Computing Second-Order-Accurate Solutions for Rational Expectation Models using Linear Solution Methods*

Giovanni Lombardo
European Central Bank†

Alan Sutherland
University of St. Andrews‡

October 2005

Abstract

This paper shows how to compute a second-order accurate solution of a non-linear rational expectation model using algorithms developed for the solution of *linear* rational expectation models. The result is a state-space representation for the realized values of the variables of the model. This state-space representation can easily be used to compute impulse responses as well as conditional and unconditional forecasts.

JEL classification: C63, E0.

Keywords Second order approximation; Solution method for rational expectation models.

*In writing this paper we benefited from comments from Thomas Werner, Pierpaolo Benigno, Massimo Rostagno, Steve Ambler and an anonymous referee. Clearly, the content of this paper does not necessarily reflect their views. The content of this paper does not necessarily reflect the views of the European Central Bank.

†European Central Bank, 60311 Frankfurt am Main, Germany. E-mail: giovanni.lombardo@ecb.int.

‡School of Economics and Finance, University of St Andrews, St Andrews, KY16 9AL, UK. E-mail: ajs10@st-and.ac.uk

1 Introduction

This paper shows how algorithms devised for the solution of linear rational expectation models can be effectively employed to solve non-linear rational expectation models that are approximated to the second order of accuracy. Currently, researchers can choose from a number of algorithms for the solution of linear rational expectation models, i.e. models approximated to the first order of accuracy. An incomplete list would include direct methods like Blanchard and Kahn (1980), Sims (2000a) and Klein (2000) and methods based on the undetermined coefficients technique like Uhlig (1999) and Christiano (1998). At the same time a growing macroeconomic literature is addressing issues that can be studied only by taking into account (at least) the second-order terms of the rational expectation models. The welfare-based monetary policy analysis in Woodford (2003) is emblematic of this new focus. A number of papers describe how to derive the second-order expansion of rational expectation models and how to solve the approximated system. A non-exhaustive list should include Schmitt-Grohé and Uribe (2004), Jin and Judd (2002), Sims (2000b), Kim and Kim (2003), Kim *et al* (2003), Benigno and Woodford (2004a, 2004b) and Sutherland (2002). Most of these papers are associated with computer algorithms devised to solve the second-order-approximated models.[1] Yet, these algorithms (with the exception of Sutherland (2002)) are different from those used to solve linear rational expectation models.

In this paper we show that second-order accurate state-space solutions can be obtained simply by use of algorithms devised for linear rational expectations models. The basic structure of the solution technique employed in this paper follows the method suggested by Sutherland (2002). Nevertheless, our paper makes two important extensions to the results shown in Sutherland (2002). Firstly, we are able to derive second-order accurate solutions in *state-space form*. Secondly, we derive second-order accurate solutions for the *realized values* of the variables (as opposed to their conditional forecast). Thus, contrary to what is stated in Sutherland (2002), the two-step solution method described here is as general as any other second-order accurate solution method currently available in the literature (including those described

[1] Benigno and Woodford (2004a, 2004b) represent an exception since their aim is to give an analytical solution to the model. Their approach is nevertheless very similar to that followed by Sutherland (2002). The general method proposed by Sutherland (2002) was developed independently but is similar to the procedure adopted by Canton (1996) in the context of a specific model.

by Schmitt-Grohé and Uribe (2004) and Sims (2000b)).

In a similar way to perturbation-based approaches (e.g. Schmitt-Grohé and Uribe (2004)), our technique relies on the basic principle that "... all the higher-order terms of the Taylor series expansion ... are solutions to linear problems once one computes the first-order terms" (Jinn and Judd (2002, p. 3)). The main difference between our technique and the perturbation approach concerns the way in which the second-order terms are computed. The typical perturbation algorithm requires postulating a solution to the non-linear model. This solution takes the form of a second-order state-space representation with unknown coefficients. The coefficients of the second-order Taylor expansion of the postulated solution are then obtained by imposing consistency with the second order expansion of the original model. These steps are clearly reminiscent of the method of undetermined coefficients used in the literature for the solution of linear rational expectation models.

Our technique, on the contrary, follows a direct solution approach (cf. Klein (2000)), which does not involve postulating explicitly a solution to the non-linear problem. We first solve a *first-order* approximation of the model in order to generate an auto-recursive representation of the *second-order* terms. We then take this auto-recursive structure as an additional forcing process for the linear dynamic system. Solving for the second-order coefficient matrix is then no different from finding the coefficient matrix that multiplies the exogenous forcing process in a linear state-space problem. Our method therefore amounts to a two-step process where each step involves the solution of a standard linear dynamic problem.[2] A further interesting aspect of the method we propose is that it can be described using standard linear algebra notation, of the same type that would be used in linear rational expectations models (as described, for instance, in Ljungqvist and Sargent (2000)).[3]

This paper is organized as follows. In Section 2 we outline the basic structure of the two-step solution procedure. In Section 3 the state-space form of the solutions to each step are described in more detail. Section 4 applies the solution method to the simple neoclassical growth model. This is a convenient benchmark which is used by both Sutherland (2002) and Schmitt-Grohé and Uribe (2004). Section 5 concludes.

[2]It is important to note, however, that our technique does not require solving the generalized eigenvalue problem more than once.

[3]See Juillard (2003) for a "concise" formulation of the perturbation method that relies more heavily on matrix algebra.

