3

A TOOLKIT FOR ANALYSING NONLINEAR DYNAMIC STOCHASTIC MODELS EASILY

Harald Uhlig[1]

3.1 Introduction

Researchers often wish to analyse nonlinear dynamic discrete-time stochastic models. This chapter provides a toolkit for solving such models easily, building on log-linearizing the necessary equations characterizing the equilibrium and solving for the recursive equilibrium law of motion with the method of undetermined coefficients.[2]

This chapter contains nothing substantially new. Instead, the point of it is to simplify and unify existing methods in order to make them accessible to a large audience of researchers, who may always have been interested in analysing, say, real business cycle models on their own, but hesitated to make the step of learning the numerical tools involved. This chapter reduces the pain from taking that step. The methods here can be used to analyse most of the models studied in the literature. We discuss how to log-linearize the nonlinear equations without the need for explicit differentiation and how to use the method of undetermined coefficients for models with a vector of endogenous state variables. The methods explained here follow directly from McCallum (1983), King *et al.* (1987) and Campbell (1994), among others.[3] We provide a general solution built on solving matrix quadratic equations (see also Binder and Pesaran, 1995), and provide frequency-domain techniques, building on results in King and Rebelo (1993), to calculate the second-order moments of the model in its Hodrick–Prescott filtered version without resorting to simulations. Since the method is an Euler equation based

---

[1] I am grateful to Michael Binder, Toni Braun, Paul Klein, Jan Magnus, Ramon Marimon, Ellen McGrattan, Víctor Ríos-Rull and Yexiao Xu for helpful comments. I am grateful to Andrew Atkeson for pointing out to me a significant improvement. An earlier version of this chapter was completed while visiting the Institute for Empirical Macroeconomics at the Federal Reserve Bank of Minneapolis: I am grateful for the Institute's hospitality. Any views expressed here are those of the author and not necessarily those of the Federal Reserve Bank of Minneapolis or the Federal Reserve System. This is an updated version of Discussion Paper 101 at the Institute for Empirical Macroeconomics and of CentER DP 9597. Further work was done while visiting the Institute for International Economic Studies in Stockholm: thanks are due to colleagues there, in particular Paul Klein.

[2] Note that the nonlinear model is thus replaced by a linearized approximate model. "Essential" nonlinearities like chaotic systems are unlikely to be handled well by the methods in this paper.

[3] Campbell even touts the approach followed in his paper as "analytical", but note that in his cases as well as in ours, one needs to linearize equations and solve quadratic equations. Campbell presumably attaches the attribute "analytical" to this numerical procedure, since it is rather straightforward indeed and carrying it out by hand is actually feasible in many cases. Otherwise, every numerical calculation anywhere could be called "analytical", since it could in principle be carried out and analysed by hand – it would just take a very long time.

approach rather than an approach based on solving a social planner problem, solving models with externalities or distortionary taxation does not pose additional problems. Since the (nonlinear) Euler equations usually need to be calculated in any case in order to find the steady state, applying the method described in this paper requires little in terms of additional manipulation by hand, given some preprogrammed routines to carry out the matrix calculations of Section 3.4. MATLAB programs to carry out these calculations, given the log-linearized system, are available on my home page[4] and are also discussed in Section 3.9. The method in this chapter therefore allows nonlinear dynamic stochastic models to be solved easily.

Numerical solution methods for solving nonlinear stochastic dynamic models have been studied extensively in the literature; see in particular Kydland and Prescott (1982), the comparison by Taylor and Uhlig (1990) and the methods proposed by various authors in the same issue, Judd (1991), Hansen and Prescott (1995) and Danthine and Donaldson (1995). The literature on solving linear quadratic dynamic stochastic models or linear stochastic difference equations is even larger. The key paper here is Blanchard and Kahn (1980). Furthermore, there are the textbook treatments in Sargent (1987b, Chapters IX and XI), as well as Muth (1961), McGrattan (1994) or Hansen *et al.* (1994), to name a random few. Subject to applicability, all the methods relying on a log-linear approximation to the steady state have in common that they will find the same recursive equilibrium law of motion as the method described in this chapter, since the linear space approximating a nonlinear differentiable function is unique and "immune" to differentiable transformations of the parameter space. But while McGrattan (1994) and Hansen *et al.* (1994) focus on solving models via maximizing a quadratic objective function, and while Blanchard and Kahn (1980) solve linear systems by searching for the stable manifold in the entire system of necessary equations describing the equilibrium relationships, this chapter by contrast, solves directly for the desired recursive equilibrium law of motion. This approach is quite natural. The stability condition is imposed at the point where a certain matrix quadratic equation is solved. It is shown how this matrix quadratic equation can be reduced to a standard eigenvalue problem of another matrix with twice as many dimensions.

Three related contributions are McCallum (1983), which is the key reference for the method of undetermined coefficients, Ceria and Ríos-Rull (1992) and Binder and Pesaran (1996). These contributions also derive the recursive equilibrium law of motion. McCallum (1983) reduces the coefficient-finding problem to a problem solvable with the methods in Blanchard and Kahn (1980), whereas Ceria and Ríos-Rull (1992) reduce the problem to one of solving a matrix quadratic equation as do we, but do not reduce the matrix quadratic equation problem to a standard eigenvalue problem. Binder and Pesaran (1995), finally, may be most closely related in that they reduce the matrix quadratic equation characterizing the solution to an eigenvalue problem as we do. These three contributions, however, for the most part do not distinguish between endogenous variables which have to be part of the state vector, and other endogenous variables. Thus applying these models in somewhat larger systems can result in unnecessarily large and computationally demanding eigenvalue problems in which "bubble solutions" have to be removed in a

---

[4] `http://cwis.kub.nl/~few5/center/STAFF/uhlig/toolkit.dir/toolkit.htm` is the address of the Web site for the programs.

painstaking fashion, or being forced to reduce the system beforehand to make it fit their description.[5] Furthermore, recent related contributions include Christiano and Valdivia (1994), King and Watson (1995; 1997), Sims (1999) and Klein (1998).

But all these technical differences to the existing literature are not in any way essential. It shall be stressed again that the main purpose and merit of this chapter is to make solving nonlinear dynamic stochastic models easy. In fact, this paper describes the entire method as a "cook-book recipe", which should be of great practical use to Ph.D. students and researchers alike. Since the focus here is entirely on the computational aspect of studying these models, some issues are entirely left aside. In particular, the issue of existence or multiplicity of equilibria as well as the reasons for concentrating on stable solutions are not discussed. The methods in this chapter should therefore not be applied blindly, but only in light of, say, McCallum (1983), Stokey *et al.* (1989) and the related literature.

The outline of the paper will be evident from the description of the general procedure in the next section.

3.2 The general procedure

The general procedure for solving and analysing nonlinear dynamic stochastic models consists of the following steps.

1. Find the necessary equations characterizing the equilibrium, i.e. constraints, first-order conditions, etc.; see Section 3.8.1.
2. Pick parameters and find the steady state(s); see Section 3.8.1.
3. Log-linearize the necessary equations characterizing the equilibrium of the system to make the equations approximately linear in the log-deviations from the steady state; see Sections 3.3 and 3.8.1.
4. Solve for the recursive equilibrium law of motion via the method of undetermined coefficients, employing the formulae of Section 3.4.
5. Analyse the solution via impulse-response analysis and second-order-properties, possibly taking account of, say, the Hodrick–Prescott filter. This can be done without having to simulate the model; see Section 3.6.

The next section skips directly to step 3 of the procedure outlined above and describes how to log-linearize nonlinear equations without explicit differentiation. Section 3.8.1 studies Hansen's (1985) benchmark real business cycle model as a prototype example, in which calculation of the Euler equations, the steady state and the log-linearization is carried out to see how this method works. Once a linearized system has been obtained, the methods in Section 3.4 provide the desired recursive equilibrium law of motion.

3.3 Log-linearization and model formulation

Log-linearizing the necessary equations characterizing the equilibrium is a well-known technique. In the context of real business cycle models, log-linearization has been proposed in particular by King *et al.* (1987) and Campbell (1994). Log-linearization also

---

[5] Furthermore, McCallum (1983) uses eigenvalue methods also to solve some other equations in his method, which are solved here by a simple linear equation solution techniques; compare his solution to equation (A.6) in his paper to equation (3.26).

appears frequently in text books such as Obstfeld and Rogoff (1996, pp. 503–505). Nonetheless, the technique often seems to create more headaches than it should. It may be useful for the purposes of this chapter to review how it is done. Section 3.8.1 simplifies the approach of Campbell (1994). Readers who are familiar enough with log-linearization or have their model already in linear form are advised to skip directly to Section 3.4.

The principle is to use a Taylor approximation around the steady state to replace all equations by approximations, which are linear functions in the log-deviations of the variables. Of course, for larger models, it may be convenient to obtain the linearized versions from the original nonlinear equations by numerical differentiation: there is nothing wrong with that, and doing so can speed things up in routine work. But the linearization step itself is not a cumbersome one, as we shall see.

Formally, let X_t be the vector of variables, $\bar{X}$ their steady state and

$$x_t = \log X_t - \log \bar{X}$$

the vector of log-deviations. The vector $100x_t$ tells us by what percentage the variables differ from their steady-state levels in period t. The necessary equations characterizing the equilibrium can be written as

$$1 = f(x_t, x_{t-1}) \tag{3.1}$$

$$1 = E_t[g(x_{t+1}, x_t)] \tag{3.2}$$

where $f(0,0) = 1$ and $g(0,0) = 1$, i.e. the left-hand side of (3.1) and (3.2). Taking first-order approximations around $(x_t, x_{t-1}) = (0,0)$ yields[6]

$$0 \approx f_1 \cdot x_t + f_2 \cdot x_{t-1}$$

$$0 \approx E_t \left[g_1 \cdot x_{t+1} + g_2 \cdot x_t \right]$$

One obtains a linear system in x_t and x_{t-1} in the deterministic equations and x_{t+1} and x_t in the expectational equations. This linear system can be solved with the method of undetermined coefficients, described in Section 3.4.

