System Reduction and Solution Algorithms for Singular Linear Difference Systems Under Rational Expectations

Robert G. King
University of Virginia and
Federal Reserve Bank of Richmond

Mark W. Watson
Princeton University*

May 1995, revised October 1997

Abstract

A first-order linear difference system under rational expectations is,

$$AEy_{t+1}|I_t = By_t + C(\mathbf{F})Ex_t|I_t,$$

where y_t is a vector of endogenous variables; x_t is a vector of exogenous variables; $Ey_{t+1}|I_t$ is the expectation of y_{t+1} given date t information; and $C(\mathbf{F})Ex_t|I_t = C_0x_t + C_1Ex_{t+1}|I_t + ... + C_nEx_{t+n}|I_t$. Many economic models can be written in this form, especially if the matrix A is permitted to be singular.

If the model is solvable, y_t can be divided into two sets of variables: dynamic variables d_t that evolve according $Ed_{t+1}|I_t = Wd_t + \Psi_d(\mathbf{F})Ex_t|I_t$ and other variables which that obey the dynamic identities $f_t = -Kd_t - \Psi_f(\mathbf{F})Ex_t|I_t$.

This paper provides an algorithm that constructs the reduced system $Ed_{t+1}|I_t = Wd_t+\Psi_d(\mathbf{F})Ex_t|I_t$ for any solvable linear difference system. We also provide algorithms for computing (i) perfect foresight solutions and (ii) Markov decision rules that can be used when there is a unique solution.

JEL: C62, C63

*We have had useful support on this research project from many people. Examples of singularities in real business cycle models provided by Marianne Baxter, Mario Crucini, and Sergio Rebelo were one of the motivations for this research. Matthew Van Vlack and Pierre Sarte aided in the development of the PC-MATLAB programs that tested our algorithms. Michael Kouparitsas graciously did the initial translations into GAUSS. Financial support was provided by the National Science Foundation through grants SES-91-22463 and SBR-94-09629.

1 Introduction

A large class of linear dynamic economic models can be represented as first order linear difference systems under rational expectations. While seemingly special, this framework has proved to be very general in that it can be used for models with (i) expectations of variables more than one period into the future, (ii) lags of endogenous variables, and (iii) lags of expectations of endogenous variables. For this workhorse model, we provide a general solution method that is fast, flexible and easy to use. Since its development several years ago, we have put this method to work in a range of research projects. It has been a routine matter to solve models with several hundred equations behavioral equations—many of which are identities introduced for the researcher's convenience—and with reduced forms that involve thirty or more endogenous state variables.

Our research focuses on a general first-order model which we call a singular linear difference system under rational expectations. Defined more precisely in the next section, this model nests the framework of Blanchard-Kahn (1980) and also encompasses many economic models that fall outside of that earlier setup. In companion research, we provide a theoretical analysis that highlights the conditions under which a unique stable rational expectations solution exists for the singular linear rational expectations model.

The model solution method developed in the current paper involves two stages. In the first, we uncover a reduced dimension dynamic system present in the model, essentially by determining how identities restrict the dynamic behavior some of the model's variables: we call this *system reduction*. In the second, we develop alternative strategies for computing the *rational expectations solution* to the reduced dimension system.

The organization of the paper is as follows. In section 2, we provide a formal definition of the problem, a review of conventional system reduction procedures, and four examples of system reduction in macroeconomic models. In section 3, we describe a reduction algorithm and establish that it can always be used to reduce any rational expectations model for which there exists a solution. In section 4, we outline ways of solving the reduced and complete model under rational expectations. In section 5, we produce a brief summary and conclusion.

2 The Problem and Examples

Many dynamic linear rational expectations models can be cast in the form:

$$AEy_{t+1}|I_t = By_t + C(\mathbf{F})Ex_t|I_t, \qquad t = 0, 1, 2, \ldots \tag{1}$$

where A and B are matrices of constants; y_t is a vector of endogenous variables and x_t is a vector of exogenous variables; and $Ey_{t+1}|I_t$ is the rational expectation of y_{t+1} given date t information which includes x_t and y_t (we also sometimes write expectations as $E_t y_{t+1}$). The specification of the model includes a "distributed lead" effect of exogenous variables, $C(\mathbf{F})Ex_t|I_t = C_0 x_t + C_1 Ex_{t+1}|I_t + ... + C_n Ex_{t+n}|I_t$, where the C_i are matrices of constants. Following Sargent (1978), $\mathbf{F}$ is the lead operator which shifts the dating of the variable but not the information set: for any random variable w_t, $\mathbf{F}^h Ew_{t+n}|I_t = Ew_{t+n+h}|I_t$.

The vector y_t is assumed to be divided into elements of the column vector k_t, which are predetermined, and elements of the column vector Λ_t, which are not predetermined. By predetermined, we mean two things. First, $Ek_{t+1}|I_t = k_{t+1}$ so that the value of k_{t+1} is known at time period t. Second, we require that initial conditions are given for k_0 and that these initial conditions are not constrained in any way by the model. Finally, we require that the researcher know which of the variables in the model are predetermined. Without loss of generality, we assume that the variables in y_t are ordered as $y_t = (\Lambda_t' \quad k_t')'$.[1]

If A is nonsingular, then we can write the dynamic system (1) as $Ey_{t+1}|I_t = A^{-1}By_t + A^{-1}C(\mathbf{F})Ex_t|I_t$. In this case, Blanchard and Kahn (1980) show how to solve for "stable" solutions to this linear difference system, i.e., those which depend in a stable manner on expectations of x and initial conditions for k. In the process of their canonical variable treatment of this topic, they develop rank and order conditions on the transformation between original and canonical variables that must be satisfied if there is to be a unique stable solution.

But many models in this class have A singular and, hence, some form of reduction is necessary. Below we provide a set of motivating examples, which also show how the singularity of B can complicate the reduction process. If A is singular, then a necessary condition for the existence of a unique solution is that the determinantal polynomial $\Delta(z) = |Az - B|$ is not zero for all values of z. Under this condition, our companion research develops a canonical variables representation of (1) and show that there are direct generalizations of the Blanchard-Kahn (1980) conditions which assure a unique solution.[2]

2.1 Definition of System Reduction

If (1) has a unique stable solution, King and Watson (1997) show that y_t can be divided into two groups: dynamic variables d_t that evolve according to

$$Ed_{t+1}|I_t = Wd_t + \Psi_d(\mathbf{F})Ex_t|I_t \tag{2}$$

and other variables that evolve according to

$$f_t = -Kd_t - \Psi_f(\mathbf{F})Ex_t|I_t. \tag{3}$$

[1] Two types of macroeconomic models cannot be expressed in the form form we study in this paper. First, in a some models, agents are assumed to receive different types of information at different points within a discrete time period. (An example is Svennson's (1985) work on the effects of monetary disturbances on exchange rates, prices and interest rates). Second, in other models, agents are assumed to learn about current events at least partly from the values of current prices and quantities. (An example is Lucas's (1972) work on the effects of monetary disturbances on nominal prices and real activity). The solution of both timing and incomplete information models is facilitated by consideration of solution of the singular linear difference system that we study in this paper. In King and Watson [1995], we provide a method for solving "timing models" that works for most, but perhaps not all, singular dynamic systems that we can solve under the assumption of a single informational period.

[2] Another approach to solving singular linear difference systems is provided in work by Sims (1989), who uses the QZ-algorithm to transform the a singular system into an equivalent system that can be solved recursively. All models that can be solved with our technology can also be solved with Sims'. The QZ-algorithm originates in Moler and Stewart (1971) and is discussed by Golub and Van Loan [1989, p. 394].

(Our choice of the notation "f" is based on the ideas that these are "flow" variables, like controls or point-in-time shadow prices, and at times below we will use this descriptive phrase to indicate "f".) These variables are nondynamic in the sense that we can express their evolution as a function simply of d_t and $\Psi_f(\mathbf{F})Ex_t|I_t$, but we are not required to otherwise connect f_t and y_{t+1}. Combining (2) and (3) produces the system:

$$\begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} E_t \begin{bmatrix} f_{t+1} \\ d_{t+1} \end{bmatrix} = \begin{bmatrix} I & K \\ 0 & W \end{bmatrix} \begin{bmatrix} f_t \\ d_t \end{bmatrix} + \begin{bmatrix} \Psi_f(\mathbf{F}) \\ \Psi_d(\mathbf{F}) \end{bmatrix} E_t x_t. \tag{4}$$

This transformed system can be obtained from the original system (1) and shares some of its key properties, notably that the eigenvalues of W are the same as the finite roots of $|Az - B|$. To obtain this system, it may be necessary to reorder the elements of the vector of nonpredetermined variables Λ so as to place f at the top as in (4). Equivalently, there is a permutation matrix[3] L such that:

$$\begin{bmatrix} f_t \\ d_t \end{bmatrix} = Ly_t,$$

where L does not reorder the predetermined variables of the model but only the nonpredetermined variables. This rearrangement of the nonpredetermined variables, say $L_\Lambda \Lambda_t = (f_t' \quad \lambda_t')'$, identifies the f_t as "static" nonpredetermined variables and the remaining elements, λ_t, as dynamic nonpredetermined variables. Correspondingly, then, we have that $d_t = (\lambda_t' \quad k_t')'$.

