Real Business Cycle Models: Linear Approximation and GMM Estimation

Craig Burnside[*]
The World Bank

May 1, 1999
Revision 7 (Preliminary)

1 Introduction

These notes are intended to provide interested students with a straightforward guide to solving and estimating real business cycle models, although the methods are applicable more generally. The solution method described here fits within the framework of Blanchard and Kahn (1980) and has been described previously by King, Plosser and Rebelo (1988a,b) (KPR). This method obtains approximate solutions by linearizing the Euler equations (appropriately detrended) of a model around its nonstochastic steady state solution. The interested reader should also consult King and Watson (1997), Uhlig (1997) and Christiano (1998), among others, for more general solution methods based on linear approximation. The estimation method implemented here is a variant of Hansen's (1982) Generalized Method of Moments (GMM).

The methods described here are useful when solving models which have no closed form solutions for the decision variables. They also map easily into the general framework of linear-quadratic (LQ) stochastic optimization and are very popular in the literature. A good understanding of the KPR method is obtained by considering LQ stochastic optimization problems. Sections 2 and 3 consider such problems in some detail. Section 2 describes the solution of a simple univariate LQ problem. Section 3 generalizes this method for multivariate LQ problems. Section 4 introduces three simple real business cycle models with deterministic growth for which the solution method is illustrated. Section 5 discusses the role played by boundedness assumptions in obtaining unique solutions, and what happens when boundedness assumptions are not sufficient for uniqueness. Section 6 describes the computation of impulse response functions, which are useful devices for studying RBC models. Section 7

[*] As with previous versions of these notes, this version is likely to contain some typographical errors (at the very least). Consequently, please use these notes at your own risk, and only redistribute them with this warning attached! Any comments or corrections are much appreciated. Thanks to Dave DeJong, Sangjoon Jun, Charles Leung and Humam Sakhnini for pointing out errors in previous versions. Nothing in these notes should be taken to be the opinion of the World Bank. Correspondence should be directed to: Craig Burnside, The World Bank, 1818 H Street NW, Washington DC 20433. E-mail: aburnside@worldbank.org.

illustrates how the solutions to LQ problems can be used to compute the autocovariances of variables of interest. It also illustrates how the autocovariances of Hodrick and Prescott (HP) (1997) filtered data can be obtained from the regular autocovariances. Section 8 discusses how models with stochastic growth can be solved, and how relevant moments can be computed. Section 9 discusses estimation by GMM amd hypothesis testing. Section 10 describes the programs which accompany these notes.

2 Univariate LQ Stochastic Optimization

This example is taken from Sargent (1987), Chapter XIV. This is an example of the decision problem of competitive firms investing in an uncertain environment. Suppose the representative firm's problem is to maximize

$$v_0 = E_0 \sum_{t=0}^{\infty} \beta^t \left[p_t f_0 k_t - J_t (k_t - k_{t-1}) - \frac{d}{2}(k_t - k_{t-1})^2 \right]$$

by choosing a stochastic process $\{k_t\}_{t=0}^{\infty}$ subject to k_{-1} given. The firm is a price taker with respect to the stochastic processes $\{J_t\}$, the price of capital, and $\{p_t\}$, the market price of output. The industry demand curve for output at time t is given by

$$p_t = A_0 - A_1 Y_t + u_t, \qquad A_0,\ A_1\ > 0,$$

where Y_t is industry output. Since firm output is given by, $f_0 k_t$, and there are n firms, $Y_t = n f_0 k_t$. At time 0 the firm has an information set, Ω_0 consisting of at least $\{p_0, p_{-1}, \ldots, J_0, J_{-1}, \ldots, u_0, u_{-1}, \ldots\}$. E_0 denotes the expectations operator conditional on Ω_0.