In the large majority of cases, there is no need to differentiate the functions f and g formally. Instead, the log-linearized system can usually be obtained as follows. Multiply

---

[6] An alternative to approximate (3.2) rewrites it as

$$0 = \log \left\{ E_t \left[\exp\left(\tilde{g}(x_{t+1}, x_t) \right) \right] \right\}$$

where $\tilde{g} = \log g$. Assuming x_t and x_{t+1} to be (approximately) conditionally jointly normally distributed with an (approximately) constant conditional variance-covariance matrix, and assuming that

$$\log g(0,0) \approx \frac{1}{2} \mathrm{Var}_t \left[\tilde{g}_1 \cdot x_{t+1} + \tilde{g}_2 \cdot x_t \right] \tag{3.3}$$

independent of t (rather than $\log g(0,0) = 0$) yields

$$0 \approx \log E_t \left[\exp\left(\tilde{g}(0,0) + \tilde{g}_1 \cdot x_{t+1} + \tilde{g}_2 \cdot x_t \right) \right]$$

$$\approx E_t \left[\tilde{g}_1 \cdot x_{t+1} + \tilde{g}_2 \cdot x_t \right]$$

using $E[e^X] = e^{E[X] + \mathrm{Var}[X]/2}$ for normally distributed variables. The two ways of approximating (3.2) differ essentially only in their choice for $g(0,0)$, since $g_1 = \tilde{g}_1$ if $g(0,0) = 1$.

out everything before log-linearizing. Replace a variable X_t with $X_t = \bar{X}e^{x_t}$, where x_t is a real number close to zero. Likewise, let y_t be a real number close to zero. Take logarithms, where both sides of an equation only involve products, or use the following three building blocks, where a is some constant:

$$\begin{aligned} e^{x_t+ay_t} &\approx 1 + x_t + ay_t \\ x_t y_t &\approx 0 \\ E_t\left[ae^{x_{t+1}}\right] &\approx E_t\left[ax_{t+1}\right] \text{ up to a constant} \end{aligned}$$

For example, these building blocks yield

$$\begin{aligned} e^{x_t} &\approx 1 + x_t \\ aX_t &\approx a\bar{X}x_t \text{ up to a constant} \\ (X_t + a)Y_t &\approx \bar{X}\bar{Y}x_t + (\bar{X} + a)\bar{Y}y_t \text{ up to a constant} \end{aligned}$$

Constants drop out of each equation in the end, since they satisfy steady-state relationships, but they are important in intermediate steps: compare, for example, the two equations above.

Rather than describing the general principles further, it is fruitful to consider a specific example instead. Take Hansen's (1985) benchmark real business cycle model and log-linearize it in the manner described above. Details are described in Section 3.8.1. One obtains

$$-c_t = \lambda_t \tag{3.4}$$

$$n_t = y_t + \lambda_t \tag{3.5}$$

$$\bar{R}r_t = \rho\frac{\bar{Y}}{\bar{K}}(y_t - k_{t-1}) \tag{3.6}$$

$$y_t = z_t + \rho k_{t-1} + (1 - \rho)n_t \tag{3.7}$$

$$\bar{C}c_t + \bar{K}k_t = \bar{Y}y_t + (1 - \delta)\bar{K}k_{t-1} \tag{3.8}$$

$$\lambda_t = E_t[\lambda_{t+1} + r_{t+1}] \tag{3.9}$$

$$z_t = \psi z_{t-1} + \epsilon_t \tag{3.10}$$

Here, c_t denotes the log-deviation of consumption, while $\bar{C}$ denotes the steady-state value of consumption, etc. Those familiar with Hansen's model will readily recognize equation (3.4) to be the first-order condition with respect to consumption C_t (λ_t is the log-linearized Lagrange multiplier on the resource constraint), equation (3.5) to be the first-order condition with respect to labour, equation (3.6) to be the equation defining the return on capital, equation (3.7) to be the log-linearized version of the Cobb–Douglas production function, equation (3.8) to be the resource constraint, equation (3.9) to be the Lucas asset pricing equation, and equation (3.10) to show the evolution of the exogenous total factor productivity. Depending on taste, one obtains a different set of equations which are, however, all equivalent if the calculations are done correctly. For example, one may decide to eliminate λ_t by substituting $-c_t$ right away in all equations.

Note that we use the *dating convention* k_{t-1} for capital created in period $t-1$ and used in production in period t: in general, we shall date all variables with respect to the date at which they are *known*. This contrasts with the notational practice in parts of the literature, but there obviously is no substantial difference: it is just important to keep this in mind when applying the method.

Note that all equations only contain variables dated $t, t-1$ or, possibly, expectations as of time t of variables dated $t+1$. This is the form we need to proceed further. For some models, this may require introducing "dummy" variables to capture further lags of variables or expectations of variables further into the future. So instead of, for example,

$$x_t = E_t[y_{t+2}]$$

write

$$\begin{aligned} x_t &= E_t\left[y_{t+1}^{(1)}\right] \\ y_t^{(1)} &= E_t[y_{t+1}] \end{aligned}$$

and instead of

$$y_t = x_{t-2} + z_t$$

write

$$y_t = x_{t-1}^{(1)} + z_t \tag{3.11}$$

$$x_t^{(1)} = x_{t-1} \tag{3.12}$$

Strictly speaking, the last equation violates our dating convention, but never mind.

Once a linear or linearized system such as (3.4)–(3.10) is obtained, one can proceed to solve for its dynamics.

3.4 Solving recursive stochastic linear systems with the method of undetermined coefficients

This section describes how to find the solution to the recursive equilibrium law of motion in general, using the method of undetermined coefficients. MATLAB programs performing the calculations in this section are available on my home page. The idea is to write all variables as linear functions (the "recursive equilibrium law of motion") of a vector of endogenous variables x_{t-1} and exogenous variables z_t, which are given at date t – that is, which cannot be changed at date t. These variables are often called state variables or predetermined variables. In the real business cycle example of Section 3.8.1, these are at least k_{t-1} and z_t, since they are clearly unchangeable as of date t and, furthermore, show up in the linearized equations system. In principle, any endogenous variable dated $t-1$ or earlier could be considered a state variable. Thus, in Section 3.4.1, we use brute force and simply declare all endogenous variables to be state variables, whereas in Section 3.4.2 we try to be a bit more sensitive and exploit more of the available structure. The latter is typically done in practice; see, for example, Campbell (1994). Both subsections will characterize the solution with a matrix quadratic equation; see also Ceria and Ríos-Rull (1992) and Binder and Pesaran (1995). Section 3.4.3 shows how to solve that equation. For models with just one endogenous state variable, such as the real business cycle model of Section 3.8.1 when analysed with the more structured approach in Section 3.4.2, the matrix quadratic equation is simply a quadratic equation in a real number: for an explicit example and its high-school algebra solution, see Section 3.8.3. This case is contained as a special case of the general solution in Section 3.4.3. In Section 3.5 we discuss our solution method, and compare it in particular to the Blanchard and Kahn (1980) approach.

3.4.1 *With brute force...*

To begin, one may simply use all variables without distinction as a vector of endogenous state variables[7] x_{t-1} of size $m \times 1$ or as a vector of exogenous stochastic processes z_t of size $k \times 1$. It is assumed that the log-linearized equilibrium relationships can be written as

$$0 = E_t[Fx_{t+1} + Gx_t + Hx_{t-1} + Lz_{t+1} + Mz_t] \tag{3.13}$$

$$z_{t+1} = Nz_t + \epsilon_{t+1}, \qquad E_t[\epsilon_{t+1}] = 0 \tag{3.14}$$

where F, G, H, L and M are matrices collecting the coefficients. It is assumed that N has only stable eigenvalues. The real business cycle example (3.4)–(3.10) above can be easily written in this form. For example, the resource constraint (3.8) would be

$$0 = E_t[\bar{C}c_t + \bar{K}k_t - \bar{Y}y_t - (1-\delta)\bar{K}k_{t-1}]$$

since c_t, k_t, k_{t-1} and y_t are already known at date t and hence nothing changes when one takes their expectations given all information up to date t. Note that $F = L = 0$ for this equation. Of course, there are other equations in the real business cycle model, and one of them involves non-zero entries for F and L.

What one is looking for is the recursive equilibrium law of motion

$$x_t = Px_{t-1} + Qz_t \tag{3.15}$$

that is, matrices P and Q such that the equilibrium described by these rules is stable. The solution is characterized in the following theorem; see also Binder and Pesaran (1995). The characterization involves a matrix quadratic equation; see equation (3.16). Section 3.4.3 discusses how it can be solved. For the purpose of that section, let m be the length of the vector x_t, and let $l = n = 0$.

Theorem 3.1 *If there is a recursive equilibrium law of motion solving equations* (3.13), *and* (3.14), *then the following must be true.*

---

[7] To make this work really general, one should actually not only include all the variables dated $t-1$ but also all the variables dated $t-2$ as part of the state vector x_{t-1}. Even more is required if the equations already contain further lags of endogenous variables, see also footnote 8. Usually, however, this is unnecessary.

1. *P satisfies the (matrix) quadratic equation*

$$0 = FP^2 + GP + H \tag{3.16}$$

The equilibrium described by the recursive equilibrium law of motion (3.15) *and* (3.14) *is stable if and only if all eigenvalues of P are less than unity in absolute value.*

2. *Given P, let V denote the matrix*

$$V = N' \otimes F + I_k \otimes (FP + G)$$

Then,

$$VQ = -\mathrm{vec}(LN + M) \tag{3.17}$$

where $\mathrm{vec}(\cdot)$ *denotes columnwise vectorization.*

Obviously, if the matrix V in this theorem is invertible, then multiplication of equation (3.17) by V^{-1} yields the unique solution for Q.

Proof Plugging the recursive equilibrium law of motion (3.15) into equation (3.13) twice and using (3.14) to calculate the expectations yields

$$\begin{aligned} 0 = {} & ((FP + G)P + H)x_{t-1} \\ & + ((FQ + L)N + (FP + G)Q + M)z_t \end{aligned} \tag{3.18}$$

The coefficient matrices of x_{t-1} and z_t need to be zero. Equating the coefficient of x_{t-1} to zero yields equation (3.16) for P. Taking the columnwise vectorization of the coefficient matrices of z_t in this equation and collecting terms in $\mathrm{vec}(Q)$ yields the equation (3.17) for Q. □

3.4.2 ... *or with sensitivity*

We now exploit more of the structure in the linearized model. Analysing the equations of the real business cycle example of Section 3.8.1, one sees that the only endogenous variable dated $t - 1$ which shows up in any of the equations is capital, k_{t-1}. It is thus a reasonable guess to treat k_{t-1} as the only endogenous state variable together with the exogenous state variable z_t. This principle is general: in the vast majority of cases, this is how one can identify the vector of state variables.[8] In practice, one often sees researchers exploiting some of the equilibrium equations to get rid of some variables, and have only a few variables remaining. For the real business cycle example of Section 3.8.1, it is actually possible to reduce everything to a single equation for the endogenous variables, containing only k_{t+1}, k_t, and k_{t-1}. Often, one sees reductions to a system involving two equations in two endogenous variables such as λ_t and k_{t-1} (see e.g. Campbell, 1994; and

---

[8] There are exceptions. Variables chosen at a date earlier than $t - 1$ may need to be included: this can be treated as in equations (3.11), (3.12). One may also need to add additional variables like e.g. c_{t-1} or k_{t-2} as state variables, even though they don't show up in the equations with these dates, when the model exhibits sun spot dynamics. This can be done in the same manner, but one needs to be careful with interpreting the results. Appendix 3.8 and in particular 3.8.8 elaborates on this in more careful detail.