2 A Two-Step Solution Method

It is assumed that the second-order approximation of the equations of a model can be written in the following matrix form[4]

$$A_1 \begin{bmatrix} s_{t+1} \\ E_t[c_{t+1}] \end{bmatrix} = A_2 \begin{bmatrix} s_t \\ c_t \end{bmatrix} + A_3 x_t + A_4 \Lambda_t + A_5 E_t[\Lambda_{t+1}] + O(\epsilon^3) \quad (1)$$

$$x_t = N x_{t-1} + \varepsilon_t \quad (2)$$

$$\Lambda_t = \text{vech}\left(\begin{bmatrix} x_t \\ s_t \\ c_t \end{bmatrix} \begin{bmatrix} x_t & s_t & c_t \end{bmatrix} \right) \quad (3)$$

where s is a vector of predetermined variables (i.e. $E_t[s_{t+1}] = s_{t+1}$), c is a vector of jump variables, x is a vector of exogenous forcing processes, ε is a vector of i.i.d. shocks. Λ_t is a vector of all the squares and cross-products of the variables of the model.[5] A_1.. A_5 are matrices of coefficients, E_t is the expectations operator conditional on information at time t and $O(\epsilon^3)$ contains all terms which are of order three or higher in deviations from the point of approximation.[6]

The objective is to use (1) to derive second-order accurate time paths of s and c. The solution method described in this paper is based on the following two observations: (i) second-order accurate solutions to (1) can be obtained using purely linear methods if a second-order accurate solution for the time-path of Λ is known; and (ii) a second-order accurate solution for the time path of Λ can itself be obtained using purely linear solution methods.

[4]The second-order approximation of a model is generated by replacing each side of each equation with a second-order Taylor series expansion around an appropriate point of approximation. It is usually convenient to approximate around a non-stochastic steady state. It is also usually convenient to measure variables as log-deviations from this non-stochastic steady state.

It is important to note that, in taking second-order approximations, expectations operators should be preserved in the positions they arise in the non-approximated model. This is because (unlike the case of first-order approximation) certainty equivalence can not be assumed in the second-order approximated model.

[5]The cross-products could involve variables with different time subscripts. By using the state-space solution discussed below, these cross-products can be easily reduced to products between contemporaneous realizations of the variables, i.e. Λ_t. See the Appendix for an explanation of the *vech* notation.

[6]It is assumed the distribution and dynamics of the exogenous driving processes in the model are such that no x variable can ever deviate from its deterministic steady state by more than ϵ.

The first observation is self-evidently true. If the time path of Λ is known then (1) can be regarded as a linear rational expectations system with exogenous forcing processes Λ and x. Such a system can be solved using any standard linear solution method.

The second observation is less obvious. To understand (ii) notice that terms of order two and above in the behaviour of x, s and c become terms of order three and above in the squares and cross products of x, s and c. It must therefore follow that the second-order accurate behaviour of Λ depends only on the first-order accurate behaviour of x, s and c. Thus it is possible to generate second-order accurate solutions for Λ by considering first-order accurate solutions for x, s and c. First-order accurate solutions for x, s and c can easily be obtained by solving the linear system

$$A_1 \begin{bmatrix} s_{t+1} \\ E_t\left[c_{t+1}\right] \end{bmatrix} = A_2 \begin{bmatrix} s_t \\ c_t \end{bmatrix} + A_3 x_t + O\left(\epsilon^2\right) \tag{4}$$

which is derived from the first-order terms in (1). Here $O\left(\epsilon^2\right)$ contains all terms of order two and above in deviations from the non-stochastic steady state of the model.

It is now simple to state the two-step solution process.

> **Step 1:** *Use the first-order dynamic system (4) to derive a second-order accurate solution for* Λ.
>
> **Step 2:** *Use the solution for* Λ *derived in step 1 and the second-order dynamic system (1) to drive second-order accurate solutions for* s *and* c.

An important difference between the current paper and Sutherland (2002) is that in Step 1 we are able to derive a linear state-space representation of the realised behaviour of Λ. The combination of this linear state-space representation of the dynamics of Λ and (1) yields an augmented system where the dynamics of Λ are treated as an additional set of linear exogenous forcing processes. Thus the non-linear system (1) is recast as a purely linear system with linear forcing processes. The solution to Step 2 can therefore also be written in a simple state-space form which can be used to generate second-order accurate impulse responses or second-order accurate values for conditional first and second moments at any horizon.

3 State-Space Solutions to Steps 1 and 2

In this section we describe the state-space solutions to Steps 1 and 2 in more detail and show explicitly how the second-order (i.e. non-linear) problem can be solved using purely linear solution methods. In this section we stress that what matters is the state-space representation of the solutions, not the particular algorithm used to derive the solutions. In the Appendix we describe in more detail how the QZ decomposition (as described in Klein (2000)) can be used to derive state-space solutions to each step. Matlab codes which implement the solution algorithm described in the Appendix are available from the authors.