We call the process of moving from (1) to (4) *system reduction*. A central purpose of this paper is to develop an algorithm for computing this system reduction, i.e., for computing the matrices K, W, L and the polynomials $\Psi_d(\mathbf{F})$ and $\Psi_f(\mathbf{F})$, given the system matrices A, B, and $C(\mathbf{F})$.

2.2 Conventional System Reduction

In many contexts in economics and engineering, the researcher has detailed knowledge about how the dynamic system works and a conventional system reduction method is employed (see, for example, the discussion in Luenberger's (1979) dynamic systems textbook). For example, we frequently know which of the elements of y_t are nondynamic variables, such as controls and point-in-time shadow prices, from aspects of the economic problem. This knowledge leads to the conventional form of system reduction, since it implies that we know there is a system of the form:

$$\begin{bmatrix} 0 & 0 \\ A_{df} & A_{dd} \end{bmatrix} E_t \begin{bmatrix} f_{t+1} \\ d_{t+1} \end{bmatrix} = \begin{bmatrix} B_{ff} & B_{fd} \\ B_{df} & B_{dd} \end{bmatrix} \begin{bmatrix} f_t \\ d_t \end{bmatrix} + \begin{bmatrix} C_f(\mathbf{F}) \\ C_d(\mathbf{F}) \end{bmatrix} E_t x_t. \tag{5}$$

[3] An $m \times m$ permutation matrix P is constructed by permuting (switching) the rows of the $m \times m$ identity matrix I_m. Thus, Py simply reorders the elements of y.

The conventional system reduction procedure is to solve $n(f)$ equations for f_t:[4]

$$f_t = -B_{ff}^{-1} B_{fd} d_t - B_{ff}^{-1} C_f(\mathbf{F}) E_t x_t$$

and to use this to obtain:

$$aE_t d_{t+1} = b\ d_t + \Psi_d(\mathbf{F})\ E_t x_t \tag{6}$$

where $a = [A_{dd} - A_{df} B_{ff}^{-1} B_{fd}]$; $b = [B_{dd} - B_{df} B_{ff}^{-1} B_{fd}]$; and $\Psi_d(\mathbf{F}) = [C_d(\mathbf{F}) - B_{df} B_{ff}^{-1} C_f(\mathbf{F}) + A_{df} B_{ff}^{-1} C_f(\mathbf{F})\mathbf{F}]$.

In terms of the structure of this paper, there are several important observations to be made about this reduction of the system. First, it requires that B_{ff} is nonsingular, so that there are required to be as many equations as elements of f. Second, reduction of the dynamic system is typically completed by inverting a to get a dynamic system in the form $Ed_{t+1}|I_t = a^{-1} b d_t + a^{-1} \Psi_d(\mathbf{F}) E x_t | I_t$. This places the system in the form (4) but it also imposes an additional nonsingularity requirement: a must be invertible. Third, reduction typically adds a lead, i.e., there is an additional power of $\mathbf{F}$ in $\Psi_d(\mathbf{F})$ in (6).

We have just discussed conventional system reduction in terms of using some equations to "substitute out" for some variables. Hoever, in terms of the analysis below, it is important to also recognize that conventional system reduction may alternatively be viewed as the application of $T(F) = GF + H$ to the dynamic system with

$$T(F) = GF + H = \begin{bmatrix} 0 & 0 \\ A_{df} B_{ff}^{-1} & 0 \end{bmatrix} F + \begin{bmatrix} B_{ff}^{-1} & 0 \\ -B_{df} B_{ff}^{-1} & I \end{bmatrix}.$$

That is, we create a new dynamic system from $(A\mathbf{F} - B) E_t y_t = C(\mathbf{F}) E_t x_t$ by multiplying both sides by $T(\mathbf{F})$. Notice that since $|T(\mathbf{F})| = |B_{ff}^{-1}| = \tau$, the characteristic polynomial of the transformed system, $|T(z)(Az - B)| = \tau |Az - B|$, has the same roots as that of the original system. It is also notable that the application of $T(F)$ to the dynamic system $[A\mathbf{F} - B] E y_t | I_t = C(\mathbf{F}) E x_t | I_t$ has some different effects on the order of the system's internal and exogenous dynamics. Since there is a special structure to $T(\mathbf{F})$, it follows that $T(\mathbf{F})[A\mathbf{F} - B] = [A^* \mathbf{F} - B^*]$, i.e., premultiplication does not change the first-order nature of the difference equation because of the structure of G and H. However, $T(\mathbf{F}) C(\mathbf{F})$ is typically higher order than $C(\mathbf{F})$, which is another way of stating that a lead is added by conventional system reduction.

As we proceed to discuss the system reduction algorithm, we will find it useful to sometimes discuss aspects of it in terms of "solving equations" and then to express these as "multiplication by matrix polynomials in $\mathbf{F}$": the former is generally more intuitive and the latter more revealing about the consequences of these operations.

2.3 Examples of Singular Systems and System Reduction

To highlight various characteristics of the system reduction process we discuss the problem in the context of four specific models. The first two are variants of the standard log-linear

[4] Throughout the paper, we denote the number of elements in a arbitrary vector w_t by $n(w)$. We also use the notation n_s to indicate the number of variables s in several contexts.

model of price level and interest rate determination introduced by Sargent and Wallace (1975). We then consider two examples of real business cycle models.[5] Within this section, we concentrate on developing simple examples that are transparent; there is no necessary notational overlap with the rest of the paper—model parameters and variables will be used elsewhere in the document in other ways.

2.3.1 Monetary Models

In our first two examples, let R_t be the nominal interest rate, P_t be the price level and M_t be the money stock. Each model contains the Fisher equation:

$$R_t = E_t P_{t+1} - P_t \tag{7}$$

The Basic Sargent-Wallace Model: There is a single exogenous variable (M_t), two endogenous variables (R_t, P_t), and no predetermined variables. The economic specification is:

$$M_t - P_t = -\alpha\ R_t, \tag{8}$$

and the representation $A\ E_t y_{t+1} = B\ y_t + C(\mathbf{F}) E_t x_t$, is:

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} E_t R_{t+1} \\ E_t P_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ \alpha & -1 \end{bmatrix} \begin{bmatrix} R_t \\ P_t \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} M_t. \tag{9}$$

Notice that the matrix A has rank 1.

Looking at the system (7)-(8), one can see that there are a variety of ways to overcome the singularity of A and simultaneously reduce the system. In terms of the economics of the model, it is perhaps most natural to substitute $R_t = E_t P_{t+1} - P_t$ and to express the system's dynamics in terms of $E_t P_{t+1}$ and P_t. The result is a first-order difference equation, $\alpha E_t P_{t+1} = (1+\alpha)P_t - M_t$, which can be readily solved. That is, the reduced system has the form $E_t d_{t+1} = W\ d_t + \Psi_d(\mathbf{F}) E_t x_t$ with $d_t = P_t$, $W = (1+\alpha)/\alpha$ and $\Psi_d(\mathbf{F}) E_t x_t = -(1/\alpha) M_t$. To recover R_t, one possibility is to use $R_t = E_t P_{t+1} - P_t$. However, in specifying (4), we required that the nondynamic variables, f_t, can be written as $f_t = -K\ d_t - \Psi_d(\mathbf{F}) E_t x_t$: the previous reduction is inadmissible because it depends on expectations of future y (*i.e.*, on $E_t P_{t+1}$). To correct this problem in the current example, we must use the money demand function to eliminate R_t rather than the Fisher equation: $R_t = -(M_t - P_t)/\alpha$.

The solution described above is not unique. Another reduction is given by the following procedure: substituting $P_t = M_t + \alpha\ R_t$ into $R_t = E_t P_{t+1} - P_t$, we find that $\alpha E_t R_{t+1} = (1+\alpha)R_t - M_t + E_t M_{t+1}$. That is, this system reduction adds a lead of $E_t M_t$ to the exogenous variables. In our general discussion in the previous section, we found that system reduction typically adds leads, but this example's pair of system reductions indicate that the addition of leads is specific to the particular reduction selected. In the current system reduction, we again use the money demand function to produce: $P_t = M_t - \alpha R_t$.