Section 3.8.2 below), presumably because this allows thinking in terms of a state-space diagram (see e.g. Blanchard and Fisher, 1989, Chapter 2). However, there is no reason to bother with "eliminating" variables by hand, using some of the equations: since this is all just simple linear algebra applied to a system of equations, it is far easier to leave all the equations in, and leave it to the formulae to sort it all out. That is what is done below.

We thus make the following assumptions.[9] There is an endogenous state vector x_t, of size $m \times 1$, a list of other endogenous variables ("jump variables") y_t, of size $n \times 1$, and a list of exogenous stochastic processes z_t, of size $k \times 1$. The equilibrium relationships between these variables are

$$0 = Ax_t + Bx_{t-1} + Cy_t + Dz_t \tag{3.19}$$

$$0 = E_t[Fx_{t+1} + Gx_t + Hx_{t-1} + Jy_{t+1} + Ky_t + Lz_{t+1} + Mz_t] \tag{3.20}$$

$$z_{t+1} = Nz_t + \epsilon_{t+1}; \quad E_t[\epsilon_{t+1}] = 0 \tag{3.21}$$

where it is assumed that C is of size $l \times n$, $l \geq n$ and[10] of rank n, that F is of size $(m + n - l) \times n$, and that N has only stable eigenvalues. Note that one could have written all equations (3.19) in the form of equation (3.20) with the corresponding entries in the matrices F, J and L set to zero. Essentially, that is what is done in Section 3.4.1. Instead, the point here is somehow to exploit the structure inherent in equations of the form (3.19), which do not involve taking expectations.

What one is looking for is the recursive equilibrium law of motion

$$x_t = Px_{t-1} + Qz_t \tag{3.22}$$

$$y_t = Rx_{t-1} + Sz_t \tag{3.23}$$

that is, matrices P, Q, R, and S such that the equilibrium described by these rules is stable. The solution is characterized in the next theorem. To calculate the solution, one needs to solve a matrix quadratic equation: how this is done is explained in Section 3.4.3.

The important special case $l = n$ is treated in Corrollary 3.3. The special case $l = n = 0$ was the topic of Section 3.4.1.

Theorem 3.2 *If there is a recursive equilibrium law of motion solving equations* (3.19), (3.20), *and* (3.21), *then the coefficient matrices can be found as follows. Let C^+ be the pseudo-inverse*[11] *of C. Let C^0 be an $(l - n) \times l$ matrix whose rows form a basis for the null space*[12] *of C'.*

---

[9] Note that the notation differs from the notation in Section 3.3.

[10] The case $l < n$ can be treated as well: the easiest approach is simply to "redeclare" some other endogenous variables to be state variables instead – that is, to raise m and thus lower n – until $l = n$.

[11] The pseudo-inverse of the matrix C is the $n \times l$ matrix C^+ satisfying $C^+CC^+ = C^+$ and $CC^+C = C$. Since it is assumed that rank$(C) \geq n$, one obtains $C^+ = (C'C)^{-1}C'$; see Strang (1980, p. 138). The MATLAB command to compute the pseudo-inverse is pinv(C).

[12] C^0 can be found via the singular value decomposition of C'; see Strang (1980, p. 142). The MATLAB command for computing C^0 is (null(C'))$'$.

1. *P satisfies the (matrix) quadratic equations*

$$0 = C^0 AP + C^0 B \tag{3.24}$$

$$0 = (F - JC^+ A)P^2 - (JC^+ B - G + KC^+ A)P - KC^+ B + H \tag{3.25}$$

The equilibrium described by the recursive equilibrium law of motion (3.22), (3.23) *and by* (3.21) *is stable if and only if all eigenvalues of P are less than unity in absolute value.*

2. *R is given by*

$$R = -C^+(AP + B)$$

3. *Given P and R, let V be the matrix*

$$V = \begin{bmatrix} I_k \otimes A, & I_k \otimes C \\ N' \otimes F + I_k \otimes (FP + JR + G) & N' \otimes J + I_k \otimes K \end{bmatrix}$$

where I_k is the identity matrix of size $k \times k$. Then

$$V \begin{bmatrix} \text{vec}(Q) \\ \text{vec}(S) \end{bmatrix} = - \begin{bmatrix} \text{vec}(D) \\ \text{vec}(LN + M) \end{bmatrix} \tag{3.26}$$

where vec(·) *denotes columnwise vectorization.*

Obviously, if V in this theorem is invertible, then multiplication of equation (3.26) with V^{-1} yields the unique solution for Q.

Proof Plug the recursive equilibrium law of motion into equation (3.19). This yields

$$(AP + CR + B)x_{t-1} + (AQ + CS + D)z_t = 0 \tag{3.27}$$

which has to hold for arbitrary x_{t-1} and z_t. Thus, the coefficient matrices on x_{t-1} and z_t in (3.27) are zero. Plugging the recursive equilibrium law of motion into equation (3.20) twice and using (3.21) yields

$$\begin{aligned} 0 =& ((FP + JR + G)P + KR + H)x_{t-1} \\ &+ ((FQ + JS + L)N + (FP + JR + G)Q + KS + M)z_t \end{aligned} \tag{3.28}$$

Again, the coefficient matrices on x_{t-1} and z_t need to be zero. Taking the columnwise vectorization of the coefficient matrices of z_t in equations (3.27) and (3.28) and collecting terms in vec(Q) and vec(S) yields the formula for Q and S. To find P and thus R, rewrite the coefficient matrix on x_{t-1} in equation (3.27) as

$$\begin{aligned} R &= -C^+(AP + B) \\ 0 &= C^0 AP + C^0 \end{aligned} \tag{3.29}$$

noting that the matrix $[(C^+)', (C^0)']$ is non-singular and that $C^0 C = 0$; see Strang (1980, p. 88). Use (3.29) to replace R in the coefficient matrix on x_{t-1} in (3.28), yielding (3.25). Note finally that the stability of the equilibrium is determined by the stability of P, since N has stable roots by assumption. □

Corollary 3.3 *Suppose that $l = n$, that is, that there are as many expectational equations as there are endogenous state variables. If there is a recursive equilibrium law of motion solving equations* (3.19), (3.20), *and* (3.21), *then their coefficient matrices can be found as follows.*

1. *P satisfies the (matrix) quadratic equation*

$$(F - JC^{-1}A)P^2 - (JC^{-1}B - G + KC^{-1}A)P - KC^{-1}B + H = 0 \qquad (3.30)$$

The equilibrium described by the recursive equilibrium law of motion (3.22), (3.23) *and by* (3.21) *is stable if and only if all eigenvalues of P are less than unity in absolute value.*

2. *R is given by*

$$R = -C^{-1}(AP + B)$$

3. *Q satisfies*

$$\begin{aligned}(N' \otimes (F - JC^{-1}A) + I_k \otimes (JR + FP + G - KC^{-1}A))\text{vec}(Q) \qquad (3.31)\\ = \text{vec}((JC^{-1}D - L)N + KC^{-1}D - M)\end{aligned}$$

where I_k is the identity matrix of size $k \times k$, provided the matrix which needs to be inverted in this formula is indeed invertible.

4. *S is given by*

$$S = -C^{-1}(AQ + D)$$

Proof This corollary can be obtained directly by inspecting the formulae of Theorem 3.2 above for the special case $l = n$. In particular, C^+ is just the inverse of C. Alternatively, a direct proof can be obtained directly by following the same proof strategy as above. □

The formulae in these theorems become even simpler if $m = 1$ or $k = 1$. If $m = 1$, there is just one endogenous state variable and the matrix quadratic equation above becomes a quadratic equation in the real number P, which can be solved using high-school algebra: this is the case for the real business cycle model and thus the case which Campbell (1994) analyses. If $k = 1$, there is just one exogenous state variable, in which case the Kronecker product $\otimes$ in the formulae above becomes multiplication, and in which case $\text{vec}(Q) = Q$ and $\text{vec}(S) = S$, since Q and S are already vectors rather than matrices.

3.4.3 *Solving the matrix quadratic equation*

To solve the matrix quadratic equations (3.16) or (3.24), (3.25) for P, write them as

$$\Psi P^2 - \Gamma P - \Theta = 0 \qquad (3.32)$$

For equations (3.24) and (3.25), define

$$\Psi = \begin{bmatrix} 0_{l-n,m} \\ F - JC^{+}A \end{bmatrix}$$

$$\Gamma = \begin{bmatrix} C^{0}A \\ JC^{+}B - G + KC^{+}A \end{bmatrix}$$

$$\Theta = \begin{bmatrix} C^{0}B \\ KC^{+}B - H \end{bmatrix}$$

where $0_{l-n,m}$ is an $(l-n) \times m$ matrix with only zero entries. In the special case $l = n$, the formulae for Ψ, Γ and Θ become slightly simpler:

$$\Psi = F - JC^{-1}A$$

$$\Gamma = JC^{-1}B - G + KC^{-1}A$$

$$\Theta = KC^{-1}B - H$$

For equation (3.16), simply use $\Psi = F$, $\Gamma = -G$ and $\Theta = -H$.

Equation (3.32) can now be solved in two ways. In Theorem 3.4, we will solve it by turning it into a generalized eigenvalue and eigenvector problem,[13] for which most mathematical packages have preprogrammed routines.[14] In Theorem 3.5, we will solve it, using the QZ method.

Recall that a generalized eigenvalue λ and eigenvector s of a matrix Ξ with respect to a matrix Δ are defined to satisfy

$$\lambda \Delta s = \Xi s \tag{3.33}$$

A standard eigenvalue problem is obtained if Δ is the identity matrix. More generally, the generalized eigenvector problem can be reduced to a standard one, if Δ is invertible, by calculating standard eigenvalues and eigenvectors for $\Delta^{-1}\Xi$ instead.

Theorem 3.4 *To solve the quadratic matrix equation*

$$\Psi P^{2} - \Gamma P - \Theta = 0, \tag{3.34}$$

for the $m \times m$ matrix P, given $m \times m$ matrices Γ and Θ, define the $2m \times 2m$ matrices Ξ and Δ by

$$\Xi = \begin{bmatrix} \Gamma & \Theta \\ I_m & 0_{m,m} \end{bmatrix}$$

---

[13] An earlier version of this chapter proposed to study an altered version of these equations by postmultiplying equation (3.24) with P. This altered equation, together with (3.25), can then often be reduced to a standard rather than a generalized eigenvalue problem, but has the drawback of introducing spurious zero roots. The version presented here does not involve this alteration, and thus does not introduce spurious zero roots. This update is due to Andy Atkeson (1997), and I am very grateful to him for pointing it out to me. Any errors here are mine, of course.

[14] The MATLAB command for finding the generalized eigenvalues and eigenvectors is eig(Ξ,Δ).

and

$$\Delta = \begin{bmatrix} \Psi & 0_{m,m} \\ 0_{m,m} & I_m \end{bmatrix}$$

where I_m is the identity matrix of size m, and where $0_{m,m}$ is the $m \times m$ matrix with only zero entries.