3.1 Step 1

The first-order representation of our system (4) can be solved using any standard linear rational expectations method to yield a state-space representation of the following form

$$s_t^f = F_1 x_{t-1} + F_2 s_{t-1}^f \tag{5}$$

$$c_t^f = P_1 x_t + P_2 s_t^f \tag{6}$$

where the superscript 'f' indicates that these are first-order accurate solutions.[7] It is convenient to rewrite this solution in a more compact form as

$$\begin{bmatrix} x_t \\ s_t^f \\ c_t^f \end{bmatrix} = \Omega \begin{bmatrix} x_t \\ s_t^f \end{bmatrix} \tag{7}$$

$$\begin{bmatrix} x_t \\ s_t^f \end{bmatrix} = \Phi \begin{bmatrix} x_{t-1} \\ s_{t-1}^f \end{bmatrix} + \Gamma \varepsilon_t \tag{8}$$

where

$$\Omega = \begin{bmatrix} \underset{(n_x \times n_x)}{I} & \underset{(n_x \times n_s)}{0} \\ \underset{(n_s \times n_x)}{0} & \underset{(n_s \times n_s)}{I} \\ \underset{(n_c \times n_x)}{P_1} & \underset{(n_c \times n_s)}{P_2} \end{bmatrix}, \quad \Phi = \begin{bmatrix} \underset{(n_x \times n_x)}{N} & \underset{(n_x \times n_s)}{0} \\ \underset{(n_s \times n_x)}{F_1} & \underset{(n_s \times n_s)}{F_2} \end{bmatrix}, \quad \Gamma = \begin{bmatrix} \underset{(n_x \times n_\varepsilon)}{I} \\ \underset{(n_s \times n_\varepsilon)}{0} \end{bmatrix} \tag{9}$$

[7] Henceforth to simplify notation the term representing the approximation residual is omitted from all equations.

where n_i denotes the number of elements in vector i. We also define n_{U1} and n_{U2} as the number of rows and the number of columns of any given matrix U, respectively.[8]

Using the matrices L^c and L^h such that[9]

$$\begin{aligned} \text{vech}(\cdot) &= L^c \,\text{vec}(\cdot) \\ L^h \text{vech}(\cdot) &= \text{vec}(\cdot) \end{aligned}$$

it is easy to see that

$$\Lambda_t = RV_t \tag{10}$$

$$V_t = \tilde{\Phi} V_{t-1} + \tilde{\Gamma}\, \tilde{\varepsilon}_t + \tilde{\Psi}\tilde{\xi}_t \tag{11}$$

where

$$\begin{aligned} R &= \underset{(n_{\Lambda 1} \times n_{\Omega 1}^2)}{L^c} (\Omega \otimes \Omega) \underset{(n_{\Omega 2}^2 \times n_{V1})}{L^h} \\ \tilde{\Phi} &= \underset{(n_{V1} \times n_{\Phi 1}^2)}{L^c} (\Phi \otimes \Phi) \underset{(n_{\Phi 2}^2 \times n_{V1})}{L^h} \\ \tilde{\Gamma} &= \underset{(n_{V1} \times n_{\Gamma 1}^2)}{L^c} (\Gamma \otimes \Gamma) \underset{(n_{\Gamma 2}^2 \times n_{\varepsilon}^2)}{L^h} \\ \tilde{\varepsilon}_t &= \text{vech}(\varepsilon_t \, \varepsilon_t') \\ V_t &= \text{vech}(\begin{bmatrix} x_t \\ s_t^f \end{bmatrix} \begin{bmatrix} x_t & s_t^f \end{bmatrix}), \\ \tilde{\Psi} &= \underset{(n_{V1} \times n_{\Phi 1} n_{\Gamma 1})}{L^c} [(\Phi \otimes \Gamma) + (\Gamma \otimes \Phi) P'] \\ \tilde{\xi}_t &= \text{vec}\left(\begin{bmatrix} x_{t-1} \\ s_{t-1}^f \end{bmatrix} \varepsilon_t' \right) \end{aligned}$$

(See the Appendix for a definition of the $\otimes$ operator and also a discussion of the derivation of the 'permutation' matrix P.) Thus a second-order accurate representation of the dynamics of Λ can be written as a self-contained system in state-space form.

[8] In general, we don't specify the size of the matrices if it can be easily inferred from the context.

[9] Note that $L^h \, L^c = I$. See Hamilton (1996, p 300-302). Note also that the use of these matrices is not necessary in order to solve the model. Indeed one could simply vectorize the variance covariance dynamic system (use vec instead of vech). The suggested representation is clearly dictated by efficiency reasons.

3.2 Step 2

We can now use equation (10) to substitute out Λ_t and Λ_{t+1} in equation (1). This gives a new augmented form for the second-order accurate representation of the model as follows

$$A_1 \begin{bmatrix} s_{t+1} \\ E_t\left[c_{t+1}\right] \end{bmatrix} = A_2 \begin{bmatrix} s_t \\ c_t \end{bmatrix} + A_3 x_t + G V_t + H\Sigma \tag{12}$$

$$V_t = \tilde{\Phi} V_{t-1} + \tilde{\Gamma}\,\tilde{\varepsilon}_t + \tilde{\Psi}\tilde{\xi}_t \tag{13}$$

$$x_t = N\, x_{t-1} + \varepsilon_t \tag{14}$$

$$s_t^f = F_1 x_{t-1} + F_2 s_{t-1}^f \tag{15}$$

where[10]

$$G = \left(A_4 R + A_5 R\tilde{\Phi}\right), \quad H = A_5\, R\tilde{\Gamma}, \quad \Sigma = E_t\tilde{\varepsilon}_{t+1} \tag{16}$$

The important point to notice is that this new representation of the second-order approximation of the model can now be solved in state-space form using any linear rational expectations solution method.[11]

It is useful to note that, in solving the augmented system (12) to (16), it is not necessary to solve the (generalized) eigenvalue problem a second time, because the matrices A_1 and A_2 are the same as those that appear in the linear system (4) which was solved in Step 1. For the sake of computational efficiency one could store the solution matrices of the eigenvalue problem (e.g. the QZ decomposition of A_1 and A_2) and use this decomposition in the solution of (12) to (16).[12]

A state-space representation of the solution to our dynamic system is the following

$$s_t = F_1 x_{t-1} + F_2 s_{t-1} + F_3 V_{t-1} + F_4\Sigma \tag{17}$$

$$c_t = P_1 x_t + P_2 s_t + P_3 V_t + P_4\Sigma \tag{18}$$

$$V_t = \tilde{\Phi} V_{t-1} + \tilde{\Gamma}\,\tilde{\varepsilon}_t + \tilde{\Psi}\tilde{\xi}_t \tag{19}$$

$$x_t = N\, x_{t-1} + \varepsilon_t \tag{20}$$

$$s_t^f = F_1 x_{t-1} + F_2 s_{t-1}^f \tag{21}$$

[10] Note that $E_t[\tilde{\xi}_{t+1}] = 0$.