Differentiate v_0 with respect to k_t and you obtain

$$p_t f_0 - J_t - d(k_t - k_{t-1}) + \beta E_t \left[J_{t+1} + d(k_{t+1} - k_t) \right] = 0.$$

Then substitute the equilibrium condition that $p_t = A_0 - A_1 n f_0 k_t + u_t$ to get

$$\beta d E_t k_{t+1} - \left[A_1 n f_0^2 + d(1+\beta) \right] k_t + d k_{t-1} = -A_0 f_0 - f_0 u_t + J_t - \beta E_t J_{t+1}$$

or

$$\begin{aligned} E_t k_{t+1} - \left[A_1 n f_0^2 (d\beta)^{-1} + (1+\beta)\beta^{-1} \right] k_t + \beta^{-1} k_{t-1} \quad &= \\ -\frac{A_0 f_0}{\beta d} - \frac{f_0}{\beta d} u_t + \frac{1}{\beta d} J_t \quad &- \quad \frac{1}{d} E_t J_{t+1}. \end{aligned} \tag{1}$$

To solve this difference equation it's useful to define the backshift operator B. The backshift operator only operates on expectations with respect to some information set. Specifically $B E_t x_{t+j} = E_t x_{t+j-1}$. The backshift operator ignores the information set and applies only to the variable whose expectation is being calculated. Another important feature of B is that you can only apply polynomials in negative powers of B to both sides of an equation if you intend to maintain an equality.

Therefore, defining

$$z_t = -\frac{A_0 f_0}{\beta d} - \frac{f_0}{\beta d} u_t + \frac{1}{\beta d} J_t - \frac{1}{d} E_t J_{t+1}$$

equation (1) can be written

$$\left\{1 - \left[A_1 n f_0^2 (d\beta)^{-1} + (1+\beta)\beta^{-1}\right] B + \beta^{-1} B^2\right\} E_t k_{t+1} = z_t \tag{2}$$

or

$$\begin{aligned} (1 + \phi\beta^{-1} B + \beta^{-1} B^2) E_t k_{t+1} &= z_t \\ (1 - \lambda_1 B)(1 - \lambda_2 B) E_t k_{t+1} &= z_t, \end{aligned}$$

where $\phi = -A_1 n f_0^2 / d - (1+\beta)$, $\lambda_1 \lambda_2 = \beta^{-1}$ and $\lambda_1 + \lambda_2 = -\phi\beta^{-1}$. Notice that the solution for the λ's is obtained from

$$\begin{aligned} \lambda &= \frac{-\phi\beta^{-1} \pm \sqrt{\phi^2 \beta^{-2} - 4\beta^{-1}}}{2}, \\ &= \frac{\beta^{-1}}{2}\left[-\phi \pm \sqrt{\phi^2 - 4\beta}\right]. \end{aligned}$$

Without loss of generality I'll let λ_1 be the one with the minus and λ_2 be the one with the plus. Notice that $-\phi > 1 + \beta$. Therefore,

$$\begin{aligned} \lambda_2 &> \frac{\beta^{-1}}{2}(1 + \beta + \sqrt{1 - 2\beta + \beta^2}) \\ &= \beta^{-1}. \end{aligned}$$

Thus, $\lambda_1 < 1 < \beta^{-1} < \lambda_2$. Therefore, solving $(1 - \lambda_2 B)^{-1}$ forward

$$(1 - \lambda_1 B) E_t k_{t+1} = \frac{-\lambda_2^{-1} B^{-1}}{1 - \lambda_2^{-1} B^{-1}} E_t z_t$$

which ends up giving you

$$\begin{aligned} E_t k_{t+1} &= \lambda_1 k_t - \lambda_2^{-1} \sum_{j=0}^{\infty} \lambda_2^{-j} B^{-j} E_t z_{t+1} \\ &= \lambda_1 k_t - \lambda_2^{-1} \sum_{j=0}^{\infty} \lambda_2^{-j} E_t z_{t+1+j}. \end{aligned} \tag{3}$$

There are two ways to go from here. We can go straight to the solution which involves updating any expectations operator in (3) by one period. I.e.

$$k_{t+1} = \lambda_1 k_t - \lambda_2^{-1} \sum_{j=0}^{\infty} \lambda_2^{-j} E_{t+1} z_{t+1+j}$$

or

$$k_t = \lambda_1 k_{t-1} - \lambda_2^{-1} \sum_{j=0}^{\infty} \lambda_2^{-j} E_t z_{t+j}. \tag{4}$$

Or, proceeding carefully, we can substitute the expression for $E_t k_{t+1}$ in (3) into (2) to get

$$\lambda_1 k_t - \lambda_2^{-1} \sum_{j=0}^{\infty} \lambda_2^{-j} E_t z_{t+1+j} - (\lambda_1 + \lambda_2) k_t + \lambda_1 \lambda_2 k_{t-1} = z_t$$

which implies

$$\lambda_2 k_t = \lambda_1 \lambda_2 k_{t-1} - \sum_{j=0}^{\infty} \lambda_2^{-j} E_t z_{t+j}$$

or

$$k_t = \lambda_1 k_{t-1} - \lambda_2^{-1} \sum_{j=0}^{\infty} \lambda_2^{-j} E_t z_{t+j}.$$

Notice that this solution is the same as the solution for the certainty case, except that all variables are replaced by their expectations conditional on information at time t. This illustrates the principle of "certainty equivalence".