1. *If s is a generalized eigenvector and λ the corresponding generalized eigenvalue of Ξ with respect to Δ, then s can be written as $s' = [\lambda x', x']$ for some $x \in \mathbb{R}^m$.*
2. *If there are m generalized eigenvalues $\lambda, \dots, \lambda_m$ together with generalized eigenvectors $s_1, \dots, s_m$ of Ξ with respect to Δ, written as $s_i' = [\lambda_i x_i', x_i']$ for some $x_i \in \mathbb{R}^m$, and if $(x_1, \dots, x_m)$ is linearly independent, then*

   $$P = \Omega \Lambda \Omega^{-1}$$

   is a solution to the matrix quadratic equation (3.34), where $\Omega = [x_1, \dots, x_m]$ and $\Lambda = \text{diag}(\lambda, \dots, \lambda_m)$. The solution P is stable if $\mid \lambda_i \mid < 1$ for all $i = 1, \dots, m$. Conversely, any diagonalizable solution P to (3.34) can be written in this way.
3. *If $m = 1$, then the solutions P to equation (3.34) are given by*

   $$P_{1,2} = \frac{1}{2\Psi}(\Gamma \pm \sqrt{\Gamma^2 + 4\Psi\Theta})$$

   if $\Psi \neq 0$, and by

   $$P = -\frac{\Theta}{\Gamma}$$

   if $\Psi = 0$ and $\Gamma \neq 0$.

Proof First, examine the last m rows of equation (3.33) to see that any eigenvector s for some eigenvalue λ of the matrix Ξ with respect to Δ can indeed be written as

$$s = \begin{bmatrix} \lambda x \\ x \end{bmatrix}$$

for some $x \in \mathbb{R}^m$ because of the special form of Ξ and Δ. Examining the first m rows of equation (3.33) then shows that

$$\lambda^2 \Psi x - \lambda \Gamma x - \Theta x = 0 \tag{3.35}$$

It follows that

$$\Psi \Omega \Lambda^2 - \Gamma \Omega \Lambda - \Theta \Omega = 0$$

and hence

$$\Psi P^2 - \Gamma P - \Theta = 0$$

as claimed, after multiplying with Ω^{-1} from the right.

Reversing the steps shows that any diagonalizable solution P to (3.34) can be written in this way. □

One further generalization step can be made by using the QZ decomposition instead of a (generalized) diagonalization; see, in particular, Sims (1999). In particular, the QZ decomposition can deal with repeated eigenvalues easily, and is thus the preferred method. The next theorem shows how it works, but before stating it, we need an additional piece of notation. For any $2m \times 2m$ matrix X, say, write its partition as

$$X = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix}$$

where X_{ij} denotes a submatrix of size $m \times m$. Likewise, any vector $v \in \mathbb{R}^{2m}$ shall be partitioned in this way.

Theorem 3.5 *To solve the quadratic matrix equation*

$$\Psi P^2 - \Gamma P - \Theta = 0 \tag{3.36}$$

for the $m \times m$ matrix P, given $m \times m$ matrices Γ and Θ, define the $2m \times 2m$ matrices Ξ and Δ by

$$\Xi = \begin{bmatrix} \Gamma & \Theta \\ I_m & 0_{m,m} \end{bmatrix}$$

and

$$\Delta = \begin{bmatrix} \Psi & 0_{m,m} \\ 0_{m,m} & I_m \end{bmatrix}$$

where I_m is the identity matrix of size m, and where $0_{m,m}$ is the $m \times m$ matrix with only zero entries.

Find the QZ decomposition (or generalized Schur decomposition) of Δ and Ξ, that is, find unitary[15] *matrices*[16] *Y and Z as well as upper triangular matrices Σ and Φ such that*[17]

$$Y'\Sigma Z = \Delta$$
$$Y'\Phi Z = \Xi$$

Assume that Z_{21} and Y_{21} are invertible.

1. *The matrix*

$$P = -Z_{21}^{-1} Z_{22}$$

solves the matrix quadratic equation (3.36).

---

[15] The matrix Y is unitary if and only if $Y'Y = I_{2m}$, where Y' denotes the complex conjugate transpose of Y.

[16] Matrix Y is usually denoted by Q, but we have used that symbol for another matrix already.

[17] Such a QZ decomposition always exists, although it may not be unique.

2. *Let ϕ_{ij} be the (i, j)th element of Φ and, likewise, let σ_{ij} be the (i, j)th element of Σ. Suppose that the QZ decomposition has been chosen so that the ratios $|\phi_{ii}/\sigma_{ii}|$ of the diagonal elements are in ascending order.*[18] *Furthermore, assume that $|\phi_{mm}/\sigma_{mm}| < 1$. Then, P is stable in the sense that $P^n x \to 0$ as $n \to \infty$ for any $x \in \mathbb{R}^m$.*

Proof We first show the first claim, that P indeed solves the matrix quadratic equation (3.36). We need to show that

$$\left(\Psi P^2 - \Gamma P - \Theta\right) x = 0$$

for any vector $x \in \mathbb{R}^m$. Equivalently, define $v(x) \in \mathbb{R}^{2m}$ via

$$v(x) = \begin{bmatrix} Px \\ x \end{bmatrix}$$

We need to show that

$$\Delta v(Px) = \Xi Px \tag{3.37}$$

or

$$\begin{bmatrix} \Psi P^2 x \\ Px \end{bmatrix} = \begin{bmatrix} \Gamma Px + \Theta x \\ Px \end{bmatrix} \tag{3.38}$$

for any $x \in \mathbb{R}^m$. Equation (3.38) shows that the claim (3.37) is already trivially true for the last m rows.

Define $w(x) = Zv(x)$ and note that

$$w(x) = \begin{bmatrix} Z_{11}Px + Z_{12}x \\ 0_{m,1} \end{bmatrix}$$

Equation (3.37) can therefore be written as

$$\begin{bmatrix} Y_{11}\Sigma_{11}\left(Z_{11}P^2x + Z_{12}Px\right) \\ Y_{21}\Sigma_{11}\left(Z_{11}P^2x + Z_{12}Px\right) \end{bmatrix} = \begin{bmatrix} Y_{11}\Phi_{11}\left(Z_{11}Px + Z_{12}x\right) \\ Y_{21}\Phi_{11}\left(Z_{11}Px + Z_{12}x\right) \end{bmatrix} \tag{3.39}$$

which needs to be shown. Comparing the last m rows of equations (3.38) and (3.39) and using the invertibility of Y_{21}, we see that

$$\Sigma_{11}\left(Z_{11}P^2x + Z_{12}Px\right) = \Phi_{11}\left(Z_{11}Px + Z_{12}x\right)$$

Substituting this into the first m rows of (3.39) and comparing it to equation (3.38), we

---

[18] This can always be achieved, starting from a given QZ decomposition, by "swapping" axes in appropriate pairs. The MATLAB command to compute one QZ decomposition for A and B is qz(A, B). An algorithm for MATLAB for performing the swapping has been written by C. Sims to accompany his paper, Sims (1996). The reader is referred to these programs or to the literature on linear algebra for further discussion.

finally see that

$$\begin{aligned}\Psi P^2 x &= Y_{11}\Sigma_{11}\left(Z_{11}P^2 x + Z_{12}Px\right)\\ &= Y_{11}\Sigma_{11}\left(Z_{11}Px + Z_{12}x\right)\\ &= \Gamma Px + \Theta x\end{aligned}$$

as claimed.

We now show the second claim about stability. For any $x \in \mathbb{R}^m$ as above, let

$$v(x) = \begin{bmatrix} Px \\ x \end{bmatrix}$$

and

$$w(x) = Zv(x) = \begin{bmatrix} Z_{11}Px + Z_{12}x \\ 0_{m,1} \end{bmatrix}$$

Fix $x \in \mathbb{R}^m$. Note that $P^n x \longrightarrow 0$ if and only if $v(P^n x) \longrightarrow 0$, which is true if and only if $w(P^n x) \longrightarrow 0$. Let $n \geq 1$. Note that $\Delta v(P^n x) = \Xi v(P^{n-1}x)$ or, equivalently, $\Sigma w(P^n x) = \Phi w(P^n - 1x)$. Write out the latter expression explicitly. Keep in mind that $(w(P^{n-1}x))_2 = (w(P^n x))_2 = 0$. The first m rows deliver

$$(w(P^n x))_1 = \Sigma_{11}^{-1}\Phi_{11}(w(P^{n-1}x))_1$$

where Σ_{11} must be invertible and where $\left(\Sigma_{11}^{-1}\Phi_{11}\right)^n \longrightarrow 0$ due to our assumption about the diagonal elements. Hence,

$$(w(P^n x))_1 \longrightarrow 0 \quad \text{as } n \longrightarrow \infty$$

completing the proof. □

3.5 Discussion

Theorems 3.4 links the approach used here to Blanchard and Kahn (1980), which is the key reference for solving linear difference equations. A more detailed discussion, in particular with respect to the differences between saddle-point stable models *vis-à-vis* models with indeterminacies can be found in Section 3.8.

Consider solving the second-order difference equation

$$\Psi x_{t+1} - \Gamma x_t - \Theta x_{t-1} = 0 \tag{3.40}$$

The approach in Blanchard and Kahn (1980) amounts to finding the stable roots of Ξ by instead analysing the dynamics of the "stacked" system $s_t' = [x_t', x_{t-1}']$,

$$\Delta s_{t+1} = \Xi s_t$$

that is, by reducing (3.40) to a first-order difference equation. The approach here solves for the matrix P in the recursive equilibrium law of motion $x_{t+1} = Px_t$. Theorem 3.4 above states that both approaches amount to the same problem. The advantage of the method here is that it is easily applied to the entire system (3.19)–(3.21), reducing it eventually to (3.40), while finding the stable roots in the entire system given by these equations and at the same time taking care of the expectation operators, using the Blanchard and Kahn (1980) procedure is often perceived as complicated. Fundamentally, there is no difference.