[11] This is despite the presence of the cross-product term $\tilde{\xi}_t$. The cross-product term is zero in expectation and therefore does not affect the forward-looking dynamics of the model. The forward-looking dynamics of the model are therefore entirely linear.

[12] Nevertheless, for relatively small models, solving the eigenvalue problem twice would take only an extra fraction of a second on a typical PC.

For any given initial conditions for s, V and x, this state-space system can be used to generate second-order accurate impulse responses to the exogenous shocks.[13] It can also be used to generate second-order accurate stochastic simulations for computer generated random realisations of the innovations.

Furthermore, the state-space representation provides a convenient way to calculate second-order accurate solutions for conditional first and second moments for the time-paths of the variables of the model. By simply applying the conditional expectation operator through all the equations in (17) to (21) we can compute first and second conditional moments at all horizons.[14]

The solution given by (17) to (21) is also in a form which allows filtering techniques to be applied to second-order accurate simulated data. For example, one could apply any linear filter (e.g. FFT as described in Uhlig (1999)) to the state-space solution and compare second-order-accurate simulated filtered moments with analogous moments computed with real data.

4 An Example: The Neoclassical Growth Model

As an example of the use of the above algorithm consider the simple neoclassical growth model consisting of three equations: an Euler consumption (c) equation, a capital (k) accumulation equation and an i.i.d. process for the

[13] Notice that, in this case, the cross product term $\tilde{\xi}_t$ is zero in all periods because x_{t-1} and s^f_{t-1} are zero in the first period of the impulse response simulation and ε_t is zero in all periods other than the first period of the impulse response simulation. Equation (21) is therefore not relevant for generating an impulse response solution.

[14] An increasing number of macroeconomic papers make use of second-order approximation methods in order to analyze the welfare effects of fiscal and monetary policies as well as in order to derive optimal policies. This requires solutions for first and second moments rather than solutions for realised values. This is in fact the main focus of Sutherland (2002) and Benigno and Woodford (2004a, 2004b). Notice that the cross-product term, ξ_t, is irrelevant for generating expected paths because it is zero in expectation. Equation (21) is therefore also irrelevant in this case.

(log) of the productivity shock (a).[15] That is

$$c_t^{-\gamma} = \alpha\beta E_t\left[a_{t+1}k_{t+1}^{\alpha-1}c_{t+1}^{-\gamma}\right] \tag{22}$$

$$k_{t+1} = a_t k_t^{\alpha} - c_t \tag{23}$$

$$\hat{a}_t \equiv \log a_t = \varepsilon_t \tag{24}$$

The equation-by-equation second-order Taylor expansion of this simple model is as follows (where hats indicate log-deviations from a non-stochastic steady state).

$$-\gamma\hat{c}_t + (1/2)\gamma\hat{c}_t^2 = -\gamma E_t\hat{c}_{t+1} + (\alpha-1)\hat{k}_{t+1} + (1/2)E_t\left[\left(\hat{a}_{t+1} - \gamma\hat{c}_{t+1} + (\alpha-1)\hat{k}_{t+1}\right)^2\right] \tag{25}$$

$$\theta\hat{k}_{t+1} + (1/2)\theta\hat{k}_{t+1}^2 = \hat{a}_t + \alpha\hat{k}_t - \phi\hat{c}_t - (1/2)\phi\hat{c}_t^2 + (1/2)\alpha^2\hat{k}_t^2 + (1/2)\hat{a}_t^2 + \alpha\hat{a}_t\hat{k}_t \tag{26}$$

$$\hat{a}_t = \varepsilon_t \tag{27}$$

where $\phi = \frac{c_{ss}}{c_{ss}+k_{ss}}$, $\theta = \frac{k_{ss}}{c_{ss}+k_{ss}}$. The approximation-error term is not shown for simplicity.[16] Equations (22), (23) and (24) are obtained by replacing each side of equations (25), (26) and (27) with a second-order (logarithmic) Taylor series expansion around the non-stochastic steady state. Notice that the conditional expectations operator which appears in (22) is preserved in equation (25).[17]

Next, we cast the model in matrix notation as follows

$$A_1\begin{bmatrix}\hat{k}_{t+1}\\ E_t\left[\hat{c}_{t+1}\right]\end{bmatrix} = A_2\begin{bmatrix}\hat{k}_t\\ \hat{c}_t\end{bmatrix} + A_3 a_t + A_4\Lambda_t + A_5 E_t\left[\Lambda_{t+1}\right] \tag{28}$$

where

$$\Lambda_t' = \begin{bmatrix}\hat{a}_t^2 & \hat{a}_t\hat{k}_t & \hat{k}_t^2 & \hat{a}_t\hat{c}_t & \hat{k}_t\hat{c}_t & \hat{c}_t^2\end{bmatrix}$$

[15] This model corresponds to one of the examples used by Schmitt-Grohé and Uribe (2004). The assumption of zero persistence in the productivity shock and no depreciation in the capital stock are also made in Schmitt-Grohé and Uribe (2004). These assumptions are made for simplicity only and are not required for the application of the solution algorithm.