[5] PC-MATLAB code for solving these models is contained on the directory `\examples` on the distribution diskette obtainable from the authors.

Overall, our first example shows that the system reduction process is not unique and that it typically, although not necessarily, adds leads to the reduced dynamic system.

Predetermined Components of Money Demand: We now add a "gradual adjustment" of money demand to the price level so that our system contains a predetermined variable.

$$M_t - \theta\ P_t - (1-\theta)P_{t-1} = -\alpha\ R_t \tag{10}$$

Our dynamic system now is:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} E_t R_{t+1} \\ E_t P_{t+1} \\ E_t P_t \end{bmatrix} = \begin{bmatrix} \alpha & -\theta & -(1-\theta) \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_t \\ P_t \\ P_{t-1} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} M_t \tag{11}$$

Notice that the matrix A is again singular. It is again possible to find an appropriate system representation simply by inspection: one simply solves the first equation for R_t and uses the result to eliminate this variable from the dynamic subsystem. However, as dynamic systems become larger, it is increasingly difficult to undertake such reductions analytically. Further, in this example, if the value of the parameter α is set to zero, then the reduced system contains only one dynamic variable rather than two. That is, the values of $n(f)$ and $n(d)$ change when $\alpha = 0$. Thus, while it is possible to solve this model using the conventional system reduction procedure outlined above, the system (5) changes discontinuously when $\alpha = 0$. Thus, as a practical matter, the reduced system would have to be re-specified for this particular parameter value.

2.3.2 Real Models

The next two related examples concern variants of the neoclassical growth model: these are both models for which a researcher would likely guess that conventional system reduction can be employed, but it works only in the first of the two examples.

The One-Sector Growth Model Log-linear approximation dynamics of the one sector growth model about its steady state lead to the following equations describing a set of "control" variables and point-in-time shadow prices:

$$c_t = (-\frac{1}{\sigma})p_t \tag{12}$$

$$p_t = \lambda_t \tag{13}$$

$$A_t + s_k k_t = s_i i_t + s_c c_t \tag{14}$$

where A_t is the log of date t total factor productivity; c_t is the log of date t consumption; i_t is the log of date t investment; k_t is the log of date t capital stock; p_t is the log of the shadow-price of consumption (or the multiplier on the economy's flow budget constraint);

and λ_t is the log of date t shadow price of capital taken out of the period (or the multiplier on the accumulation equation for capital).[6] The parameters s_i and s_c are the output shares of investment and consumption; s_k is the capital income share; and $(1/\sigma)$ is the intertemporal substitution elasticity in preferences. The model is completed by specifying the dynamic equations for the capital stock and the shadow price of capital:

$$E_t k_{t+1} = (1-\delta)k_t + (i/k)i_t \tag{15}$$

$$\eta_k E_t k_{t+1} + \eta_A E_t A_{t+1} + E_t \lambda_{t+1} = \lambda_t \tag{16}$$

where η_A and η_k are elasticities of the gross marginal product schedule with respect to its two arguments; δ is the depreciation rate on capital and (i/k) is the steady-state investment-capital ratio.

Taken together, (12)-(16) can be written in first-order form as:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & \eta_k \end{bmatrix} E_t \begin{bmatrix} c_{t+1} \\ i_{t+1} \\ p_{t+1} \\ \lambda_{t+1} \\ k_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1/\sigma & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ s_c & s_i & 0 & 0 & -s_k \\ 0 & i/k & 0 & 0 & 1-\delta \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} c_t \\ i_t \\ p_t \\ \lambda_t \\ k_t \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \\ \eta_A \mathbf{F} \end{bmatrix} E_t A_t$$

While the A matrix for the system is singular, the system can be reduced using conventional system reduction since it is in the form (5) with $f_t = (c_t \quad i_t \quad p_t)'$, $d_t = (\lambda_t \quad k_t)'$ and $x_t = A_t$. This is essentially the method that King, Plosser and Rebelo (1988a,b), Rotemberg and Woodford (1992) and many other researchers use to solve models in this class.

The system reduction method described in the next section will implement conventional system reduction automatically, enabling a researcher to simply specify behavioral equations in an arbitrary order and not to distinguish between f_t and d_t. However, in using the algorithm in this way, it is important to again point out that the reduction process is typically not unique. For example, in this one sector growth model, one can select any two of c_t, p_t and λ_t to be f_t elements and the third to be the element of d_t. This indeterminacy is a familiar one, as it is reflected in the differing, but formally equivalent analytical presentations of the global dynamics of the growth model which alternatively use the phase plane in (c, k) space or in (λ, k) space.

A Two Location Variation If we enrich our version of the growth model to include two locations of production of the same final consumption/investment good, then there are two capital stocks k_{1t} and k_{2t}, two investment flows i_{1t} and i_{2t} which augment these capital stocks, two shadow values of capital λ_{1t} and λ_{2t}, and two productivity levels A_{1t} and A_{2t}. We continue to have a single consumption quantity c_t and a single shadow price of consumption p_t.

[6] In these examples, we are considering consumption, investment and output as measured relative to a deterministic trend.

Since this system is a direct generalization of the previous model, we will not write down the full set of matrices that describe the model's dynamics. However, note that the natural way of writing the vector f_t is $f_t = (c_t\, i_{1t}\, i_{2t}\, p_t)'$ and the natural way of writing d_t is $d_t = (\lambda_{1t}\ \lambda_{2t}\ k_{1t}\, k_{2t}\,)'$. This yields a representation of the form (5). However, it impossible to use the conventional system reduction procedure, since the model's equations $0 = B_{ff} f_t + B_{fd} d_t + C_f(F) E_t x_t$ include the pair of equations $p_t = \lambda_{1t}$ and $p_t = \lambda_{2t}$. These equations mean that B_{ff} is singular, rendering the conventional state reduction impossible since it requires inversion of B_{ff}. (The economics behind this singularity is clear: the two investment goods are perfect substitutes from the standpoint of resource utilization so that their shadow prices must be equated if there is a positive amount of each investment, *i.e.*, $\lambda_{1t} = \lambda_{2t}$.)

It is possible to use knowledge of the model's characteristics to avoid this problem, as in Crucini's (1991) two country real business cycle model: one can eliminate one of the two shadow prices "by hand", say λ_{1t}, so as to eliminate the singularity and solve the reduced model with $d_t = (\lambda_{2t}\ k_{1t}\, k_{2t}\,)'$. However, the appearance of the singularity is sometimes more subtle. For example, it occurs in Baxter and Crucini's (1993) two country real business cycle model only when adjustment costs are set equal to zero. Thus, as in the second monetary example, the system changes discontinuously as a function of one the parameters of the model. The algorithm described in the next section carries out system reduction without necessitating prior knowledge of the location of singularities.

The two "growth model" examples illustrate that singularities in A and B can arise naturally in optimizing models. They also illustrate that there is an indeterminacy of system reduction in these models.

3 The Reduction Algorithm

The reduction algorithm begins with the dynamic system (1) and produces a sequence of equivalent dynamic systems ending in the reduced system (4) or in a system that cannot be further reduced but which retains a nonsingular a. In this section, we consider in turn (i) the concept of equivalent dynamic systems; (ii) the steps in the algorithm; and (iii) a proof that any solvable model will result in a reduced dynamic system of the form (4). The reduction algorithm that we construct in this section is closely related to the "shuffle algorithm" developed by Luenberger (1978); Luenberger works directly with the matrices A, B, and C, applying transformations that "shuffle" the rows of the matrices so as to extract an identity linking elements of y, solves the identity and then searches for additional ones.[7] By contrast, we use the singular value decomposition (SVD) and QR factorization to

[7] We thank Adrian Pagan for pointing out the relationship between our work and Luenberger's work on "descriptor systems", which are specifications of the form $Ay_{t+1} = By_t + Cx_t$. Luenberger's terminology makes clear that he has in mind systems for which equilibria are not the solution to a control problem, but rather contain some "descriptive" dynamic equations. While this accords with our view that the analysis of "suboptimal dynamic equilibria" is a major motivation for "singular linear difference systems under rational expectations," it is worthwhile stressing that there are benefits to posing even standard control problems in this manner, as suggested by the two location growth model example.

find groups of identities at a time. Further, in our algorithm, we impose constraints on the admissable transformations that rule out including elements of k in f.

3.1 Equivalent Dynamic Systems

The systems produced by the system reduction algorithm are equivalent in the sense that there is one-to-one transformation connecting them. Specifically, two types of transformations are undertaken.