To apply Theorem 3.4, one needs to select m out of $2m$ possible eigenvalues. Note that P has only non-zero eigenvalues if the state space was chosen to be of minimal size: attention can be restricted to the roots $\mid \lambda_i \mid > 0$ in that case. In general, there may be quite a bit of choice left. In practice, however, there will often be exactly m stable eigenvalues remaining so that the stable solution is unique.[19] For a one-dimensional vector of endogenous state variables, this condition is called saddle-point stability. The literature on solving linear rational expectations equilibria typically assumes this condition to hold or shows it to hold in social planning problems under reasonable conditions; see Blanchard and Kahn (1980), Kollintzas (1985) and Hansen *et al.* (1994). If there are fewer stable eigenvalues than endogenous state variables, the equilibrium might be inherently unstable. The method above then still permits calculation of an equilibrium which satisfies the nonlinear equilibrium conditions at least locally. In particular, in models involving more than one agent or sector or country, one may find as many unit roots as there are more agents (sectors, countries) than one since shocks may affect the relative wealth (capital) of any two agents (sectors, countries) and thus may result in permanent changes in their consumption paths (or capital stocks): in these cases, the method above allowing for unit roots still gives useful results, which obviously should then be used with some care. These unit roots typically already show up as an undetermined steady state: any of the possible steady states can then serve as a starting point for the dynamic calculation, keeping in mind that a simulation based on the dynamics calculated here will eventually wander away too far to be numerically useful. If there are more stable eigenvalues than endogenous state variables, enlarging the number of endogenous state variables by including further lagged values might help. Nonetheless, the presence of an excess of stable roots may then point to the existence of sunspots or endogenous fluctuations; see, example Farmer and Guo (1994). These matters are discussed in greater detail in Section 3.8.

If not all eigenvalues of Ξ are distinct, P in turn might have repeated eigenvalues. Since the eigenspace for a repeated eigenvalue is (usually) multi-dimensional, there will be infinitely many choices for the eigenvectors and hence infinitely many choices for P in that case. Note, for example, that for any given λ and any three real numbers a, b, c satisfying $a^2 + bc = \lambda^2$, all matrices

$$P = \begin{bmatrix} a & b \\ c & -a \end{bmatrix}$$

---

[19] Another approach to select a unique solution is given in McCallum (1983), who suggests using those roots that can be obtained continuously from the zero roots of the equation $\Psi P^2 - \Gamma P - \alpha\Theta$ for $\alpha = 0$, as α changes from 0 to 1. However, not only is following these roots as functions of α computationally very demanding, it is also the case that uniqueness is lost once two or more such paths cross each other. If these paths do not cross in a particular application, and if additionally all roots for all α are positive real numbers, say, then the McCallum proposal simply amounts to using the roots of minimal value. The MATLAB programs supplied by the author use the roots of minimal absolute value, subject to eliminating spurious zero roots, and try to use complex roots in conjugate pairs, as described below.

solve

$$P^2 - \begin{bmatrix} \lambda^2 & 0 \\ 0 & \lambda^2 \end{bmatrix} = 0$$

These cases are rare in practice, since Ξ is diagonalizable with distinct eigenvalues generically in the coefficients of the system (3.19)–(3.21). In any case, the QZ method can deal with them without difficulty; see Theorem 3.5.

More disconcerting is the possibility that some of the roots may be complex rather than real. Consider, for example, $\Psi = I_2$, $\Gamma = -I_2$ and

$$\Theta = \begin{bmatrix} 0.23 & 0.64 \\ -0.64 & 0.23 \end{bmatrix}$$

Using the theorem above, one obtains exactly two stable roots, which happen to be complex, $\lambda_{1,2} = 0.3 \pm 0.4\mathrm{i} = 0.5e^{\pm \alpha \mathrm{i}}$, where $\alpha \approx 0.9273$. Their associated eigenvectors are complex, too. Calculating P results in a matrix with only real entries, however, given by

$$P = \begin{bmatrix} 0.3 & 0.4 \\ -0.4 & 0.3 \end{bmatrix} = 0.5 \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}$$

Since Ξ is a real-valued matrix, complex eigenvalues only arise in complex conjugate pairs. When using both roots of a complex conjugate pair to calculate Λ and thus P, the resulting solution should be a real-valued matrix. In order to do this, one may have to enlarge the state space of endogenous state variables to be at least two-dimensional; see again Farmer and Guo (1994) for an example. The complex roots then give rise to endogenous damped cycles of frequency α. Again, see Section 3.8 (in particular, Section 3.8.8) for further discussion.

3.6 Interpreting the results

The results obtained, that is, the recursive equilibrium law of motion

$$\begin{aligned} x_t &= Px_{t-1} + Qz_t \\ y_t &= Rx_{t-1} + Sz_t \\ z_t &= Nz_{t-1} + \epsilon_t \end{aligned}$$

can be used to examine model implications. Since x_t, y_t and z_t are log-deviations, the entries in P, Q, R, S and N can be understood as elasticities and interpreted accordingly; see, for example, Campbell (1994).

Impulse responses to a particular shock ϵ_1 can be calculated by setting $x_0 = 0$, $y_0 = 0$ and $z_0 = 0$, as well as $\epsilon_t = 0$ for $t \geq 2$, and recursively calculating z_t and then x_t and y_t, given x_{t-1}, y_{t-1}, z_{t-1} and ϵ_t for $t = 1, \ldots, T$ with the recursive equilibrium law of motion and the law of motion for z_t.

To find the second moment properties of the model such as variances and autocorrelations of certain variables, as well as the small-sample properties of their estimators, simulation methods are often used. Before calculating these moments, the Hodrick–Prescott filter is typically applied. This section demonstrates a frequency-domain technique to obtain these moments (albeit without the small-sample properties of their estimators) without the need for any simulations.[20] Obviously, the methods here do not deliver properties of the small-sample distribution, which may be necessary for testing.

The matrix-valued spectral density for $[x_t', z_t']'$ is given by

$$f(\omega) = \frac{1}{2\pi}\begin{bmatrix}(I_m - Pe^{-i\omega})^{-1}Q \\ I_k\end{bmatrix}(I_k - Ne^{-i\omega})^{-1}\Sigma$$
$$\times (I_k - N'e^{i\omega})^{-1}\left[Q'(I_m - P'e^{i\omega})^{-1}, I_k\right]$$

where I_k and I_m are the identity matrices of dimension k and m; see formula (10.4.43) in Hamilton (1994). Two ways to calculate the matrix-valued spectral density for the entire vector of variables $s_t = [x_t', y_t', z_t']'$ are

$$g(\omega) = \begin{bmatrix} I_m & 0_{m,k} \\ Re^{-i\omega} & S \\ 0_{k,m} & I_k \end{bmatrix} f(\omega) \begin{bmatrix} I_m & R'e^{i\omega} & 0_{m,k} \\ 0_{k,m} & S' & I_k \end{bmatrix} = Wf(\omega)W'$$

where

$$W = \begin{bmatrix} I_m & 0_{m,k} \\ RP^+ & S - RP^+Q \\ 0_{k,m} & I_k \end{bmatrix}$$

where P^+ is the pseudo-inverse of P and where the last equality exploits $s_t = W[x_t', z_t']'$, replacing x_{t-1} with $P^+x_t - P^+Qz_t$ in the recursive equilibrium law of motion for y_t. The Hodrick–Prescott filter aims to remove a smooth trend τ_t from some given data s_t by solving

$$\min_{\tau_t}\sum_{t=1}^{T}\left((s_t - \tau_t)^2 + \lambda\left((\tau_{t+1} - \tau_t) - (\tau_t - \tau_{t-1})\right)^2\right)$$

The solution is a linear lag polynomial $r_t = s_t - \tau_t = h(L)s_t$ which has the transfer function

$$\tilde{h}(\omega) = \frac{4\lambda(1 - \cos(\omega))^2}{1 + 4\lambda(1 - \cos(\omega))^2}$$

(see, King and Rebelo, 1993). Thus, the matrix spectral density of the Hodrick–Prescott filtered vector is simply

$$g_{\mathrm{HP}}(\omega) = \tilde{h}^2(\omega)g(\omega)$$

---

[20] Some of these methods were originally contained in an early version of Uhlig and Xu (1996), but were eventually deleted from that paper.

from which one can obtain the autocorrelations of r_t in the time domain via an inverse Fourier transformation,

$$\int_{-\pi}^{\pi} g_{\mathrm{HP}}(\omega) e^{i\omega k} d\omega = E[r_t r_{t-k}']$$

see formula (10.4.4) in Hamilton (1994). Inverse Fourier transformations are part of many numerical packages.

3.7 Conclusions

We have provided a toolkit to analyse nonlinear dynamic stochastic models easily. The main contribution of this chapter is to simplify and unify existing approaches, showing how to log-linearize the necessary equations characterizing the equilibrium without explicit differentiation, to provide a general solution to a linearized system using the method of undetermined coefficients, allowing in particular for a vector of endogenous states, and to provide simulation-free frequency-domain based method to calculate the model implications in its Hodrick–Prescott filtered version. These methods are easy to use if a numerical package such as MATLAB is available. This chapter should therefore be useful for anybody interested in analysing nonlinear stochastic dynamic models.

3.8 Appendix: Undetermined Coefficients versus Blanchard–Kahn: two examples

The purpose of this appendix is to go through the log-linearization exercise for one model in detail, to solve the model "by hand" and to relate the method of undetermined coefficients to the Blanchard–Kahn approach for the case of saddle-point stability and for the case of indeterminacy.

3.8.1 *Hansen's real business cycle model: saddle-point stability*

The following model is the benchmark real business cycle model due to Hansen (1985) and explained there in detail. Here, the mathematical description shall suffice. The main point of this example, as far as we are concerned, is to explain how to perform the first three steps of the general procedure.

The social planner solves the problem of the representative agent

$$\max E \sum_{t=1}^{\infty} \beta^t (\log C_t - AN_t)$$

such that

$$\begin{aligned} C_t + K_t &= Y_t + (1-\delta)K_{t-1} \qquad (3.41) \\ Y_t &= Z_t K_{t-1}^{\rho} N_t^{1-\rho} \\ \log Z_t &= (1-\psi)\log \bar{Z} + \psi \log Z_{t-1} + \epsilon_t \end{aligned}$$

where the ϵ_t are independently and identically normally distributed with zero mean and variance σ^2, C_t is consumption, N_t is labour, I_t is investment, Y_t is production, K_t is capital, Z_t is the total factor productivity, and A, β, δ, ρ, $\bar{Z}$, ψ and σ^2 are parameters.

Collapse the first two equations into one, and let Λ_t be its Lagrange multiplier. The first-order conditions are

$$\frac{1}{C_t} = \Lambda_t$$

$$A = \Lambda_t (1-\rho)\frac{Y_t}{N_t}$$

$$\Lambda_t = \beta E_t \left[\Lambda_{t+1} R_{t+1}\right] \tag{3.42}$$

where

$$R_t = \rho \frac{Y_t}{K_{t-1}} + 1 - \delta \tag{3.43}$$

Note that we wrote the Lagrange multiplier as Λ_t rather than λ_t for the sake of consistency with our notational convention to use lower-case letters for log-deviations. Equation (3.42) is the Lucas (1978) asset pricing equation which typically arises in these models.

In contrast to some of the real business cycle literature and to avoid confusion in the application of the method in Section 3.4, it is very useful to stick to the following *dating convention*. A new date starts with the arrival of new information. If a variable is chosen and/or (eventually) known at date t, it will be indexed with t. Use only variables dated t and $t-1$ in deterministic equations and variables dated $t+1$, t and $t-1$ in equations involving expectations $E_t[\cdot]$.