The solution we have obtained is not unique. Consider the candidate solution

$$k_t = \lambda_1 k_{t-1} - \lambda_2^{-1} \sum_{j=0}^{\infty} \lambda_2^{-j} E_t z_{t+j} + \xi_t. \tag{5}$$

The Euler equation is $E_t k_{t+1} - (\lambda_1 + \lambda_2) k_t + \lambda_1 \lambda_2 k_{t-1} = z_t$. Substituting our candidate solution, (5), into the Euler equation we obtain

$$\lambda_1 k_t - \lambda_2^{-1} \sum_{j=0}^{\infty} \lambda_2^{-j} E_t z_{t+1+j} + E_t \xi_{t+1} - \lambda_1 k_t - \lambda_2 (k_t - \lambda_1 k_{t-1}) = z_t.$$

Further susbtitution reveals the restriction $E_t \xi_{t+1} = \lambda_2 \xi_t$. Thus, there is an infinite number of solutions, corresponding to different processes ξ_t. However, because $\lambda_2 > 1$, it is possible to rule out all solutions except $\xi_t = 0$, for all t, if the focus is on solutions which are bounded. Recursive substitution of the general solution reveals that $\lim_{j \to \infty} E_t k_{t+j}$ does not exist for arbitrary ξ_t, unless $\xi_t = 0$, for all t.

3 A General LQ Stochastic Optimization Problem

Consider the general problem of choosing a contingency plan for $\{u_t\}_{t=0}^{\infty}$ to maximize

$$E_0 \sum_{t=0}^{\infty} \beta^t [x_t' A_{xx} x_t + 2 x_t' A_{xu} u_t + 2 x_t' A_{xz} z_t + u_t' A_{uu} u_t + 2 u_t' A_{uz} z_t + z_t' A_{zz} z_t],$$

subject to

$$B_n x_{t+1} = B_x x_t + B_u u_t + B_z z_t.$$

The variables are: x_t, the state is $n_s \times 1$, u_t the control is $n_c \times 1$, and z_t, the exogenous variable is $n_e \times 1$. The matrix B_n is $n_{cs} \times n_s$.

Set up a Lagrangean for the problem as

$$\mathcal{L} = E_0 \sum_{t=0}^{\infty} \beta^t \, [x_t' A_{xx} x_t + 2 x_t' A_{xu} u_t + 2 x_t' A_{xz} z_t + u_t' A_{uu} u_t + 2 u_t' A_{uz} z_t + z_t' A_{zz} z_t +$$

$$2\lambda_t'(B_x x_t + B_u u_t + B_z z_t - B_n x_{t+1})] \,.$$

The Lagrange multipliers, λ_t, are often called co-state variables, and have dimension $n_{cs} \times 1$. The Euler equation for u_t is

$$A_{uu} u_t + A_{xu}' x_t + A_{uz} z_t + B_u' \lambda_t = 0 \tag{6}$$

while the Euler equation for x_{t+1} is

$$-B_n' \lambda_t + \beta E_t (A_{xx} x_{t+1} + A_{xu} u_{t+1} + A_{xz} z_{t+1} + B_x' \lambda_{t+1}) = 0 \tag{7}$$

and the constraint is

$$B_x x_t + B_u u_t + B_z z_t - B_n x_{t+1} = 0. \tag{8}$$

In the notation of KPR (1988b), rewrite (6) as

$$\begin{aligned}
A_{uu} u_t &= \begin{pmatrix} -A_{xu}' & -B_u' \end{pmatrix} \begin{pmatrix} x_t \\ \lambda_t \end{pmatrix} - A_{uz} z_t \quad \text{or} \\
M_{cc} u_t &= M_{cs} \begin{pmatrix} x_t \\ \lambda_t \end{pmatrix} + M_{ce} z_t \\
u_t &= M_{cc}^{-1} M_{cs} \begin{pmatrix} x_t \\ \lambda_t \end{pmatrix} + M_{cc}^{-1} M_{ce} z_t \qquad (9)
\end{aligned}$$