[16] Nevertheless, it is useful to recall that this is a *local* approximation and hence the error term might be large for large departures from the approximation point (the steady state in our case) (see Jin and Judd (2002) for a discussion of the importance of the *local* perspective in this kind of exercises).

[17] Note that, by definition, $E_t[k_{t+1}] = k_{t+1}$ and $E_t[a_{t+1}] = 0$.

$$A_1 = \begin{bmatrix} \theta & 0 \\ 1-\alpha & \gamma \end{bmatrix} \quad A_2 = \begin{bmatrix} \alpha & -\phi \\ 0 & \gamma \end{bmatrix} \quad A_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} 1/2 & \alpha & \alpha^2/2 & 0 & 0 & -\phi/2 \\ 0 & 0 & 0 & 0 & 0 & -\gamma/2 \end{bmatrix}$$

$$A_5 = \begin{bmatrix} 0 & 0 & -\theta/2 & 0 & 0 & 0 \\ 1/2 & \alpha-1 & (\alpha-1)^2/2 & -\gamma & -\gamma(\alpha-1) & \gamma^2/2 \end{bmatrix}$$

The following parameter values are used: $\gamma = 2$, $\alpha = 0.3$, $\beta = 0.95$, $\theta = 0.285$, $\phi = 0.715$.

We are now ready to use the two-step algorithm outlined above. Step 1 of the algorithm yields the following state-space representation for the evolution of Λ_t (i.e. equations (10) and (11)):[18]

$$\begin{bmatrix} \hat{a}_t^2 \\ \hat{a}_t\hat{k}_t^f \\ (\hat{k}_t^f)^2 \\ \hat{a}_t\hat{c}_t^f \\ \hat{k}_t^f\hat{c}_t^f \\ (\hat{c}_t^f)^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.84174 & 0.25252 & 0 \\ 0 & 0.84174 & 0.25252 \\ 0.70853 & 0.42512 & 0.063768 \end{bmatrix} \begin{bmatrix} \hat{a}_t^2 \\ \hat{a}_t\hat{k}_t^f \\ (\hat{k}_t^f)^2 \end{bmatrix} \tag{29}$$

$$\begin{bmatrix} \hat{a}_t^2 \\ \hat{a}_t\hat{k}_t^f \\ (\hat{k}_t^f)^2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1.9517 & 1.171 & 0.17565 \end{bmatrix} \begin{bmatrix} \hat{a}_{t-1}^2 \\ \hat{a}_{t-1}\hat{k}_{t-1}^f \\ (\hat{k}_{t-1}^f)^2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \left[\varepsilon_t^2\right] + \begin{bmatrix} 0 & 0 \\ 1.397 & 0.41911 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{a}_{t-1} \\ \hat{k}_{t-1}^f \end{bmatrix} \varepsilon_t \tag{30}$$

$$\begin{bmatrix} \hat{a}_t \\ \hat{k}_t^f \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1.397 & 0.41911 \end{bmatrix} \begin{bmatrix} \hat{a}_{t-1} \\ \hat{k}_{t-1}^f \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} [\varepsilon_t] \tag{31}$$

Step 2 of the algorithm yields the following state-space representation of

[18] In what follows $\hat{k}^f$ and $\hat{c}^f$ denote first-order accurate solutions for capital and consumption while $\hat{k}$ and $\hat{c}$ denote second-order accurate solutions for capital and consumption.

the second-order accurate solution of the model:

$$\begin{bmatrix}\hat{k}_{t+1}\\ \hat{c}_t\end{bmatrix} = \begin{bmatrix}1.397 & 0.41911\\ 0.84174 & 0.25252\end{bmatrix}\begin{bmatrix}\hat{a}_t\\ \hat{k}_t\end{bmatrix} + \frac{1}{2}\begin{bmatrix}-0.077802 & -0.046681 & -0.0070022\\ -0.056866 & -0.034120 & -0.005118\end{bmatrix}\begin{bmatrix}\hat{a}_t^2\\ \hat{a}_t\,\hat{k}_t^f\\ (\hat{k}_t^f)^2\end{bmatrix} + \frac{1}{2}\begin{bmatrix}0.4820\\ -0.1921\end{bmatrix}\sigma^2 \tag{32}$$

These numbers are identical to those reported in Schmitt-Grohé and Uribe (2004) for the same model.

Schmitt-Grohé and Uribe (2004) report results relating to two other models. We have applied our algorithm to both these other examples and have confirmed that it generates identical results to those reported by Schmitt-Grohé and Uribe (2004).

5 Conclusion

In this paper we have shown how a non-linear rational expectation model, approximated to the second order of accuracy, can be recast as a linear structure which can be solved in state-space form by means of standard algorithms developed for the solution of linear rational expectation models. This state-space form can be used to produce second-order accurate impulse responses as well as conditional and unconditional forecasts. We suggest that our algorithm is a convenient alternative to other second-order accurate solution methods proposed in recent literature. Compared to other methods, our algorithm seem to require a much more modest departure from the existing techniques used in dynamic-rational-expectations macroeconomics.

Appendix

Glossary of Matrix Algebra Notation and Rules

1. $vec(X)$**:** Vectorization. All columns of the $m \times n$ matrix X are stacked one under the other (starting with the first column).