The steady state for the real business cycle model above is obtained by dropping the time subscripts and stochastic shocks in the equations above, characterizing the equilibrium. Formally, this amounts to finding steady-state values such that $f(0,0) = 1$ and $g(0,0) = 1$ in the notation of Section 3.3.[21] For example, equations (3.42) and (3.43) result in

$$1 = \beta \bar{R}$$

$$\bar{R} = \rho \frac{\bar{Y}}{\bar{K}} + 1 - \delta$$

where bars over variables denote steady-state values. One needs to decide what one wants to solve for. If one fixes β and δ, these two equations will imply values for $\bar{R}$ and $\bar{Y}/\bar{K}$. Conversely, one can fix $\bar{R}$ and $\bar{Y}/\bar{K}$ and then these two equations yield values for β and δ. The latter procedure maps observable characteristics of the economy into "deep parameters", and is the essence of calibration; see Kydland and Prescott (1991).

Introduce lower-case letters to denote log-deviations: write

$$C_t = \bar{C}e^{c_t}$$

---

[21] Alternatively, find the steady state so that (3.3) is satisfied. This is, however, rarely done.

for example. The resource constraint (3.41) then reads

$$\bar{C}e^{c_t} + \bar{K}e^{k_t} = \bar{Y}e^{y_t} + (1-\delta)\bar{K}e^{k_{t-1}}$$

This can be written approximately as

$$\bar{C}(1+c_t) + \bar{K}(1+k_t) = \bar{Y}(1+y_t) + (1-\delta)\bar{K}(1+k_{t-1})$$

Since $\bar{C} + \delta\bar{K} = \bar{Y}$ due to the definition of the steady state, the constant terms drop out[22] and one obtains

$$\bar{C}c_t + \bar{K}k_t = \bar{Y}y_t + (1-\delta)\bar{K}k_{t-1} \tag{3.44}$$

The resource constraint is now stated in terms of percentage deviations: the steady-state levels in this equation rescale the percentage deviations to make them comparable. Note that no explicit differentiation is required to obtain the log-linearized version of the resource constraint: log-linearization is obtained just by using the building blocks described in section 3.3.

Similarly log-linearizating the other equations, one obtains:

#	Equation	Log-linearized version
(i)	$\frac{1}{C_t} = \Lambda_t$	$-c_t = \lambda_t$
(ii)	$A = \Lambda_t(1-\rho)\frac{Y_t}{N_t}$	$n_t = y_t + \lambda_t$
(iii)	$R_t = \rho\frac{Y_t}{K_{t-1}} + 1 - \delta$	$\bar{R}r_t = \rho\frac{\bar{Y}}{\bar{K}}(y_t - k_{t-1})$
(iv)	$Y_t = \bar{Z}e^{z_t}K_{t-1}^{\rho}N_t^{1-\rho}$	$y_t = z_t + \rho k_{t-1} + (1-\rho)n_t$
(v)	$C_t + K_t = Y_t + (1-\delta)K_{t-1}$	$\bar{C}c_t + \bar{K}k_t = \bar{Y}y_t + (1-\delta)\bar{K}k_{t-1}$
(vi)	$\Lambda_t = \beta E_t[\Lambda_{t+1}R_{t+1}]$	$\lambda_t = E_t[\lambda_{t+1} + r_{t+1}]$
(vii)	$\log Z_t = (1-\psi)\log\bar{Z} + \psi\log Z_{t-1} + E_t$	$Z_t = \psi Z_{t-1} + E_t$

The equations are written so that only variables dated t, $t-1$ and expectation at date t of variables dated $t+1$ appear. To find the state variables, one needs to find all (linear combinations of) variables dated $t-1$ in these equations: the endogenous state variable is capital, k_{t-1}, whereas the exogenous state variable is the technology parameter, z_{t-1}. Note that there are as many expectational equations as there are endogenous state variables.

3.8.2 *Simplify*

The coefficients of the equations above need to be collected in the appropriate matrices to restate these equations in the form required for Section 3.4: this is a straightforward

---

[22] Another way to see that constants can in the end be dropped is to note that the steady state is characterized by $c_t = k_t = y_t = k_{t-1} = 0$. If one replaces all log-deviations with zero, only the constant terms remain, and that equation can be subtracted from the equation for general c_t, k_t, y_t and k_{t-1} above.

exercise. Here, however, we plan to proceed to solve the model "by hand". Aside from a bit of algebra, this turns out to be reasonably easy to do.

Formally, equations (i)–(iv) in their log-linearized form can be solved for c_t, n_t, y_t, r_t in terms of k_t, λ_t and z_t at leads and lags by matrix inversion: the solution can then be used to substitute out these variables in equations (v) and (vi). Doing it by hand proceeds as follows. Replace n_t in (iv) with (ii):

$$y_t = \frac{1}{\rho} z_t + k_{t-1} + \frac{1-\rho}{\rho} \lambda_t$$

Now, eliminate c_t, n_t, y_t, r_t in (v) and (vi). Using the abbreviations

$$\begin{aligned}
\alpha_1 &= \frac{\bar{Y}}{\bar{K}} + (1-\delta) \\
\alpha_2 &= \frac{\bar{C}}{\bar{K}} + \frac{1-\rho}{\rho} \frac{\bar{Y}}{\bar{K}} \\
\alpha_3 &= \frac{\bar{Y}}{\rho \bar{K}} \\
\alpha_4 &= 0 \\
\alpha_5 &= 1 + (1-\rho) \frac{\bar{Y}}{\bar{R}\bar{K}} \\
\alpha_6 &= \frac{\bar{Y}}{\bar{R}\bar{K}}
\end{aligned}$$

we obtain

$$0 = -k_t + \alpha_1 k_{t-1} + \alpha_2 \lambda_t + \alpha_3 z_t \tag{3.45}$$

$$0 = E_t[-\lambda_t + \alpha_4 k_t + \alpha_5 \lambda_{t+1} + \alpha_6 z_{t+1}] \tag{3.46}$$

(We keep α_4 for later use. Obviously, we could drop it here.)

3.8.3 *Method* 1: *Undetermined coefficients*

Postulate that

$$k_t = \eta_{KK} k_{t-1} + \eta_{Kz} z_t \tag{3.47}$$

$$\lambda_t = \eta_{\lambda K} k_{t-1} + \eta_{\lambda z} z_t \tag{3.48}$$

Note that $\eta_{\lambda K}$ etc. can be interpreted as elasticities: if $k_{t-1} = 0.01$, that is, if K_{t-1} deviates from its steady state by 1 per cent, and $z_t = 0$, then $\lambda_t = 0.01 * \eta_{\lambda K}$, that is, Λ_t

deviates from its steady state by $\eta_{\lambda K}$ per cent. In terms of the notation in Section 3.4:

$$\begin{aligned}\eta_{\lambda K} &= R\\ \eta_{\lambda z} &= S\\ \eta_{KK} &= P\\ \eta_{Kz} &= Q\end{aligned}$$

Plug the postulated relationships into equations (3.45) once and (3.46) "twice" and exploit $E_t[z_{t+1}] = \psi z_t$, so that only k_{t-1} and z_t remain:

$$\begin{aligned}0 =& (-\eta_{KK} + \alpha_1 + \alpha_2\eta_{\lambda K})\,k_{t-1}\\ &+ (-\eta_{Kz} + \alpha_2\eta_{\lambda z} + \alpha_3)\,z_t\\ 0 =& (-\eta_{\lambda K} + \alpha_4\eta_{KK} + \alpha_5\eta_{\lambda K}\eta_{KK})\,k_{t-1}\\ &+ (-\eta_{\lambda z} + \alpha_4\eta_{Kz} + \alpha_5\eta_{\lambda K}\eta_{Kz} + (\alpha_5\eta_{\lambda z} + \alpha_6)\,\psi)\,z_t\end{aligned}$$

Compare coefficients.

1. On k_{t-1}:

$$\begin{aligned}0 &= -\eta_{KK} + \alpha_1 + \alpha_2\eta_{\lambda K}\\ 0 &= -\eta_{\lambda K} + \alpha_4\eta_{KK} + \alpha_5\eta_{\lambda K}\eta_{KK}\end{aligned}$$

   Solve the first equation for $\eta_{\lambda K}$ and substitute out in the second. One gets the *characteristic quadratic equation*

$$0 = \eta_{KK}^2 - \left(\alpha_1 - \frac{\alpha_2}{\alpha_5}\alpha_4 + \frac{1}{\alpha_5}\right)\eta_{KK} + \frac{\alpha_1}{\alpha_5} \tag{3.49}$$

   solvable with the usual formula:

$$\eta_{KK} = \frac{1}{2}\left(\left(\alpha_1 - \frac{\alpha_2}{\alpha_5}\alpha_4 + \frac{1}{\alpha_5}\right) \pm \sqrt{\left(\alpha_1 - \frac{\alpha_2}{\alpha_5}\alpha_4 + \frac{1}{\alpha_5}\right)^2 - 4\frac{\alpha_1}{\alpha_5}}\right) \tag{3.50}$$

   Note finally that

$$\eta_{KK,1}\eta_{KK,2} = \frac{\alpha_1}{\alpha_5} = \bar{R} = \frac{1}{\beta}$$

   to see that at most one root is stable. Use that one. Calculate $\eta_{\lambda K}$.

