \epsilon_{t+1}) + F6 P_3(dy_t; \xi \epsilon_{t+1}) + F7 \xi^2 + F8 \xi^2 P_2(y_t; \epsilon_{t+1}) + F9 P_4(y_t; \epsilon_{t+1}), \]
where \( Q1, Q2, Q3, Q4, Q5, Q6, Q7, F3, F4, F5, F6, F7, F8 \) and \( F9 \) are matrices/vectors.

2.3. Related approaches
The computational approach used here differs from that of Judd and Guu (1993), Judd and Gaspar (1996), Judd (1998) (and most subsequent papers that compute Taylor expansions of the policy function). These authors obtain the coefficients of an \( n \)-th order Taylor expansion of by computing the 1st to \( n \)-th order (cross-) partial derivatives of \( H(\omega, \xi) = E(G(f(\omega, \xi \epsilon_{t+1}, \xi), \omega, \xi \epsilon_{t+1})), \) with respect to \( \omega \) and \( \xi \), at the steady state. Note that these derivatives all have to equal zero, as \( H(\omega, \xi) = 0 \quad \forall \omega, \xi \geq 0 \). Thus:
\[ \partial^i H(\omega, \xi)/\partial(\omega, \xi)^i | \omega = \omega, \xi = 0 = 0 \quad \text{for } i = 1, \ldots, n. \]

This gives a system of equations in the (unknown) 1st to \( n \)-th order (cross-) partial derivatives of the policy function \( f \). These partial derivatives can be obtained sequentially: the intercept of \( f(\omega, \xi \epsilon_{t+1}, \xi) \) is determined by the condition \( G(f(\omega, 0, 0), \omega, 0) = 0 \); first-order derivatives can be found by considering (56) with \( i = 1 \). (56) with \( i = 2 \) pins down the second order derivatives, etc.

The basic difference between that approach and the approach used here can be illustrated using the following simple static model (see Judd (1998), p.449):
\[ g(x, \epsilon) = 0, \]
where \( x \) and \( \epsilon \) are an endogenous and an exogenous variable, respectively. The solution of that model is given by a function
\[ x = f(\epsilon) \quad \text{that satisfies} \quad g(f(\epsilon), \epsilon) = 0 \quad \forall \epsilon. \]
We are interested in computing a Taylor expansion of $f$ around benchmark value $\varepsilon_0$: 
\[ \Delta x = f'(\varepsilon_0)\Delta\varepsilon + \frac{1}{2} f''(\varepsilon_0)(\Delta\varepsilon)^2 + \ldots, \]
with $\Delta x = x - x_0$, $\Delta\varepsilon = \varepsilon - \varepsilon_0$, $g(x_0, \varepsilon_0) = 0$. (All derivatives are evaluated at $\varepsilon_0$).

The Judd approach at determining $f'$, $f''$, etc. is based on these conditions:
\[ \frac{\partial^i g(f(\varepsilon), \varepsilon)}{\partial \varepsilon^i} \bigg|_{\varepsilon = \varepsilon_0} = 0 \quad i = 1, \ldots, k. \]

For example: $g_1 f' + g_2 = 0$, which implies that $f' = -g_2/g_1$; $g_{11}(f')^2 + 2g_{12}f' + g_1f'' + g_{22} = 0$.

Substituting $f' = -g_2/g_1$ into this expression allows to determine $f''$:
\[ f'' = -\left[ g_{11}(g_2/g_1)^2 - 2g_{12}(g_1/g_2) + g_{22} \right]. \]

The approach adopted here, by contrast, computes a second-order approximation using a (slightly) different procedure: namely a second-order approximations of the squared terms (of second-order Taylor expansion) is computed using the first-order solution. A first-order Taylor expansion of (57) gives $g_1\Delta x + g_2\Delta \varepsilon + R_2 = 0$, which implies that 
\[ (\Delta x)^{(1)} = -(g_2/g_1) \Delta \varepsilon. \]
A second-order Taylor expansion gives:
\[ g_1\Delta x + g_2\Delta \varepsilon + \frac{1}{2} g_{11}(\Delta x)^2 + g_{12}\Delta x \Delta \varepsilon + \frac{1}{2} g_{22}(\Delta \varepsilon)^2 + R_3 = 0. \]

As $(\Delta x)^{(2)} = (\Delta x)^{(1)})^2$ and $(\Delta \varepsilon)^{(2)} = (\Delta \varepsilon)^{(1)}$ we have:
\[ (\Delta x)^{(2)} = -(g_1)^{-1}[g_2 \Delta \varepsilon + \frac{1}{2} g_{11}((\Delta x)^{(1)})^2 + g_{12}(\Delta x)^{(1)} \Delta \varepsilon + \frac{1}{2} g_{22}(\Delta \varepsilon)^2]. \]

This implies that
\[ (\Delta x)^{(2)} = -(g_2/g_1) \Delta \varepsilon - \frac{1}{2}[g_{11}(g_2/g_1)^2 - 2g_{12}(g_1/g_2) + g_{22}](\Delta \varepsilon)^2. \]

Thus, the implied first and second derivatives of the policy function are identical to those obtained using the Judd approach.

The approach here is closely related to work by, i.a., Kim and Kim (1999), Woodford (1999), and Woodford and Benigno (2003) who have shown that a second-order accurate evaluation of conditional and unconditional expected values of $\omega_t$ (in a model of type (1)) can be achieved using a first-order accurate model solution. These methods exploit the fact that a first-order accurate (i.e. linear) approximation of the policy function permits a second order accurate evaluation of the squares and cross-products of the state variables (and thus of the second moments of these variables). However, the methods presented by these authors do not readily permit to compute simulated time series $\{\omega_t\}$ that are second order accurate.

Sutherland (2002) uses a linear approximation of the model to provide a second-order accurate evaluation of the conditional expected value of the time path $\{E_0(\omega_t)\}_{t \geq 0}$, given the state of the economy at some date $t = 0$. The paper here adapts and generalizes Sutherland's (2002) approach to compute $k$-th-order accurate simulated paths $\{\omega_t\}$.  

### 3. Applying the method

---

5 These papers focus on the computation of welfare. As the vector $\omega_t$ can be specified in such a manner that one of its elements includes the utility level of the agents assumed in the model, that approach is sufficient for computing expected welfare.

6 After the research here was completed, papers by Schaumburg (2002) and Lombardo and Sutherland (2004) were brought to my attention that present second-order accurate solutions based on the same idea.
References
Anderson and Moore (1985)
Blanchard and Kahn (1980)
Collard and Juillard (2001)
Kim and Kim (1999)
Klein (2000)
Sims (2002)
Jin (2003)
Judd and Guu (1993)
Judd and Gaspar (1996)
Judd (1998)
Juillard (2003)
Kim and Kim (1999)
Kim (2003)
King, Plosser and Rebelo (1988)
Lombardo and Sutherland (2004)
Schaumburg (2002)
Sims (2000)
Sims (2002)
Sutherland (2002)
Woodford (1999)
Woodford and Benigno (2003)