2. *vech*(X)**:** As above except that only the upper triangular part of X is considered. Note that it is possible to construct a matrix L such that $L\,\mathrm{vech} = \mathrm{vec}$. Then, $(L'\,L)^{-1}\,L'\,\mathrm{vec}(X) = \mathrm{vech}(X)$.

3. $\otimes$**:** Kronecker product. E.g. $Z = X \otimes Y$. The elements of Z are the product of each element of X with the matrix Y.

4. Vectorization of a product of matrices: $\mathrm{vec}(X\,Y\,Z) = (Z' \otimes X)\mathrm{vec}(Y)$

5. The vec-permutation matrix P Here we show how to construct the permutation matrix P such that $\mathrm{vec}(Z) = P\,\mathrm{vec}(Z')$. We start by noticing that the element $z_{i,j}$ of the generic matrix Z of dimension $m \times n$ will coincide with the element $z^v_{i+m(j-1)}$ in the vector $z^v = \mathrm{vec}\,(Z)$, while it will coincide with the element $\bar{z}^v_{j+n(i-1)}$ in the vector $\bar{z}^v = \mathrm{vec}\,(Z')$. This information can be used to generate the matrix P. Generate an $m \times n$ matrix S such that $S = \mathrm{vec}^{-1}\,([1, 2 \ldots (m \cdot n)]')$, and an identity matrix I of dimension $m\,n \times m\,n$. Finally, the permutation matrix P is given by $P = I\,(:, \mathrm{vec}\,(S'))$.

State-Space Solution to the First-Order System

Consider the first-order system

$$A_1\;E_t\left[\begin{array}{c} s_{t+1} \\ c_{t+1} \end{array}\right] = A_2\left[\begin{array}{c} s_t \\ c_t \end{array}\right] + A_3 x_t \tag{33}$$

$$x_t = N x_{t-1} + \varepsilon_t \tag{34}$$

By applying the QZ decomposition (Generalized Schur Decomposition) we can factorize the matrices A_1 and A_2 into

$$qA_1 z = \left[\begin{array}{cc} a_{11} & a_{12} \\ 0 & a_{22} \end{array}\right], \quad qA_2 z = \left[\begin{array}{cc} b_{11} & b_{12} \\ 0 & b_{22} \end{array}\right]$$

where matrix z has the property $zz' = I$. Hence

$$\left[\begin{array}{cc} a_{11} & a_{12} \\ 0 & a_{22} \end{array}\right] E_t\left[\begin{array}{c} y_{1,t+1} \\ y_{2,t+1} \end{array}\right] = \left[\begin{array}{cc} b_{11} & b_{12} \\ 0 & b_{22} \end{array}\right]\left[\begin{array}{c} y_{1,t} \\ y_{2,t} \end{array}\right] + \left[\begin{array}{c} C_1 \\ C_2 \end{array}\right] x_t \tag{35}$$

where

$$\left[\begin{array}{c} y_{1,t} \\ y_{2,t} \end{array}\right] = \left[\begin{array}{cc} z'_{11} & z'_{21} \\ z'_{12} & z'_{22} \end{array}\right]\left[\begin{array}{c} s_t \\ c_t \end{array}\right]$$

and

$$\left[\begin{array}{c} C_1 \\ C_2 \end{array}\right] = qA_3$$

Without loss of generality we can assume that the system (35) has been ordered so that $b_{22}^{-1}a_{22}$ has roots inside the unit circle. Then the lower part of system (35) can be isolated and solved forward to get (absent bubbles)

$$\hat{y}_{2,t} = -\left[b_{22}^{-1}C_2 + Tb_{22}^{-1}C_2N + T^2b_{22}^{-1}C_2N^2 + ...\right]x_t \tag{36}$$

where

$$T = b_{22}^{-1}a_{22}$$

As long as the series converges we can solve for the endogenous variables as

$$y_{2,t} = -Mx_t$$

where

$$vec\,(M) = \left[I - (N' \otimes T)\right]^{-1} vec\left(b_{22}^{-1}C_2\right)$$

See the Glossary at the start of this Appendix for a general statement of the rule used to derive this expression.[19]

Finally we have

$$\hat{y}_{2,t} \equiv z'_{12}s_t + z'_{22}c_t = -Mx_t$$

so that

$$c_t = P_1x_t + P_2s_t \tag{37}$$

where

$$P_1 = -z_{22}'^{-1}M, \quad P_2 = -\,z_{22}'^{-1}z'_{12}$$

As for the state variables, solving for the upper part of (35) yields

$$\underbrace{\left(a_{11}z'_{21} + a_{12}z'_{22}\right)P_1}_{R_1}E_tx_{t+1} + \underbrace{\left[\left(a_{11}z'_{11} + a_{12}z'_{12}\right) + \left(a_{11}z'_{21} + a_{12}z'_{22}\right)P_2\right]}_{R_2}E_ts_{t+1} =$$
$$\underbrace{\left[\left(b_{11}z'_{21} + b_{12}z'_{22}\right)P_1 + C_1\right]}_{D_1}x_t + \underbrace{\left[\left(b_{11}z'_{11} + b_{12}z'_{12}\right) + \left(b_{11}z'_{21} + b_{12}z'_{22}\right)P_2\right]}_{D_2}s_t$$

Thus

$$E_t\left[R_1x_{t+1} + R_2s_{t+1}\right] = D_1x_t + D_2s_t$$

[19] Klein (referring to King and Watson (2002)) describes a computationally more efficient method to compute M.

or

$$s_{t+1} = \underbrace{\left(R_2^{-1}D_1 - R_2^{-1}R_1N\right)}_{F_1} x_t + \underbrace{R_2^{-1}D_2}_{F_2} s_t$$

where we have made use of the fact that $E_t\, s_{t+1} = s_{t+1}$ (because s is a vector of predetermined variables).