2. On z_t:

$$\begin{aligned}0 &= -\eta_{Kz} + \alpha_2\eta_{\lambda z} + \alpha_3\\ 0 &= -\eta_{\lambda z} + \alpha_4\eta_{Kz} + \alpha_5\eta_{\lambda K}\eta_{Kz} + (\alpha_5\eta_{\lambda z} + \alpha_6)\,\psi\end{aligned}$$

   Given that we already know $\eta_{\lambda K}$, these two equations are two linear equations in $\eta_{\lambda z}$ and η_{Kz}, which can be solved easily:

$$\begin{aligned}\eta_{\lambda_z} &= \frac{\alpha_4\alpha_3 + \alpha_5\eta_{\lambda K}\alpha_3 + \alpha_6\psi}{1 - \alpha_4\alpha_2 - \alpha_5\eta_{\lambda K}\alpha_2 - \alpha_5\psi}\\ \eta_{Kz} &= \alpha_2\eta_{\lambda z} + \alpha_3\end{aligned}$$

3.8.4 *Method* 2: *Blanchard–Kahn*

This section is inspired by Blanchard and Kahn (1980) as well as the work by Roger Farmer and his colleagues, in particular Farmer (1993). Let ϕ_t be the prediction error in equation (3.46),

$$0 = \phi_{t+1} - (-\lambda_t + \alpha_5 \lambda_{t+1} + \alpha_4 k_t + \alpha_6 z_{t+1}) \tag{3.51}$$

Write this equation together with equation (3.45) and the equation for z_t as a three-variable first-order difference equation,

$$X \begin{bmatrix} k_t \\ \lambda_{t+1} \\ z_{t+1} \end{bmatrix} = Y \begin{bmatrix} k_{t-1} \\ \lambda_t \\ z_t \end{bmatrix} + Z \begin{bmatrix} \epsilon_{t+1} \\ \phi_{t+1} \end{bmatrix} \tag{3.52}$$

where[23]

$$X = \begin{bmatrix} \alpha_4 & \alpha_5 & \alpha_6 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad Y = \begin{bmatrix} 0 & 1 & 0 \\ \alpha_1 & \alpha_2 & \alpha_3 \\ 0 & 0 & \psi \end{bmatrix}, \qquad Z = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \tag{3.53}$$

Multiply by X^{-1} and diagonalize the matrix $X^{-1}Y$:

$$X^{-1}Y = Q^{-1}\Omega Q$$

where Ω is diagonal,[24] containing the eigenvalues $\omega_1, \omega_2, \omega_3$. To be a little more explicit, note that

$$X^{-1} = \begin{bmatrix} 0 & 1 & 0 \\ 1/\alpha_5 & -\alpha_4/\alpha_5 & -\alpha_6/\alpha_5 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.54}$$

$$X^{-1}Y = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ -\alpha_1\alpha_4/\alpha_5 & (1-\alpha_2\alpha_4)/\alpha_5 & (-\alpha_3\alpha_4 - \psi\alpha_6)/\alpha_5 \\ 0 & 0 & \psi \end{bmatrix}$$

$$X^{-1}Z = \begin{bmatrix} 0 & 0 \\ -\alpha_6/\alpha_5 & 1/\alpha_5 \\ 1 & 0 \end{bmatrix}$$

In particular, the characteristic equation for the eigenvalues is given by

$$p(\omega) = \left((\alpha_1 - \omega)\left(\frac{1-\alpha_2\alpha_4}{\alpha_5} - \omega\right) + \frac{\alpha_1\alpha_2\alpha_4}{\alpha_5} \right)(\psi - \omega)$$

---

[23] The notation here differs from the notation in Theorem 3.5.

[24] In general, X may not be invertible, in which case it is better to try to solve a generalized eigenvalue problem. Furthermore, it may be the case that Ω is not diagonalizable. One can then use the QZ method instead; see recent papers by Sims (1999), King and Watson (1995, 1997) and Klein (1998).

Comparing the expression in the first bracket to equation (3.49), we see that one can pick ω_1 to be the explosive root in (3.50), ω_2 to be the stable root in (3.50) and $\omega_3 = \psi$. Let

$$x_t = Q\begin{bmatrix} k_{t-1} \\ \lambda_t \\ z_t \end{bmatrix}, \qquad \zeta_{t+1} = QX^{-1}\begin{bmatrix} \epsilon_{t+1} \\ \phi_{t+1} \end{bmatrix}$$

so that the first-order matrix difference equation can be written in "decoupled" form as

$$\begin{bmatrix} x_{1,t+1} \\ x_{2,t+1} \\ x_{3,t+1} \end{bmatrix} = \begin{bmatrix} \omega_1 & 0 & 0 \\ 0 & \omega_2 & 0 \\ 0 & 0 & \omega_3 \end{bmatrix}\begin{bmatrix} x_{1,t} \\ x_{2,t} \\ x_{3,t} \end{bmatrix} + \begin{bmatrix} \zeta_{1,t+1} \\ \zeta_{2,t+1} \\ \zeta_{3,t+1} \end{bmatrix} \tag{3.55}$$

(To compare this to Farmer, note that he usually calculates ω_i^{-1} as roots λ_i. There is no substantive difference.) What can we learn from this?

1. The first of these three equations can be iterated forward to give

$$\begin{aligned} x_{1,t} &= \omega_1^{-1}x_{1,t+1} - \omega_1^{-1}\zeta_{1,t+1} \\ &= \omega_1^{-2}x_{1,t+2} - \omega_1^{-2}\zeta_{1,t+2} - \omega_1^{-1}\zeta_{1,t+1} \\ &= \ldots \\ &= -\sum_{j=1}^{\infty}\omega_1^{-j}\zeta_{1,t+j} \end{aligned}$$

   Taking conditional expectations with respect to t on both sides, we see that

$$x_{1,t} \equiv 0$$

   This thus imposes one linear restriction on the vector $[k_{t-1},\ \lambda_t,\ z_t]'$,

$$0 = Q_{11}k_{t-1} + Q_{12}\lambda_t + Q_{13}z_t$$

   It turns out that this linear restriction can be written as

$$\lambda_t = \eta_{\lambda K}k_{t-1} + \eta_{\lambda z}z_t$$

   with the same coefficients $\eta_{\lambda K}$ and $\eta_{\lambda z}$ as calculated in Section 3.8.3.

2. Furthermore, we obtain

$$\zeta_{1,t+1} = x_{1,t+1} - \omega_1 x_{1,t} \equiv 0$$

   which imposes a linear relationship between the forecast error ϕ_{t+1} and the productivity innovation ϵ_{t+1},

$$0 = (QX^{-1})_{11}\epsilon_{t+1} + (QX^{-1})_{12}\phi_{t+1}$$

   This too turns out to be a restatement of what we have already found in Section 3.8.3: with the decision rules there, one can calculate

$$\begin{aligned} \phi_{t+1} &= (-\lambda_t + \alpha_4 k_t + \alpha_5\lambda_{t+1} + \alpha_6 z_{t+1}) \\ &\quad - E_t[-\lambda_t + \alpha_4 k_t + \alpha_5\lambda_{t+1} + \alpha_6 z_{t+1}] \\ &= (\alpha_5\eta_{\lambda z} + \alpha_6)\,\epsilon_{t+1} \end{aligned}$$

3. Equipped with this knowledge, replace ϕ_{t+1}, λ_t and λ_{t+1} in the second equation in (3.55). Additionally taking conditional expectations with respect to t,[25] this equation in (3.55) now turns out to be rewritable as

$$k_t = \eta_{KK} k_{t-1} + \eta_{Kz} z_t$$

with the coefficients calculated in Section 3.8.3.
4. Finally, the third equation in (3.55) is the original equation

$$z_{t+1} = \psi z_t + \epsilon_{t+1}$$

We see that we re-created the solution found in Section 3.8.3.

3.8.5 *Farmer's model: Indeterminacy*

In Farmer's (1993) model, there are decreasing returns at the firm level but increasing returns in the aggregate: the aggregate production function is given by

$$Y_t = Z_t K_{t-1}^{\mu} N_t^{\nu}$$

whereas the factors capital and labour receive the shares $\tilde{\mu}$ and $\tilde{\nu}$. In addition, the instantaneous utility function is given by

$$\log C_t - A \frac{N_t^{1+\chi}}{1+\chi}$$

with $\chi \geq 0$. Finally, there is a time trend in log-productivity growth, which introduces a factor γ in some equations. The model in Section 3.8.1 is the special case

$$\rho = \mu = 1 - \nu = \tilde{\mu} = 1 - \tilde{\nu}, \qquad \chi = 0, \qquad \gamma = 1$$

and hence,

$$\frac{1+\chi}{1+\chi-\nu} = \frac{1}{\rho}$$

Collecting the equations for Farmer's model, one obtains the following:

#	Equation	Log-linearized
(i)	$\frac{1}{C_t} = \Lambda_t$	$-c_t = \lambda_t$
(ii)	$AN_t^{\chi} = \Lambda_t \tilde{\nu} \frac{Y_t}{N_t}$	$(1+\chi)n_t = y_t + \lambda_t$
(iii)	$R_t = \tilde{\mu} \frac{Y_t}{K_{t-1}} + 1 - \delta$	$\bar{R} r_t = \tilde{\mu} \frac{\bar{Y}}{\bar{K}} (y_t - k_{t-1})$
(iv)	$Y_t = \bar{Z} e^{z_t} K_{t-1}^{\mu} N_t^{\nu}$	$y_t = z_t + \mu k_{t-1} + \nu n_t$
(v)	$C_t + \gamma K_t = Y_t + (1-\delta) K_{t-1}$	$\bar{C} c_t + \gamma \bar{K} k_t = \bar{Y}\ y_t + (1-\delta) \bar{K} k_{t-1}$
(vi)	$\Lambda_t = \frac{\beta}{\gamma} E_t[\Lambda_{t+1} R_{t+1}]$	$\lambda_t = E_t[\lambda_{t+1} + r_{t+1}]$
(vii)	$\log Z_t = (1-\psi) \log \bar{Z} + \psi \log Z_{t-1} + E_t$	$Z_t = \psi Z_{t-1} + E_t$

---

[25] Note that k_t is chosen at date t, and thus it cannot depend on information dated $t+1$ or later. Regardless of the timing convention for the subscript of k_t, this has to be true. For this reason, the notation k_t rather than k_{t+1} for the capital stock chosen at date t may be preferable.

As before, use (ii) to replace n_t in (iv), yielding

$$y_t = \frac{1+\chi}{1+\chi-\nu} z_t + \frac{\mu(1+\chi)}{1+\chi-\nu} k_{t-1} + \frac{\nu}{1+\chi-\nu} \lambda_t$$

and as before, reduce the remaining equations to the two-equation system (3.45) and (3.46), where now

$$\begin{aligned}
\alpha_1 &= \frac{\mu(1+\chi)}{1+\chi-\nu} \frac{\bar{Y}}{\gamma \bar{K}} + \frac{1-\delta}{\gamma} \\
\alpha_2 &= \frac{\bar{C}}{\gamma \bar{K}} + \frac{\nu}{1+\chi-\nu} \frac{\bar{Y}}{\gamma \bar{K}} \\
\alpha_3 &= \frac{1+\chi}{1+\chi-\nu} \frac{\bar{Y}}{\gamma \bar{K}} \\
\alpha_4 &= \left(\frac{\mu(1+\chi)}{1+\chi-\nu} - 1 \right) \frac{\tilde{\mu}\bar{Y}}{\bar{R}\bar{K}} \\
\alpha_5 &= 1 + \frac{\tilde{\mu}\nu}{1+\chi-\nu} \frac{\bar{Y}}{\bar{R}\bar{K}} \\
\alpha_6 &= \frac{\tilde{\mu}(1+\chi)}{1+\chi-\nu} \frac{\bar{Y}}{\bar{R}\bar{K}}
\end{aligned}$$

3.8.6 *Undetermined coefficients: A first try*

One could try to solve this model with the method of undetermined coefficients just as in Section 3.8.3. So, let us postulate the law of motion (3.48), (3.47) and see what we get. As before, we get the characteristic equation (3.49) with the two solutions given by equation (3.50). Since Hansen's model is contained as a special case, we may again get one stable and one unstable root and then we can proceed to the complete solution as above in Section 3.8.3. However, there is now no longer a guarantee that at most one root is stable. In fact, the whole point of Farmer's investigation is to provide a model in which both roots are stable. So, what should one do? Should one just pick one of the two roots in some arbitrary manner? Or does one need to start from another law of motion? We will come back to answer these questions in Section 3.8.8.