The first type of transformations involves non-singular matrices T and V, which transform the equations (T) and the variables (V) of the model:

$$T[A\mathbf{F} - B]V^{-1}VEy_t|I_t = TC(\mathbf{F})Ex_t|I_t.$$

These transformations lead to a new dynamic system of the form,

$$[A^*\mathbf{F} - B^*]\ Ey_t^*|I_t = C^*(\mathbf{F})Ex_t|I_t.$$

with $[A^*\mathbf{F} - B^*] = T[A\mathbf{F} - B]V^{-1}$, $C^*(\mathbf{F}) = TC(\mathbf{F})$, and new variables, $Ey_t^*|I_t = V\ Ey_t|I_t$. Since T and V are non-singular, the transformation produces an equivalent system. Further, both the new system and the original system have the same roots to their determinantal polynomials, since $|A^*z - B^*| = |T||Az - B||V^{-1}|$. The transformations that we consider are very simple for variables and relatively sophisticated for equations. In particular, V is restricted to simply reorder the variables rather than taking general linear combinations. This restriction makes it easy to interpret the reduced dimension dynamic system in terms of observable variables.

The second set of transformations that the algorithm employs are transformations summarized above as conventional system reduction. Section (2.2) showed how this transformation could be summarized by multiplying the model's equations by a specific polynomial $T(\mathbf{F}) = G\mathbf{F} + H$, with two key properties. First, $|T(z)| = \tau \neq 0$, so that again, this transformation does not alter the roots of the model's determinantal equation. Second, the matrices G and H are chosen so that the system remains first order.

At each iteration of the algorithm, we begin with a dynamic system with the following characteristics: we have already identified some elements of y that will are elements of f , which we call $\widetilde{f}$ and we also have the remaining elements of y which we call $\widetilde{d}$. The transformations that have been previously undertaken have not altered the fact that the predetermined variable k are the last elements of the transformed y vector, which we call $\widetilde{y}$; this also implies that the initial elements of $\widetilde{d}$ are nonpredetermined variables, which we call $\widetilde{\lambda}$. Overall, we have a system of the form:

$$\widetilde{A}E\widetilde{y}_{t+1}|I_t = \widetilde{B}\widetilde{y}_t + \widetilde{C}(\mathbf{F})Ex_t|I_t, \tag{17}$$

where

$$\widetilde{A} = \begin{bmatrix} 0 & 0 \\ 0 & a \end{bmatrix}, \widetilde{y} = \begin{bmatrix} \widetilde{f} \\ \widetilde{d} \end{bmatrix}, \widetilde{B} = \begin{bmatrix} I & \square \\ 0 & b \end{bmatrix}, \widetilde{C}(\mathbf{F}) = \begin{bmatrix} C_{\widetilde{f}}(\mathbf{F}) \\ C_{\widetilde{d}}(\mathbf{F}) \end{bmatrix}.$$

Technically, the vector $\widetilde{y} = \widetilde{V} y$, where $\widetilde{V}$ is a permutation matrix determined in previous iterations, and $\widetilde{f}$ denotes the flow variables determined in previous iterations. The vector $\widetilde{d}$ is partitioned as $\widetilde{d} = (\widetilde{\lambda}' \quad k')'$, so that the predetermined variables k appear as the final will be the final n_k elements. The iteration then proceeds by moving some of the elements of $\widetilde{\lambda}$ into $\widetilde{f}$. The algorithm terminates when this is no longer possible. As we show below, the resulting value of a will be non-singular if the original system satisfies the restriction $|Az - B| \neq 0$ and there exists a solution to the model.

3.2 The Algorithm Steps

We now outline each of the five steps of the algorithm.

Step 1: Initialization

We begin by setting $\widetilde{y} = \widetilde{d} = y$, $\widetilde{A} = a = A$, $\widetilde{B} = b = B$, and $\widetilde{C}(\mathbf{F}) = C_{\widetilde{d}}(\mathbf{F}) = C(\mathbf{F})$, so that $\widetilde{f}$ is absent from the model, $n(\widetilde{d}) = 0$ and $n(\widetilde{d}) = n(y)$. If a is non-singular, proceed to step 5, otherwise proceed to step 2.

Step 2: Uncovering New Candidate Flows

We first use the singular value decomposition to transform a so that there are some rows of zeros, i.e., that there are some nondynamic equations in the transformed system. Let the singular value decomposition of a be $a = U * S * V^T$, with $U * U^T = I$, $V * V^T = I$, and S is a diagonal matrix with the (nonnegative) singular values on the diagonal: these singular values are ordered in diminishing size, so that $s_i \geq s_j$ for i □ j. Let n_s denote the rank of a (equivalently the number of non-zero singular values, s_i), and let J be the row-switching matrix of the same size as a, which places the n_s rows corresponding to non-zero singular values last

$$J = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}$$

Multiplying the full system (17), which includes the elements of $\widetilde{f}$, by

$$T_1 = \begin{bmatrix} I & 0 \\ 0 & JU^T \end{bmatrix},$$

produces a transformed version of (17), $T_1 \widetilde{A} E \widetilde{y}_{t+1} | I_t = T_1 \widetilde{B} \widetilde{y}_t + T_1 \widetilde{C}(\mathbf{F}) E x_t | I_t$. By construction, the dynamic component of this new system has the form:

$$\begin{bmatrix} 0 & 0 \\ a_{2\lambda} & a_{2k} \end{bmatrix} E_t \begin{bmatrix} \widetilde{\lambda}_{t+1} \\ k_{t+1} \end{bmatrix} = \begin{bmatrix} b_{1\lambda} & b_{1k} \\ b_{2\lambda} & b_{2k} \end{bmatrix} \begin{bmatrix} \widetilde{\lambda}_t \\ k_t \end{bmatrix} + \begin{bmatrix} \Psi_1(\mathbf{F}) \\ \Psi_2(\mathbf{F}) \end{bmatrix} E_t x_t. \tag{18}$$

Let n_{cf} denote the number of equations in the blocks with a leading subscript 1 (the subscript "cf" denotes candidate flows). We now want to solve for some elements of $\widetilde{\lambda}_t$ as flows, which is done in next two steps.

Step 3: Isolating New Flows

Focusing on the first block of equations of (18): $b_{1\lambda}\widetilde{\lambda_t} + b_{1k}\ k_t = \Psi_1(\mathbf{F})E_t x_t$, we first determine the number of linearly independent restrictions on $\widetilde{\lambda}_t$ that these represent (which we call n_{nf} for number of new flows). We also want to transform the equations and variables so as to facilitate solution of the equations. This is easily accomplished using the QR factorization of $b_{1\lambda}$: $QR = b_{1\lambda}P$.[8] Since P is an $(n_{cf} \times n_{cf})$ permutation matrix, this factorization implies that we can reorder the elements of $\widetilde{\lambda}_t$ by multiplying by P^T. Thus, for the full vector $\widetilde{y}$, the reordering is produced by:

$$V_1 = \begin{bmatrix} I & 0 & 0 \\ 0 & P^T & 0 \\ 0 & 0 & I \end{bmatrix}$$

That is, we rewrite the dynamic system $T_1\widetilde{A}E\widetilde{y}_{t+1}|I_t = T_1\widetilde{B}y_t + T_1\widetilde{C}(\mathbf{F})Ex_t|I_t$ as $T_1\widetilde{A}EV_1^{-1}V_1\widetilde{y}_{t+1}|I_t = T_1\widetilde{B}V_1^{-1}V_1\widetilde{y}_t + T_1\widetilde{C}(\mathbf{F})Ex_t|I_t$. While reordering $\widetilde{\lambda}$, V_1 preserves the ordering of $\widetilde{f}$and k. We also want to reorganize the system's equations using the QR factorization of $b_{1\lambda}$

$$T_2 = \begin{bmatrix} I & 0 & 0 \\ 0 & Q^T & 0 \\ 0 & 0 & I \end{bmatrix}$$

Hence, the transformations of the system through step 2 are: $T_2T_1\widetilde{A}EV_1^{-1}V_1\widetilde{y}_{t+1}|I_t = T_2T_1\widetilde{B}V_1^{-1}V_1y_t + T_2T_1\widetilde{C}(\mathbf{F})Ex_t|I_t$.