To sum up, the solution to the dynamic system (33) is

$$s_t = F_1 x_{t-1} + F_2 s_{t-1} \tag{38}$$
$$c_t = P_1 x_t + P_2 s_t \tag{39}$$
$$x_t = N\, x_{t-1} + \varepsilon_t \tag{40}$$

This is the solution given in (5) and (6) in the main text.

State-Space Solution to the Second-Order System

Consider now the augmented second-order system

$$A_1 \begin{bmatrix} s_{t+1} \\ E_t\left[c_{t+1}\right] \end{bmatrix} = A_2 \begin{bmatrix} s_t \\ c_t \end{bmatrix} + A_3 x_t + G V_t + H\Sigma \tag{41}$$
$$V_t = \tilde{\Phi} V_{t-1} + \tilde{\Gamma}\, \tilde{\varepsilon}_t + \tilde{\Psi}\tilde{\xi}_t \tag{42}$$
$$x_t = N\, x_{t-1} + \varepsilon_t \tag{43}$$
$$s_t^f = F_1 x_{t-1} + F_2 s_{t-1}^f \tag{44}$$

Define $\bar{V} = (I - \tilde{\Phi})^{-1}\tilde{\Gamma}$ then

$$E_t[V_{t+n}] = \bar{V}\Sigma + \tilde{\Phi}^n(V_t - \bar{V}\Sigma)$$

Applying the QZ decomposition yields

$$\begin{bmatrix} a_{11} & a_{12} \\ 0 & a_{22} \end{bmatrix} E_t \begin{bmatrix} y_{1,t+1} \\ y_{2,t+1} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} \\ 0 & b_{22} \end{bmatrix} \begin{bmatrix} \hat{y}_{1,t} \\ \hat{y}_{2,t} \end{bmatrix} + \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} x_t + \begin{bmatrix} \hat{G}_1 \\ \hat{G}_2 \end{bmatrix} V_t + \begin{bmatrix} \hat{H}_1 \\ \hat{H}_2 \end{bmatrix} \Sigma \tag{45}$$

where the matrices a, b, q and z are all identical to those defined in the previous section and

$$\begin{bmatrix} \hat{G}_1 \\ \hat{G}_2 \end{bmatrix} = qG, \quad \begin{bmatrix} \hat{H}_1 \\ \hat{H}_2 \end{bmatrix} = qH$$

Again the lower part of system (45) can be isolated and solved forward to yield

$$\begin{aligned} y_{2,t} = &- \left[b_{22}^{-1}C_2 + Tb_{22}^{-1}C_2N + T^2b_{22}^{-1}C_2N^2 + ...\right] x_t \\ &- \left[b_{22}^{-1}\hat{G}_2 + Tb_{22}^{-1}\hat{G}_2\tilde{\Phi} + T^2b_{22}^{-1}\hat{G}_2\tilde{\Phi}^2 + ...\right] (V_t - \bar{V}\Sigma) \\ &- \left[I + T + T^2 + ...\right] b_{22}^{-1}(\hat{G}_2\bar{V} + \hat{H}_2)\Sigma \end{aligned} \tag{46}$$

where

$$T = b_{22}^{-1}a_{22}$$

As long as the series converges we can solve for the endogenous variables as

$$y_{2,t} = -M_1x_t - M_2(V_t - \bar{V}\Sigma) - M_3\Sigma$$

where

$$\begin{aligned} vec\,(M_1) &= \left[I - (N' \otimes T)\right]^{-1} vec\left(b_{22}^{-1}C_2\right) \\ vec\,(M_2) &= \left[I - \left(\tilde{\Phi}' \otimes T\right)\right]^{-1} vec\left(b_{22}^{-1}\hat{G}_2\right) \\ M_3 &= \left[I - T\right]^{-1} b_{22}^{-1}(\hat{G}_2\bar{V} + \hat{H}_2) \end{aligned}$$

Finally we have

$$y_{2,t} \equiv z_{12}'s_t + z_{22}'c_t = -M_1x_t - M_2(V_t - \bar{V}\Sigma) - M_3\Sigma$$

so that

$$c_t = P_1x_t + P_2s_t + P_3V_t + P_4\Sigma \tag{47}$$

where

$$\begin{aligned} P_1 &= -z_{22}'^{-1}M_1 \\ P_2 &= -\ z_{22}'^{-1}z_{12}' \\ P_3 &= -z_{22}'^{-1}M_2 \\ P_4 &= -z_{22}'^{-1}[M_3 - M_2\bar{V}] \end{aligned}$$

The solution for the state variables can be obtained by solving for the

upper part of (45). This yields

$$\underbrace{\left(a_{11}z'_{21}+a_{12}z'_{22}\right)P_1}_{R_1}E_tx_{t+1}+\underbrace{\left[\left(a_{11}z'_{11}+a_{12}z'_{12}\right)+\left(a_{11}z'_{21}+a_{12}z'_{22}\right)P_2\right]}_{R_2}E_ts_{t+1}$$
$$+\underbrace{\left(a_{11}z'_{21}+a_{12}z'_{22}\right)P_3}_{R_3}E_tV_{t+1}+\underbrace{\left(a_{11}z'_{21}+a_{12}z'_{22}\right)P_4}_{R_4}\Sigma=$$
$$\underbrace{\left[\left(b_{11}z'_{21}+b_{12}z'_{22}\right)P_1+C_1\right]}_{D_1}x_t+\underbrace{\left[\left(b_{11}z'_{11}+b_{12}z'_{12}\right)+\left(b_{11}z'_{21}+b_{12}z'_{22}\right)P_2\right]}_{D_2}s_t$$
$$+\underbrace{\left[\left(b_{11}z'_{21}+b_{12}z'_{22}\right)P_3+\hat{G}_1\right]}_{D_3}V_t+\underbrace{\left[\left(b_{11}z'_{21}+b_{12}z'_{22}\right)P_4+\hat{H}_1\right]}_{D_4}\Sigma$$