Equations (7) and (8) can be rewritten as

$$\begin{pmatrix} \beta A_{xx} & \beta B_x' \\ -B_n & 0 \end{pmatrix} E_t \begin{pmatrix} x_{t+1} \\ \lambda_{t+1} \end{pmatrix} + \begin{pmatrix} 0 & -B_n' \\ B_x & 0 \end{pmatrix} \begin{pmatrix} x_t \\ \lambda_t \end{pmatrix} = \begin{pmatrix} -\beta A_{xu} \\ 0 \end{pmatrix} E_t u_{t+1} +$$

$$\begin{pmatrix} 0 \\ -B_u \end{pmatrix} u_t + \begin{pmatrix} -\beta A_{xz} \\ 0 \end{pmatrix} E_t z_{t+1} + \begin{pmatrix} 0 \\ -B_z \end{pmatrix} z_t$$

or

$$M_{ss}^0 E_t \begin{pmatrix} x_{t+1} \\ \lambda_{t+1} \end{pmatrix} + M_{ss}^1 \begin{pmatrix} x_t \\ \lambda_t \end{pmatrix} = M_{sc}^0 E_t u_{t+1} + M_{sc}^1 u_t + M_{se}^0 E_t z_{t+1} + M_{se}^1 z_t. \tag{10}$$

Combining this with the (9) we get

$$(M_{ss}^0 - M_{sc}^0 M_{cc}^{-1} M_{cs}) E_t \begin{pmatrix} x_{t+1} \\ \lambda_{t+1} \end{pmatrix} + (M_{ss}^1 - M_{sc}^1 M_{cc}^{-1} M_{cs}) \begin{pmatrix} x_t \\ \lambda_t \end{pmatrix} =$$

$$(M_{se}^0 + M_{sc}^0 M_{cc}^{-1} M_{ce}) E_t z_{t+1} + (M_{se}^1 + M_{sc}^1 M_{cc}^{-1} M_{ce}) z_t$$

or

$$\bar{M}_{ss}^0 E_t \begin{pmatrix} x_{t+1} \\ \lambda_{t+1} \end{pmatrix} + \bar{M}_{ss}^1 \begin{pmatrix} x_t \\ \lambda_t \end{pmatrix} = \bar{M}_{se}^0 E_t z_{t+1} + \bar{M}_{se}^1 z_t$$

or

$$\begin{aligned}
E_t \begin{pmatrix} x_{t+1} \\ \lambda_{t+1} \end{pmatrix} &= -(\bar{M}_{ss}^0)^{-1} \bar{M}_{ss}^1 \begin{pmatrix} x_t \\ \lambda_t \end{pmatrix} + (\bar{M}_{ss}^0)^{-1} \bar{M}_{se}^0 E_t z_{t+1} + (\bar{M}_{ss}^0)^{-1} \bar{M}_{se}^1 z_t \\
&= W \begin{pmatrix} x_t \\ \lambda_t \end{pmatrix} + R E_t z_{t+1} + Q z_t. \qquad (11)
\end{aligned}$$

Whenever you have an $n \times n$ matrix A with n linearly independent eigenvectors you can form the diagonalization, $A = P\Lambda P^{-1}$, where Λ is a diagonal matrix with the eigenvalues of A on its diagonal, and P is a matrix whose columns are n linearly independent eigenvectors of A. The eigenvalues and eigenvectors are arranged in these matrices so that eigenvector in the ith column of P is an eigenvector corresponding to the ith eigenvalue on the diagonal of Λ. Assuming that $n_s + n_{cs}$ linearly independent eigenvectors exist for W, let $P\Lambda P^{-1} = W$. Then multiplying (11) through by P^{-1} we get

$$\begin{aligned} P^{-1}E_t \begin{pmatrix} x_{t+1} \\ \lambda_{t+1} \end{pmatrix} &= \Lambda P^{-1} \begin{pmatrix} x_t \\ \lambda_t \end{pmatrix} + P^{-1}RE_t z_{t+1} + P^{-1}Qz_t \\ E_t \begin{pmatrix} \tilde{x}_{t+1} \\ \tilde{\lambda}_{t+1} \end{pmatrix} &= \Lambda \begin{pmatrix} \tilde{x}_t \\ \tilde{\lambda}_t \end{pmatrix} + P^{-1}RE_t z_{t+1} + P^{-1}Qz_t. \end{aligned}$$