Step 4: Conventional System Reduction

It now follows that there are $n_{nf} = rank(b_{1\lambda})$ new flows for which we may solve. Put alternatively, if $n(\widetilde{f})$ is the number of pre-existing flows then the $(n(\widetilde{f})+n_{nf}) \times (n(\widetilde{f})+n_{nf})$ leading submatrix of $T_2T_1\widetilde{B}V_1^{-1}$ is of the form

$$B_{ff} = \begin{bmatrix} I & 0 \\ 0 & R_{11} \end{bmatrix}$$

where R_{11} is the (n_{nf}) by (n_{nf}) submatrix of the R matrix in the QR factorization of $b_{1\lambda}$. The matrix is B_{ff} is clearly invertible (since R_{11} is invertible) and, hence, we can employ the

[8] The QR factorization of a $p \times m$ matrix M with rank r is given by

$$QR = MP \tag{19}$$

where P is an m by m permutation matrix (*i.e.,* P can be constructed by permuting the rows of I_m); Q is a p by p unitary matrix; and R is a p by m matrix such that

$$R = \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix}$$

where R_{11} is a r by r upper diagonal, nonsingular matrix and R_{12} is a r by $(m-r)$ matrix.

The QR factorization is useful because it allows us to solve the equation system $My = m$ for r of the elements of y in terms of the remaining $(m-r)$ elements of y and the parameters m. That is, we can write the equation $My = m$ as $RP'y = Q'm$ and partition $P'y = [y_1'y_2']'$. Then, the solution is $y_1 = R_{11}^{-1}(\gamma_1 - R_{12}y_2)$, where γ_1 is the first r rows of $Q'm$. The equation system is consistent only if $\gamma_2 = 0$, where γ_2 is the last $(m-r)$ rows of $Q'm$.

conventional system reduction procedure detailed in section (2.2). This yields a new system in the form of (17), but with a smaller matrix a.

Step 5: Terminating the Iterative Scheme

If the resulting new value a is singular, then the sequence of steps 2-4 is repeated. Otherwise, we have accomplished our goal: eliminating the "~" superscripts from the variables, the resulting dynamic system is $aEd_{t+1}|I_t = bd_t + C_d(\mathbf{F})Ex_t|I_t$, which can be rewritten as $Ed_{t+1}|I_t = Wd_t + \Psi_d(\mathbf{F})Ex_t|I_t$ with $W = a^{-1}b$ and $\Psi_d(\mathbf{F}) = a^{-1}C_d(\mathbf{F})$.

3.3 Convergence of the algorithm

We now are in a position to discuss the conditions under which the preceding algorithm is guaranteed to produce a non-singular reduced system of the form (4) in a finite number of steps. For this convergence, we require only two conditions: (i) that there is a non-zero determinant, $|Az - B| \neq 0$; and (ii) that for every set of initial conditions (each vector k_0) there exists a solution, i.e., there exists a stochastic process $\{y_t\}_{t=0}^{\infty}$ such that the equations of the original model are satisfied at each date.

The first of these conditions is verifiable: we can check it prior to starting on the algorithm. The second of these conditions is an assumption. From our companion research, King and Watson (1997), we could spell out restrictions on the dynamic model under which there exists a unique, stable solution. In appendix B, we show that such a model always has a feasible system reduction. However, in the current context, we want to admit a wider range of possibilities: that there could be more than one solution (as in Farmer (1993)) or an unstable solution of economic interest (as in McCallum (1983)): all that matters is that the existence of a solution is postulated.

Each iteration of the algorithm involves application of a matrix polynomial $T(\mathbf{F})$ with nonzero determinant τ to the dynamic system: this operation does not alter either condition (i) since $|Az-B| \neq 0$ implies $|T(z)||Az-B| = \tau|Az-B| \neq 0$ or condition (ii) since if $\{y_t\}_{t=1}^{\infty}$ is a solution to the original model it is also a solution to the transformed model. Further, each iteration eliminates some elements of y from d and moves them into f so long as it is possible to construct another $T(\mathbf{F})$. With one or more elements moved on each iteration (typically many more than one), the algorithm must converge in less than $n(y)$ steps.

The successful construction of a $T(\mathbf{F})$ at each iteration of the algorithm will be possible unless $n_{nf} = 0$ in step 3. Since $n_{nf} = rank(b_{1\lambda})$, this implies that $b_{1\lambda} = 0$. That is, there is an equation of the transformed model that is:

$$b_{1\lambda}\widetilde{\lambda_t} + b_{1k}k_t = 0\widetilde{\lambda_t} + b_{1k}k_t = \Psi_1(\mathbf{F})E_t x_t \tag{20}$$

If in addition $b_{1k} = 0$, then there is a row of zeros in $\widetilde{A}z - \widetilde{B}$ and, hence, $0 = |\widetilde{A}z - \widetilde{B}| = |Az-B|$. In this equality, the first equality follows from the implied rows of zeros in $T_1\widetilde{A}$ and $T_1\widetilde{B}$, and the second equality follows because the roots of the determinantal equation are unaltered by any of the transformations used in the reduction algorithm. Thus, condition (i) is violated in this case. However, since condition (i) is a verifiable property of the model, we may use this to rule out $b_{1k} = 0$ if $b_{1\lambda} = 0$.[9]

On the other hand, if $b_{1\lambda} = 0$ and $b_{1k} \neq 0$, then it is possible to solve for a subset of the predetermined variables k_t as a function of the other predetermined variables and a distributed lead of the expected values of the x_t's. This outcomes violates our condition (ii), since it imposes constraints on the initial conditions for the predetermined variables. Hence, under conditions (i) and (ii), there cannot be an equation of the form (20). As a result, each successive iteration of the algorithm eliminates some new flows and ultimate convergence is guaranteed.

There is thus also a useful interpretation of the failure of the algorithm in step 3 if it occurs for a specific model: if the algorithm fails in this way when we have verified that $|Az - B| \neq 0$, then this failure is evidence that there does not exist a solution of the model.

4 Solving the Model

After application of the system reduction algorithm, the transformed dynamic system takes the form:

$$f_t = -[K_{f\lambda}\; K_{fk}] \begin{bmatrix} \lambda_t \\ k_t \end{bmatrix} - \Psi_f(\mathbf{F}) E x_t | I_t \tag{21}$$

$$E_t \begin{bmatrix} \lambda_{t+1} \\ k_{t+1} \end{bmatrix} = \begin{bmatrix} W_{\lambda\lambda} & W_{\lambda k} \\ W_{k\lambda} & W_{kk} \end{bmatrix} \begin{bmatrix} \lambda_t \\ k_t \end{bmatrix} + \begin{bmatrix} \Psi_\lambda(\mathbf{F}) \\ \Psi_k(\mathbf{F}) \end{bmatrix} E_t x_t \tag{22}$$

We discuss the solution of this model in three steps. First, we determine the implications of stability for the initial conditions on λ_t. Second, we consider the construction of perfect foresight solutions. Third, we consider the construction of dynamic decision rules.

4.1 Initial Conditions on λ_t

Our treatment of the determination of the initial conditions on λ_t follows Blanchard and Kahn (1980). To begin, we let V_u be a ($n_u$by n_d) matrix that isolates the unstable roots of W. That is, V_u has the property that

$$V_u W = \mu V_u,$$

where μ is a lower triangular (n_u by n_u) matrix with the unstable eigenvalues on its diagonal. Applying V_u to (22), $V_u\, E_t d_{t+1} = V_u\, W d_t + V_u\, \Psi_d(\mathbf{F}) E_t x_t = \mu\, V_u\, d_t + V_u\, \Psi_d(\mathbf{F}) E_t x_t$.[10] The dynamics of u are then:

[9] Luenberger (1978) uses an argument like this one to establish the inevitable convergence of his "shuffle" algorithm when $|Az - B| \neq 0$.

[10] One interpretation of this transformation is that $u_t = V_u d_t$ is the vector of unstable canonical variables and that V_u is the matrix of left eigenvectors that corresponds to the unstable eigenvalues; in this case, μ is a Jordan matrix with the entries below the main diagonal corresponding to repeated unstable roots. However, a better computational method is to use the Schur decomposition. In this case μ is a lower triangular matrix with the unstable eigenvalues on the diagonal.

$$E_t u_{t+1} = \mu \ u_t + V_u \Psi_d(\mathbf{F}) E_t x_t \tag{23}$$

This expression can be used to generate formulae for (a) perfect foresight solutions; and (b) Markov decision rules. In each case, u_t is chosen so that there is a stable system despite the existence of unstable roots. Following Sargent (1979) and Blanchard and Kahn (1980), this is accomplished by unwinding unstable roots forward.

Once we have solved for the elements of u, this information is used to uniquely determine the date t behavior of the non-predetermined variables λ_t. As stressed by Blanchard-Kahn (1980) this requires that there be the same number of number of unstable canonical variables as there are non-predetermined variables. In terms of the condition $u_t = V_u d_t = V_{u\lambda}\lambda_t + V_{uk}k_t$, this requirement means that we have an equal number of equations and unknowns. However, it is also the case that a unique solution mandates that the (square) matrix $V_{u\lambda}$ is of full rank. This condition on $V_{u\lambda}$ is an implication of the more general rank condition that Boyd and Dotsey (1993) give for higher order linear rational expectations models. Essentially, the rank condition rules out matrices W with unstable roots associated with the predetermined variables rather than nonpredetermined variables.