3.8.7 *Blanchard–Kahn*

It may be helpful to take a look first at what the Blanchard–Kahn method delivers. As before, introduce ϕ with equation (3.51), and write equations (3.45) and (3.46) as equation (3.52) with the matrices as in (3.53). We proceed in the same manner and get to equation (3.55). If exactly one of the roots ω_i is greater than unity, we can proceed as before. But the interesting case is now that in which $|\omega_i| < 1$.[26] In that case, we must

---

[26] In fact, some roots may even be complex-valued. This can impose some restrictions: obviously, we want the economic variables to be real-valued. The principle is that a real-valued characteristic polynomial will always give rise to pairs of *conjugate* complex roots, if it gives rise to complex roots at all. It is then important that either both roots of such a pair are carried along in the calculations or both are eliminated. This usually assures that all economic variables remain real-valued.

iterate all three equations "backwards". A simpler way to state this is that equation (3.52) is already the correct forward dynamics of the model,

$$\begin{bmatrix} k_t \\ \lambda_{t+1} \\ z_{t+1} \end{bmatrix} = X^{-1}Y \begin{bmatrix} k_{t-1} \\ \lambda_t \\ z_t \end{bmatrix} + X^{-1}Z \begin{bmatrix} \epsilon_{t+1} \\ \phi_{t+1} \end{bmatrix} \tag{3.56}$$

provided ϕ_{t+1} is such that k_t does *not* depend on information dated $t+1$ or beyond. Put differently, ϕ_{t+1} must be such that

$$0 = (X^{-1}Z)_{11}\epsilon_{t+1} + (X^{-1}Z)_{12}\phi_{t+1} \tag{3.57}$$

Check equation (3.54): since $(X^{-1}Z)_{11} = (X^{-1}Z)_{12} = 0$, equation (3.57) imposes *no* restrictions on ϕ_{t+1}. In particular, we see the following:

1. There is now a second, "artificial" state variable λ_t. Obviously, while λ_t has always been in the information set when choosing λ_{t+1}, that information was "irrelevant" before in Hansen's model: the only dynamic linkage between periods came from capital k_t and from productivity z_t. Put differently: sometimes the state variables may not be what you think they should be!
2. There is no longer some linear restriction on the relationship between ϵ_{t+1} and ϕ_{t+1}. Thus, ϕ_{t+1} can be "anything" (as long as it satisfies $E_t[\phi_{t+1}] = 0$). In economic terms, ϕ_{t+1} introduces *sunspots*, *self-fulfilling prophecies* or *animal spirits*. In particular, agents may "co-ordinate" on the random variable ϵ_{t+1}, $\phi_{t+1} = \tau\epsilon_{t+1}$ for some coefficient τ: any value for τ is legitimate! This creates an *endogenous* sunspot.

3.8.8 *Undetermined coefficients: A second try*

With an eye on the Blanchard–Kahn approach, we see that we need as many state variables as there are stable roots. One interesting special case is the solution in which λ_{t+1} is *predetermined*, that is, already known[27] at date t. That case can now be solved with the method of undetermined coefficients by postulating the law of motion,

$$\begin{aligned} \lambda_{t+1} &= \eta_{\lambda K}k_{t-1} + \eta_{\lambda\lambda}\lambda_t + \eta_{\lambda z}z_t \\ k_t &= \eta_{KK}k_{t-1} + \eta_{K\lambda}\lambda_t + \eta_{Kz}z_t \end{aligned}$$

Note what happened compared to equations (3.48) and (3.47):

1. There is a second state variable, which we simply pick[28] to be λ.
2. The decision rule is now for λ_{t+1} rather than λ_t, since λ_{t+1} is now postulated to be predetermined.

---

[27] Blanchard and Kahn introduced the term "predetermined". Thus, both λ_{t+1} and k_t are now predetermined. This terminology can be useful but can also be confusing. A better way may be to think of some variables to be known at date t and others at date $t+1$ etc., and to use the subscripts in a manner consistent with that. Unfortunately, it can be the case that a linear combinations of some variables is already known at date t, whereas the variables individually are only known at date $t+1$.

[28] In general, some variables may turn out to be unsuitable candidates as additional state variables.

The solution for the undetermined coefficients is simple: compare the equations above with equation (3.56) to see that the coefficients are given directly by the appropriate coefficients in $X^{-1}Y$. Also note that we implicitly picked ϕ_{t+1} to satisfy

$$0 = (X^{-1}Z)_{21}\epsilon_{t+1} + (X^{-1}Z)_{22}\phi_{t+1}$$

or, with equation (3.54),

$$\phi_{t+1} = \alpha_6\epsilon_{t+1}$$

To introduce sunspots into the solution as well, one can proceed as follows. Distinguish between λ_{t+1}, known at date $t+1$, and $\lambda_t^e = E_t[\lambda_{t+1}]$. In addition to the two equations (3.45) and (3.46), we now have a third equation

$$\lambda_t^e = E_t[\lambda_{t+1}] \tag{3.58}$$

Introduce a "sunspot" random variable θ_{t+1} with $E_t[\theta_{t+1}] = 0$. Regard λ_t^e as the additional state variable and postulate

$$\begin{aligned}
\lambda_t^e &= \eta_{\lambda^e K}k_{t-1} + \eta_{\lambda^e\lambda^e}\lambda_{t-1}^e + \eta_{\lambda^e z}z_t + \eta_{\lambda^e\theta}\theta_t \\
k_t &= \eta_{KK}k_{t-1} + \eta_{K\lambda^e}\lambda_{t-1}^e + \eta_{Kz}z_t + \eta_{K\theta}\theta_t \\
\lambda_t &= \eta_{\lambda K}k_{t-1} + \eta_{K\lambda^e}\lambda_{t-1}^e + \eta_{\lambda z}z_t + \eta_{\lambda\theta}\theta_t
\end{aligned}$$

and solve by the method of undetermined coefficients. Note that this is legitimate whether we have one or two stable roots from (3.50): so we shall proceed with both in mind. Since θ_t is not a fundamental shock, the scale of its coefficients will be undetermined. Thus, normalize by setting $\eta_{\lambda\theta} = 1$. That way,

$$\theta_t = \lambda_t - \lambda_{t-1}^e$$

is the forecast error in λ_t. Comparison with equations (3.56) and (3.54) reveals that

$$\begin{aligned}
\theta_{t+1} &= (X^{-1}Z)_{21}\epsilon_{t+1} + (X^{-1}Z)_{22}\phi_{t+1} \\
&= \frac{-\alpha_6}{\alpha_5}\epsilon_{t+1} + \frac{1}{\alpha_5}\phi_{t+1}
\end{aligned}$$

Equation (3.58) turns out to deliver

$$\eta_{\lambda K} = 0, \qquad \eta_{\lambda\lambda^e} = 1, \qquad \eta_{\lambda z} = 0$$

To proceed further, distinguish between the saddle-point stable case and the case of two stable roots:

1. If there is one stable and one unstable root, it turns out that

   $$\eta_{\lambda^e\lambda^e} = \eta_{K\lambda^e} = 0$$

   In words, even though a second state variable was added, the method of undetermined coefficients eliminates it automatically. Furthermore, there will be tight relationship between θ_{t+1} and ϵ_{t+1}.

2. If there are two stable roots, $\eta_{\lambda^e\lambda^e}$ and $\eta_{K\lambda^e}$ are no longer zero. Furthermore, because $(X^{-1}Z)_{11} = (X^{-1}Z)_{12} = 0$, there no longer will be a tight relationship between θ_{t+1} and ϵ_{t+1}, indicating the admissibility of a sunspot.

So, with the rule that one needs (at least) as many state variables as there are stable roots, the method of undetermined coefficients can be used fairly simply for the case of two stable roots as well, if one is only interested in the solution in which λ_{t+1} is "predetermined" at date t. Introducing sunspots requires a little more work, and, at least for this simple model, the Blanchard–Kahn approach seems clearer and more straightforward. On the other hand, the method of undetermined coefficients seems more straightforward for the saddle-point stable case. So it is useful to know about both.

3.8.9 *Final remarks*

The method of undetermined coefficients is particularly useful if the economy is saddle-point stable. Sometimes, however, there are "too many" stable roots. The method of undetermined coefficients can still be used then. Sunspot solutions may arise.

3.9 Appendix: Description of the MATLAB programs

Here we will describe some MATLAB programs to carry out the calculations for Sections 3.4 and 3.6.

The easiest way to learn about these programs is to store all of them, start MATLAB from the directory where they are stored and type "readme". This will execute the **readme.m** file, providing some documentation.

As the time of writing, the newest version of the files is version 2. To see how version 2 files differ from the previous version, distributed until spring 1997, type "whatsnew" within MATLAB, which executes the file **whatsnew.m**, printing relevant messages as a result. To see quickly how these files work, start MATLAB and type "exampl0" to calculate through example 0, which is the stochastic neoclassical growth model, or type "exampl1" to calculate through example 1, which is Hansen's (1985) real business cycle model of Section 3.8.1, linearized slightly differently. There are more examples, enumerated as "examplNN", where NN stands for their number. To see what any particular example, say, **exampl1.m**, does, type "help exampl1" within MATLAB. Use the example files as templates for your own work. Alternatively, declare all required matrices and type "do_it" to do all calculations. All the **examplNN.m** files call **do_it.m** at the very end.

The files which perform all the calculations (i.e. all the files aside from the **examplNN.m** files, the **readme.m** file and the **whatsnew.m** file) are as follows:

do_it.m does it all, once all needed matrices are defined. This file calls all the other programs. Thus, examining this file will tell you in which sequence all the other calculations are performed.

enlarge.m allows you to manipulate letter sizes on plots and other properties of plots. Useful for producing slides or plots for publication.

impresp.m calculates and shows impulse responses to shocks (see Section 3.6).

mom_out.m produces output. To be called after **moments.m**.

moments.m calculates second moment properties (see Section 3.6).

options.m sets the options for all programs. It is called by **do_it.m** and needs to be called if one of the following routines is used in isolation.

sol_out.m produces output. To be called after **solve.m**.

solve.m solves for the recursive equilibrium law of motion with the theorems of Section 3.4.

All files are extensively documented. Type, say, "help impresp" in MATLAB to get more information on what the program **impresp.m** does. Note that these files set some additional variables, which you may have used before: thus, be careful not to use names appearing in the programs. If you have a question, please read this chapter and the documentation carefully. These files are provided as a free service, without technical support. However, if there are serious flaws or serious ways to improve on these programs, I would like to hear about them. Feel free to copy and modify these files, and use them at your own risk. There is absolutely no guarantee that they work as intended.