A typical way of proceeding is to verify that n_{cs} eigenvalues of W are greater than 1 in absolute value, and that n_s are less than 1 in absolute value.[1] Then, Λ is constructed with the eigenvalues in increasing order of modulus and is decomposed into

$$\Lambda = \begin{pmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{pmatrix}$$

with all elements of Λ_1 less than 1, and all values of Λ_2 greater than 1. As a result the equation for $\tilde{x}_{t+1}$ should be solved backward, while the equation for $\tilde{\lambda}_{t+1}$ should be solved forward.

At this point we can partition the matrices W, R, Q, P and P^{-1} as follows:

$$W = \begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix} \qquad R = \begin{pmatrix} R_x \\ R_\lambda \end{pmatrix} \qquad Q = \begin{pmatrix} Q_x \\ Q_\lambda \end{pmatrix}$$

and

$$P = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix} \qquad P^{-1} = \begin{pmatrix} P^{11} & P^{12} \\ P^{21} & P^{22} \end{pmatrix}.$$

Notice[2] also that since $W = P\Lambda P^{-1}$ we have

$$\begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix} = \begin{pmatrix} P_{11}\Lambda_1 P^{11} + P_{12}\Lambda_2 P^{21} & P_{11}\Lambda_1 P^{12} + P_{12}\Lambda_2 P^{22} \\ P_{21}\Lambda_1 P^{11} + P_{22}\Lambda_2 P^{21} & P_{21}\Lambda_1 P^{12} + P_{22}\Lambda_2 P^{22} \end{pmatrix}.$$

Now solving the first equation backward we get

$$E_t\tilde{x}_{t+1} = \Lambda_1 \tilde{x}_t + (P^{11}R_x + P^{12}R_\lambda)E_t z_{t+1} + (P^{11}Q_x + P^{12}Q_\lambda)z_t$$

and solving the other equation forward we get

$$\begin{aligned} E_t\tilde{\lambda}_{t+1} &= \Lambda_2\tilde{\lambda}_t + (P^{21}R_x + P^{22}R_\lambda)E_t z_{t+1} + (P^{21}Q_x + P^{22}Q_\lambda)z_t \\ \tilde{\lambda}_t &= \Lambda_2^{-1}E_t\tilde{\lambda}_{t+1} - \Lambda_2^{-1}(P^{21}R_x + P^{22}R_\lambda)E_t z_{t+1} - \Lambda_2^{-1}(P^{21}Q_x + P^{22}Q_\lambda)z_t \\ &= -\sum_{j=0}^{\infty} \Lambda_2^{-(j+1)} \left[(P^{21}R_x + P^{22}R_\lambda)E_t z_{t+1+j} + (P^{21}Q_x + P^{22}Q_\lambda)E_t z_{t+j} \right]. \end{aligned}$$

[1] For some models this can be verified analytically for subsets of the parameter space. See KPR (1988b). Later, we will consider the case where the number of eigenvalues inside the unit circle is not equal to n_s.

[2] The reader should be careful to note that unless $n_s = n_{cs}$ the various submatrices are not all $n_s \times n_s$. Their dimensions are implicitly defined by conformability in the equations that follow.

Going back to the original difference equation you have

$$x_{t+1} = W_{11}x_t + W_{12}\lambda_t + R_x E_t z_{t+1} + Q_x z_t. \tag{12}$$

Notice that the mapping between the tilded ($\tilde{\bullet}$) variables and the regular variables is

$$\begin{pmatrix} \tilde{x}_t \\ \tilde{\lambda}_t \end{pmatrix} = \begin{pmatrix} P^{11} & P^{12} \\ P^{21} & P^{22} \end{pmatrix} \begin{pmatrix} x_t \\ \lambda_t \end{pmatrix}.$$