4.2 Perfect Foresight (Sequence) Solutions

Perfect foresight solutions, *i.e.*, the response of the economy to a specified sequence of x : $\{x_s\}_{s=t}^{\infty}$, are readily constructed. Expression (23) implies that:

$$u_t = -[I - \mu^{-1}\mathbf{F}]^{-1}[\mu^{-1} V_u \Psi_d(\mathbf{F})] E_t x_t \tag{24}$$

which provides a way of evaluating the expression $u_t = \Psi_u(\mathbf{F}) E_t x_t$. This equation, together with $u_t = V_{u\lambda}\lambda_t + V_{u\lambda}k_{t,}$ implies

$$\lambda_t = -V_{u\lambda}^{-1} V_{uk} k_t + V_{u\lambda}^{-1} \Psi_u(\mathbf{F}) E_t x_t. \tag{25}$$

With knowledge of λ_t, expressions (21) and (22) imply that:

$$f_t = -[K_{fk} - K_{f\lambda} V_{u\lambda}^{-1} V_{uk}] k_t - [K_{f\lambda} V_{u\lambda}^{-1} \Psi_u(\mathbf{F}) + \Psi_f(\mathbf{F})] E_t x_t \tag{26}$$

$$k_{t+1} = M_{kk} k_t + [W_{k\lambda} V_{u\lambda}^{-1} \Psi_u(\mathbf{F}) + \Psi_k(\mathbf{F})] E_t x_t \tag{27}$$

where $M_{kk} = (W_{kk} - W_{k\lambda} V_{u\lambda}^{-1} V_{uk})$. Under perfect foresight, $\mathbf{F}^n x_t = x_{t+n}$. These solutions are of interest in their own right and also as inputs into the construction of Markov decision rules, which we consider next.

4.3 Markov Decision Rules

Now suppose that x_t is generated by the process

$$x_t = \Theta \xi_t$$

and

$$\xi_t = \rho\xi_{t-1} + \theta\eta_t$$

where η_t is a martingale difference sequence. Given the first-order character of the driving process, ξ_t is sufficient to describe the distributed leads such as $\Psi_u(\mathbf{F})E_t x_t$, etc.

Consider first the evaluation of $\Psi_d(\mathbf{F})E_t x_t$, and $\Psi_f(\mathbf{F})E_t x_t$. Each of these is a polynomial of the form $\Psi(\mathbf{F}) = \Psi_0 + \Psi_1\mathbf{F} + \Psi_2\mathbf{F}^2 + ... + \Psi_n\mathbf{F}^n$. With $E_t x_{t+h} = \Theta\rho^h\xi_t$, it follows that

$$\Psi(\mathbf{F})E_t x_t = [\Psi_0\Theta + \Psi_1\Theta\rho + \Psi_2\Theta\rho^2 + ... + \Psi_n\Theta\rho^n]\ \xi_t \equiv \varphi\ \xi_t \tag{28}$$

Thus, for example it follows that $V_u\Psi_d(\mathbf{F})E_t x_t = V_u[\Psi_{d0}\Theta + \Psi_{d1}\Theta\rho + \Psi_{d2}\Theta\rho^2 + ... + \Psi_{dn}\Theta\rho^n]\xi_t = \varphi_u\xi_t$.

There now is an operational difference depending on whether we assume that μ is diagonal or simply lower triangular. In the first case, which is that typically employed, we proceed as follows. Letting φ_{ui} be the i'th row of φ_u, it follows that the unstable canonical variables each evolve according to equations of the form:

$$E_t u_{i,t+1} = \mu_i u_{it} + \varphi_{ui}\xi_t. \tag{29}$$

If we iterate this expression forward, we get:

$$u_{it} = -\sum_{j=0}^{\infty}\mu_i^{-j-1}\varphi_{ui}E_t\xi_{t+j} = -[\sum_{j=0}^{\infty}\mu_i^{-j-1}\varphi_{ui}\rho^j]\xi_t = \varphi_{ui}[\rho - \mu_i I]_t^{-1}\xi.$$

For convenience write this solution as $u_{it} = \nu_i\xi_t$, with $v_i = \varphi_{ui}[\rho - \mu_i I]^{-1}$ and correspondingly let $u_t = \nu\xi_t$, with the i'th row of ν being ν_i. (Alternatively, we may think of using an undetermined coefficients representation, $u_{it} = \nu_i\xi_t$ in $E_t u_{i,t+1} = \mu_i u_{it} + \varphi_{ui}\xi_t$ to find that $\nu_i\rho = \mu_i\nu_i I + \varphi_{ui}$: this alternative derivation will be useful in discussing the general lower triangular μ case below.) Thus, we have obtained a solution for u_t, which can be used to solve for λ_t.

In the general case with μ lower triangular (*e.g.,* when the Schur decomposition is used to construct V_u and μ), let the *ith* row of μ be denoted by $[\mu_{i1}...\mu_{ii}\ 0...0]$. The expression analogous to (29) is thus:

$$E_t u_{i,t+1} = \sum_{j=1}^{i}\mu_{ij}\ u_{jt} + \varphi_{ui}\xi_t$$

It thus follows that the first of these expressions can be solved exactly as previously $u_{1t} = \nu_1\xi_t$, with $\nu_1 = \varphi_{u1}[\rho - \mu_{11}I]^{-1}$. Given this solution, it follows that $u_{2t} = \nu_2\xi_t$, with $\nu_{2i} = (\varphi_{ui} + \mu_{21}\nu_1)[\rho - \mu_{22}I]^{-1}$. That is, we can always write the above expression as

$$\begin{aligned} E_t u_{i,t+1} &= \mu_{ii}u_{it} + \sum_{j=1}^{i-1}\mu_{ij}\ u_{jt} + \varphi_{ui}\xi_t \\ &= (\sum_{j=1}^{i-1}\mu_{ij}\eta_j + \varphi_{ui})\xi_t \end{aligned}$$

and solve it by employing the same undetermined coefficients strategy as in the previous case. Hence, there is an easy-to-implement way of calculating $u_t = \nu\xi_t$, under the Schur decomposition as well.

Since we know that $u_t = \nu\xi_t$, we can determine that:

$$\lambda_t = -V_{u\lambda}^{-1}V_{uk}k_t + V_{u\lambda}^{-1}\nu\xi_t \tag{30}$$

With knowledge of this solution for λ_t, expressions (21) and (22) imply that:

$$f_t = [K_{fk} - K_{f\lambda}V_{u\lambda}^{-1}V_{uk}]k_t + [K_{f\lambda}V_{u\lambda}^{-1}\nu + \varphi_f]\xi_t \tag{31}$$

$$k_{t+1} = M_{kk}k_t + [W_{k\lambda}V_{u\lambda}^{-1}V_{uk}\nu + \varphi_k]\xi_t \tag{32}$$

where $M_{kk} = (W_{kk} - W_{k\lambda}V_{u\lambda}^{-1}V_{uk})$ as above and φ_f and φ_k are the evaluations of the Ψ_f and Ψ_k polynomials (as in (28)).

These solutions can be grouped together with other information to produce a state space system, i.e., a pair of specifications $Z_t = \Pi S_t$ and $S_{t+1} = M\ S_t + \varepsilon_{t+1}^*$. The specific form is:

$$\begin{bmatrix} f_t \\ \lambda_t \\ k_t \\ x_t \end{bmatrix} = \begin{bmatrix} [K_{fk} - K_{f\lambda}V_{u\lambda}^{-1}V_{uk}] & [KV_{u\lambda}^{-1}\ \nu + \varphi_f] \\ -V_{u\lambda}^{-1}V_{uk} & V_{u\lambda}^{-1}\ \nu \\ I & 0 \\ 0 & \Theta \end{bmatrix} \begin{bmatrix} k_t \\ \xi_t \end{bmatrix} \tag{33}$$

$$\begin{bmatrix} k_{t+1} \\ \xi_{t+1} \end{bmatrix} = \begin{bmatrix} M_{kk} & [W_{k\lambda}V_{u\lambda}^{-1}\nu + \varphi_k] \\ 0 & \rho \end{bmatrix} \begin{bmatrix} k_t \\ \xi_t \end{bmatrix} + \begin{bmatrix} 0 \\ \theta \end{bmatrix} \eta_{t+1} \tag{34}$$

Further, given $\begin{bmatrix} f_t \\ d_t \end{bmatrix} = Ly_t$, we can recover a representation in terms of the original ordering of the variables.