Thus

$$R_1Nx_{t-1}+R_2s_t+R_3(\tilde{\Phi}V_{t-1}+\tilde{\Gamma}\Sigma)+R_4\Sigma=D_1x_{t-1}+D_2s_{t-1}+D_3V_{t-1}+D_4\Sigma$$

or

$$s_t=\underbrace{R_2^{-1}\left(D_1-R_1N\right)}_{F_1}x_{t-1}+\underbrace{R_2^{-1}D_2}_{F_2}s_{t-1}$$
$$+\underbrace{R_2^{-1}(D_3-R_3)}_{F_3}V_{t-1}+\underbrace{R_2^{-1}(D_4-R_4-R_3\tilde{\Gamma})}_{F_4}\Sigma$$

To sum up, the solution to the second-order system (41) is

$$s_t=F_1x_{t-1}+F_2s_{t-1}+F_3V_{t-1}+F_4\Sigma \tag{48}$$
$$c_t=P_1x_t+P_2s_t+P_3V_t+P_4\Sigma \tag{49}$$
$$x_t=N\,x_{t-1}+\varepsilon_t \tag{50}$$
$$V_t=\tilde{\Phi}V_{t-1}+\tilde{\Gamma}\,\tilde{\varepsilon}_t+\tilde{\Psi}\tilde{\xi}_t \tag{51}$$
$$s_t^f=F_1x_{t-1}+F_2s_{t-1}^f \tag{52}$$

This is the state-space form of the second-order solution given in equations (17) to (22) in the main text.

Notice that the QZ decomposition only needs to be applied once in the two-step procedure. The matrices a, b, q and z are the same in both steps, as are the solutions for F_1, F_2, P_1 and P_2.[20]

[20] Only in cases where the realised and expected dynamics differ would it be necessary to compute the QZ decomposition twice.

References

[1] Benigno, P and M Woodford (2004a) "Optimal Monetary and Fiscal Policy: A Linear Quadratic Approach" in Gertler and Rogoff (eds) *NBER Macroeconomics Annual*, 18, 271-333.

[2] Benigno, P and M Woodford (2004b) "Optimal Stabilisation Policy When Wages are Sticky: The Case of a Distorted Steady State" NBER Working Paper 10839.

[3] Blanchard, O J and C M Kahn (1980) "The Solution to Linear Difference Models under Rational Expectations" *Econometrica*, 48, 1305-1311.

[4] Canton, E (1996) "Business Cycles in a Two-Sector Model of Endogenous Growth" CentER Discussion Paper No 96116.

[5] Christiano, L J (1998) "Solving Dynamic Equilibrium Models by a Method of Undetermined Coefficients" NBER Working Paper T0225.

[6] Hamilton, J D (1994) *Time Series Analysis*, Princeton University Press, Princeton.

[7] Jin, H-H and Judd, K J (2002) "Perturbation Methods for General Dynamic Stochastic Models" unpublished manuscript, Stanford University.

[8] Juillard, M (2003) "Solving Stochastic Dynamic Models: A k-Order Perturbation Approach" unpublished manuscript, University Pars 8.

[9] Kim, J, S Kim, E Schaumburg and C Sims (2003) "Calculating and Using Second Order Accurate Solutions of Discrete Time Dynamic Equilibrium Models" unpublished manuscript, Princeton University.

[10] Kim, J and S H Kim (2003) 'Spurious Welfare Reversals in International Business Cycle Models', *Journal of International Economics*, 60, 471-500.

[11] King, R G and M W Watson (2002) "System Reduction and Solution Algorithms for Singular Linear Difference Systems under Rational Expectations" *Computational Economics*, 20, 57-86.

[12] Klein, P (2000) "Using the Generalised Schur Form to Solve a Multivariate Linear Rational Expectations Model" *Journal of Economic Dynamics and Control*, 24, 1405-1423.

[13] Ljungqvist, L and T J. Sargent (2000) *Recursive Macroeconomic Theory*, MIT Press, Cambridge, MA.

[14] Schmitt-Grohé, S and M Uribe (2004) "Solving Dynamic General Equilibrium Models using a Second-Order Approximation to the Policy Function", *Journal of Economic Dynamics and Control*, 28, 755-75.

[15] Sims, C (2000a) "Solving Linear Rational Expectations Models" unpublished manuscript, Princeton University.

[16] Sims, C (2000b) "Second-Order Accurate Solutions of Discrete Time Dynamic Equilibrium Models" unpublished manuscript, Princeton University.

[17] Sutherland, A (2002) "A Simple Second-Order Solution Method for Dynamic General Equilibrium Models" CEPR Discussion Paper No. 3554.

[18] Uhlig , H (1999) "A Toolkit for Analysing Nonlinear Dynamic Stochastic Models Easily" in Marimon and Scott (eds) *Computational Methods for the Study of Dynamic Economies*, OUP, Oxford.

[19] Woodford, M (2003) *Interest and Prices: Foundations of a Theory of Monetary Policy*, Princeton University Press, Princeton.