5 Conclusions

In this paper, we described (i) an algorithm for the reduction of singular linear difference systems under rational expectations and (ii) algorithms for the solution of the resulting reduced dimension system. This set of methods is flexible and easy to use.

References

[1] Baxter, Marianne and Mario J. Crucini, "Explaining Saving-Investment Correlations," *American Economic Review*, vol. 83, no. 3 (June 1993), 416-436.

[2] Blanchard, Olivier J. and Charles Kahn, "The Solution of Linear Difference Models Under Rational Expectations," *Econometrica,* vol. 48, no. 5 (July 1980): 1305-1311.

[3] Boyd, John H. III and Michael Dotsey, "Interest Rate Rules and Nominal Determinacy," Federal Reserve Bank of Richmond, working paper, August 1993, revised February 1994.

[4] Crucini, Mario, "Transmission of Business Cycles in the Open Economy," PhD dissertation, University of Rochester, 1991.

[5] Farmer, Roger E.A.*, The Macroeconomics of Self-Fulfilling Prophecies*, MIT Press, 1993.

[6] Golub G.H. and C.F. Van Loan (1989), *Matrix Computations*, 2nd Edition, Baltimore: Johns Hopkins University Press.

[7] King, R.G., C.I. Plosser and S.T. Rebelo, (1988a), "Production, Growth and Business Cycles, I: The Basic Neoclassical Model," *Journal of Monetary Economics*, vol 21, no. 2/3 (March/May) 1988, 195-232.

[8] King, R.G., C.I. Plosser and S.T. Rebelo, (1988b), "Production, Growth and Business Cycles, Technical Appendix," University of Rochester working paper.

[9] King, R.G. and M.W. Watson, "Money, Interest Rates, Prices and the Business Cycle," *Review of Economics and Statistics*, February 1996.

[10] King, R.G. and M.W. Watson, "The Solution of Singular Linear Difference Systems Under Rational Expectations," working paper, 1997.

[11] King, R.G. and M.W. Watson, "Research Notes on Timing Models," 1995.

[12] Lucas, Robert E., Jr., "Expectations and the Neutrality of Money," *Journal of Economic Theory,* vol. 4 (April 1972), pages.

[13] Luenberger, David, "Dynamic Equations in Descriptor Form," IEEE Transactions on Automatic Control, vol. AC-22, no. 3 (June 1977), 312-321.

[14] Luenberger, David, "Time Invariant Descriptor Systems," *Automatica*, vol. 14, no. ?? (month 1978), 473-480.

[15] Luenberger, David, *Introduction to Dynamic Systems: Theory, Models and Applications*, New YorK: John Wiley and Sons, 1979.

[16] McCallum, Bennett T., "Non-Uniqueness in Rational Expectations Models: An Attempt at Perspective," *Journal of Monetary Economics*, vol. 11, no. 2 (March 1983), 139-168.

[17] Moler, Cleve B. and G.W. Stewart, "An Algorithm for Generalized Matrix Eigenvalue Problems," *SIAM Journal of Numerical Analysis*, vol. 10, no. 2 (April 1973), 241-256.

[18] Rotemberg, Julio and Michael Woodford, "Oligopolistic Pricing and the Effects of Aggregate Demand on Economic Activity," *Journal of Political Economy*, vol. 100, no. 6 (December 1992), 1153-1207.

[19] Sargent, Thomas J., *Macroeconomic Theory*, New York: Academic Press, 1979.

[20] Sargent, Thomas J. and Neil Wallace, "Rational Expectations, the Optimal Money Supply Instrument and the Optimal Money Supply Rule," *Journal of Political Economy*, vol. 83, no. ?? (month 1975), 241-254.

[21] Sims, Christopher A., "Solving Non-Linear Stochastic Optimization Problems Backwards," discussion paper 15, Institute for Empirical Macroeconomics, FRB Minneapolis, May 1989.

[22] Svensson, Lars E.O., "Money and Asset Prices in a Cash In Advance Economy," *Journal of Political Economy*, vol. 93, no. ?? (month 1985), 919-944.

A Model Solution Programs

We now discuss the structure of our model solution programs (our discussion is terms of the MATLAB versions; the GAUSS programs are essentially similar, except for some minor differences based on the diverse ways in which MATLAB and GAUSS handle outputs of subroutines). Since many readers of this paper will be familiar with the "KPR" MATLAB model solution code of King, Plosser, and Rebelo (1988), we will discuss the relationship of our components to this code.

control.m:this program essentially allows for "batch execution" of the remainder of the programs, running them sequentially as specified below. In addition to executing the programs, control.m also includes some "checks of specification" that are described later.

sskw.m: this program sets up the initial model in the form of (1), i.e., $AEy_{t+1}|I_t = By_t + C(\mathbf{F})Ex_t|I_t$. This frequently involves determining the "steady state" of the dynamic model and using this information to specify the matrices A, B, and $C(\mathbf{F})$. It also includes the specification of which variables are predetermined. It is thus analogous to the ss.m program in the KPR code. After sskw.m is run by control.m, it is possible to perform two checks of specification. One of these is a check of the necessary condition $|Az - B| \neq 0$, which involves graphing the determinant for different values of z. The other check is the "long-run" properties of the model, which are calculated as $((A-B)^{-1}C(1))$.

redkw.m: this program reduces the dynamic system in the form (1) to a reduced form using the algorithm described above. For this purpose, it makes use of a subroutine **csr.m,** which undertakes conventional system reduction. Then the output is a system in the form (17), i.e.,

$$\begin{bmatrix} 0 & 0 \\ 0 & a \end{bmatrix} E_t \begin{bmatrix} f_{t+1} \\ d_{t+1} \end{bmatrix} = \begin{bmatrix} I & \square \\ 0 & b \end{bmatrix} \begin{bmatrix} f_t \\ d_t \end{bmatrix} + \begin{bmatrix} \Psi_f(\mathbf{F}) \\ \Psi_d(\mathbf{F}) \end{bmatrix} E_t x_t.$$

If the dynamic system is well specified, the matrix a is of full rank. If it is not, then a is of less than full rank (and the program indicates this).

dynkw.m: The well-specified dynamic is then placed in the form (4) and the dynamic subsystem $Ed_{t+1}|I_t = Wd_t + \Psi_d(\mathbf{F})Ex_t|I_t$ is isolated. The researcher can choose either an eigenvalue-eigenvector method or a Schur decomposition method. A subroutine **lus.m** computes the matrix called V_u in section 4 above, which is the matrix of left eigenvectors corresponding to the unstable eigenvalues or its Schur generalization. For the model to be solved uniquely, the submatrix $V_{u\lambda}$ which governs how the u_t respond to the non-predetermined variables λ_t must be square and of full rank. This condition is checked and the program terminates if $V_{u\lambda}$ is rank-deficient.

mdrkw.m: Markov decision rules are then calculated of section 5. The outputs are a dynamic system in the state space form:

$$z_t \equiv \begin{bmatrix} y_t \\ x_t \end{bmatrix} = \begin{bmatrix} \Pi_{yk} & \Pi_{y\xi} \\ 0 & Q \end{bmatrix} \begin{bmatrix} k_t \\ \xi_t \end{bmatrix}$$

$$\begin{bmatrix} k_{t+1} \\ \xi_{t+1} \end{bmatrix} = \begin{bmatrix} M_{kk} & M_{k\xi} \\ 0 & I \end{bmatrix} \begin{bmatrix} k_t \\ \xi_t \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} \eta_{t+1}$$

In using the MATLAB code that we have developed, essentially the only programs that the user needs to edit are (i) sskw.m and (ii) mdrkw.m. The first specifies the dynamic system and the second specifies the forcing process.

There are four directories on the distribution diskette that may be included with this paper or is available at http:\\ on the internet. The `\examples` directory contains a central control.m program which runs the four examples of section 4 above and the two examples with predetermined variables. The `\crazy` directory contains the reduction code and several examples of models that cannot be reduced.

For more realistic examples, the reader may wish to examine the contents of the `\models` directory and its subdirectories. Two of these contain a real business cycle model and sticky price model that we have studied in some recent research (King and Watson (1996)). These are run by a control.m program and can be used to plot model impulse responses.

Given the state-space system that results from each of these models, it is easy to compute a range of outputs. For example, **impkw.m** computes the impulse responses for the z variables. It is also straightforward to compute moments of filtered y using frequency domain methods, frequency domain representations of the model, etc.

A System Reduction of Uniquely Solvable Models

In a companion paper (King and Watson (1997)), we developed the conditions under which singular linear difference system can be uniquely solved under rational expectations. There were two necessary and sufficient conditions. First, we required that the model satisfied a determinant condition, $|Az - B| \neq 0$, also imposed in this paper. Second, we required that a particular matrix $V_{U\Lambda}$ had full rank: this matrix requirement was necessary to link non-predetermined variables Λ to unstable canonical variables U, where our definition of unstable canonical variables contained both finite and infinite eigenvalues..

Our starting point in this appendix is the solvable dynamic system written in a canonical form developed in that earlier paper,

$$(A^*\mathbf{F} - B^*)V \ E_t y_t = C^* E_t x_t,$$

where

$$[A^*\mathbf{F} - B^*] = \begin{bmatrix} (N\mathbf{F} - I) & 0 \\ 0 & (\mathbf{F}I - J) \end{bmatrix}.$$

with N being a nilpotent matrix.[11] Appendix A to King and Watson (1997) shows that such an equivalent system can always be constructed for a solvable model. The transformed

[11] A nilpotent matrix has zero elements on and below its main diagonal, with zeros and ones appearing abitrarily above the diagonal. Accordingly, there is a number l such that N^l is a matrix with all zero elements.

variables in this canonical form are

$$\begin{bmatrix} U_t \\ s_t \end{bmatrix} \equiv \begin{bmatrix} i_t \\ u_t \\ s_t \end{bmatrix} = V \; E_t y_t$$

with the partitioning defined by the magnitude of the eigenvalues. First, there are as many elements of s as there are stable eigenvalues of J. Second, there are as many elements of i_t as there are columns of N. As explained in King and Watson (1997), these are canonical variables associated with 'infinite' eigenvalues.

Premultiplication of the canonical dynamic system by

$$\eta(\mathbf{F}) = \begin{bmatrix} (N\mathbf{F} - I)^{-1} & 0 \\ 0 & I \end{bmatrix}$$

results in

$$\begin{bmatrix} I & 0 \\ 0 & (\mathbf{F}I - J) \end{bmatrix} V \; E_t y_t = \eta(\mathbf{F}) C^* E_t x_t.$$

Now, partition the transformation of variables matrix V as follows

$$V = \begin{bmatrix} V_{if} & V_{id} \\ V_{\delta f} & V_{\delta d} \end{bmatrix} = \begin{bmatrix} V_{if} & V_{i\lambda} & V_{ik} \\ V_{uf} & V_{u\lambda} & V_{uk} \\ V_{sf} & V_{s\lambda} & V_{sk} \end{bmatrix},$$

and let

$$V_{U\Lambda} = \begin{bmatrix} V_{if} & V_{i\lambda} \\ V_{uf} & V_{u\lambda} \end{bmatrix}.$$

(Recall that λ corresponds to the components of Λ that will ultimately be part of the dynamic vector d).

Without loss of generality, we will assume that the elements of y are ordered so that V_{if} is square and nonsingular. (This is always possible given that $V_{U\Lambda}$ is nonsingular).

Using the first of these partitions, the dynamic system can be written

$$\begin{bmatrix} V_{if} & V_{id} \\ (\mathbf{F}I - J)V_{\delta f} & (\mathbf{F}I - J)V_{\delta d} \end{bmatrix} E_t y_t = \eta(\mathbf{F}) C^* E_t x_t.$$

Construction of Reduced System

We now proceed to construct a reduced dimension dynamic system. To accomplish this, we premultiply by a matrix polynomial $T(\mathbf{F})$ that is partitioned as

$$\begin{bmatrix} T_{11}(\mathbf{F}) & T_{12}(\mathbf{F}) \\ T_{21}(\mathbf{F}) & \mathbf{F}_{22}(\mathbf{F}) \end{bmatrix}$$

We design the components of this matrix polynomial as follows:

$$T_{11}(\mathbf{F}) V_{if} + T_{12}(\mathbf{F})(\mathbf{F}I - J) V_{\delta f} = I$$

$$T_{11}(\mathbf{F})V_{id} + T_{12}(\mathbf{F})(\mathbf{F}I - J)V_{\delta d} = K$$

$$T_{21}(\mathbf{F})V_{if} + T_{22}(\mathbf{F})(\mathbf{F}I - J)V_{\delta f} = 0$$

$$T_{21}(\mathbf{F})V_{id} + T_{22}(\mathbf{F})(\mathbf{F}I - J)V_{\delta d} = [\mathbf{F}I - W]$$

where K and W are unknown matrices.

The first two of these conditions may be used to set

$$T_{11}(\mathbf{F}) = V_{if}^{-1}$$

$$T_{12}(\mathbf{F}) = 0$$

which jointly imply that $K = V_{if}^{-1}V_{id.}$

Employing the third and fourth condition together, we find that

$$T_{22}(\mathbf{F})\{(\mathbf{F}I - J)[V_{\delta d} - V_{\delta f}V_{if}^{-1}V_{id}]\} = \mathbf{F}I - W$$

Standard results on the determinants and inverses of partitioned matrices imply that $\hat{V} = [V_{\delta d} - V_{\delta f}V_{if}^{-1}V_{id}]$ is an invertible matrix. Accordingly, this condition may be satisfied with

$$T_{22}(\mathbf{F}) = \hat{V}^{-1}$$

and

$$W = \hat{V}^{-1}J\hat{V}.$$

A further implication is then that

$$T_{21}(\mathbf{F}) = -\hat{V}^{-1}(\mathbf{F}I - J)V_{\delta f}V_{if}^{-1}$$

Hence, we have constructed

$$T(\mathbf{F}) = \begin{bmatrix} V_{if}^{-1} & 0 \\ -\widehat{V}^{-1}(\mathbf{F}I - J)V_{\delta f}V_{if}^{-1} & \widehat{V}^{-1} \end{bmatrix}$$

Notice that $|T(\mathbf{F})| = (|V_{if}||\hat{V}|)^{-1}$, which is a nonzero constant. Accordingly, construction of the reduced system invoices operating on $(A^*\mathbf{F} - B^*)VE_ty_t = C^*E_tx_t$ with the "equation transformation" $\widehat{T}(\mathbf{F}) \equiv T(\mathbf{F})\eta(\mathbf{F})$ to produce $(A^{**}\mathbf{F} - B^{**})E_ty_t = T(\mathbf{F})\eta(\mathbf{F})E_tx_t = \Psi(\mathbf{F})E_tx_t$. In this expression, the transformed system matrices take the form:

$$A^{**} = \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}$$

$$B^{**} = \begin{bmatrix} I & K \\ 0 & W \end{bmatrix}$$

Properties of the Reduced System

We now demonstrate and discuss several properties of the reduced dimension nonsingular system. First, we indicate that the finite eigenvalues of the reduced system

$$E_t d_{t+1} = W \ d_t + \Psi_d(\mathbf{F}) E_t x_t$$

are identical to those of the original system. Previously, we have seen that

$$W = \hat{V}^{-1} J \hat{V}.$$

Accordingly, $\hat{V}$ is a left eigenvector matrix of W. The corresponding eigenvalues are the diagonal elements of J. Two additional points of interest are as follows: (i) if $x_t = 0$ for all t, there is one identical dynamic canonical variables representation of the reduced dynamic system as the original one; and (ii) the invariance of the eigenvalues may be viewed as an implication of the fact that $|\widehat{T}(\mathbf{F})| = |T(\mathbf{F})||\eta(\mathbf{F})|$ is a nonzero constant and, hence, this transformation does not affect the determinantal polynomial. Second, we want to establish that solvability of the original dynamic system implies solvability of the reduced dynamic system. The earlier solvability condition was that $V_{U\Lambda}$ was square and invertible. We now need to establish that this implies that $\hat{V}_{u\lambda}$ is square and nonsingular. To show this, we write

$$\hat{V} \equiv \begin{bmatrix} \widehat{V}_{u\lambda} & \widehat{V}_{uk} \\ \widehat{V}_{u\lambda} & \widehat{V}_{sk} \end{bmatrix} = [V_{\delta d} - V_{\delta f} V_{if}^{-1} V_{id}]$$

Using the second of the partitionings displayed above, we find that

$$\hat{V}_{u\lambda} = V_{u\lambda} - V_{uf} V_{if}^{-1} V_{i\lambda}$$

Since

$$|V_{U\Lambda}| = \left| \begin{matrix} V_{if} & V_{i\lambda} \\ V_{u\delta} & V_{u\lambda} \end{matrix} \right| = |V_{if}| \cdot |V_{u\lambda} - V_{uf} V_{if}^{-1} V_{i\lambda}|,$$

It follows that $|\hat{V}_{u\lambda}| = |V_{U\Lambda}|/|V_{if}| \neq 0$ and $\hat{V}_{u\lambda}$ is invertible.