As a result, $P^{21}x_t + P^{22}\lambda_t = \tilde{\lambda}_t$ or $\lambda_t = -(P^{22})^{-1}P^{21}x_t + (P^{22})^{-1}\tilde{\lambda}_t$. Substituting this into (12) and using the solution for partitioned W given above

$$\begin{aligned} x_{t+1} &= (P_{11}\Lambda_1 P^{11} + P_{12}\Lambda_2 P^{21})x_t - (P_{11}\Lambda_1 P^{12} + P_{12}\Lambda_2 P^{22})(P^{22})^{-1}P^{21}x_t \\ &\quad +(P_{11}\Lambda_1 P^{12} + P_{12}\Lambda_2 P^{22})(P^{22})^{-1}\tilde{\lambda}_t + R_x E_t z_{t+1} + Q_x z_t \\ &= \left(P_{11}\Lambda_1[P^{11} - P^{12}(P^{22})^{-1}P^{21}]\right)x_t + (P_{11}\Lambda_1 P^{12} + P_{12}\Lambda_2 P^{22})(P^{22})^{-1}\tilde{\lambda}_t + \\ &\quad R_x E_t z_{t+1} + Q_x z_t. \end{aligned}$$

Now, recall the partitioned inverse formula. The inverse of a matrix

$$\begin{pmatrix} E & F \\ G & H \end{pmatrix}$$

is given by

$$\begin{pmatrix} D^{-1} & -D^{-1}FH^{-1} \\ -H^{-1}GD^{-1} & H^{-1} + H^{-1}GD^{-1}FH^{-1} \end{pmatrix}$$

where $D = E - FH^{-1}G$. Therefore the term in square brackets above is equal to P_{11}^{-1} so that

$$x_{t+1} = (P_{11}\Lambda_1 P_{11}^{-1})x_t + (P_{11}\Lambda_1 P^{12} + P_{12}\Lambda_2 P^{22})(P^{22})^{-1}\tilde{\lambda}_t + R_x E_t z_{t+1} + Q_x z_t. \tag{13}$$

This expresses the solution for x_{t+1} as a function of the past state, and, given the solution for $\tilde{\lambda}_t$, as a function of current and expected future values of the exognous variables, z_t. To get the solution for λ_t simply do as above to get

$$\lambda_t = -(P^{22})^{-1}P^{21}x_t + (P^{22})^{-1}\tilde{\lambda}_t \tag{14}$$

while the decision rule for the control is given by (9) above.

Suppose we choose a simple AR(1) representation for z_t, so that $z_{t+1} = \Pi z_t + \epsilon_{t+1}$. Then $E_t z_{t+j} = \Pi^j z_t$. To make sure this formula converges as $j \to \infty$ we need to assume that the eigenvalues of Π are less than 1 in modulus. This is the equivalent of the roots of $|(I - \Pi Z)| = 0$ having roots greater than 1 in modulus. The easiest way to solve the model is to go back to the solution for $\tilde{\lambda}_t$ given above

$$\begin{aligned} \tilde{\lambda}_t &= -\sum_{j=0}^{\infty} \Lambda_2^{-(j+1)} \left[(P^{21}R_x + P^{22}R_\lambda)E_t z_{t+1+j} + (P^{21}Q_x + P^{22}Q_\lambda)E_t z_{t+j}\right] \\ &= -\sum_{j=0}^{\infty} \Lambda_2^{-(j+1)}(\Phi_0 E_t z_{t+1+j} + \Phi_1 E_t z_{t+j}) \\ &= -\left[\sum_{j=0}^{\infty} \Lambda_2^{-(j+1)}(\Phi_0 \Pi + \Phi_1)\Pi^j\right] z_t \\ &= \Psi z_t. \end{aligned}$$

We can develop an explicit formula for the rows of Ψ by exploiting the diagonality of Λ_2. Defining Λ_{2i} as the ith diagonal element of Λ_2, and Φ_{ji} as the ith row of Φ_j, $j = 0,\ 1$, it follows that the ith row of Ψ, denoted Ψ_i, is given by

$$\begin{aligned}
\Psi_i &= -\left[\sum_{j=0}^{\infty} \Lambda_{2i}^{-(j+1)}(\Phi_{0i}\Pi + \Phi_{1i})\Pi^j\right] \\
&= -\Lambda_{2i}^{-1}(\Phi_{0i}\Pi + \Phi_{1i})\sum_{j=0}^{\infty} \Lambda_{2i}^{-j}\Pi^j \\
&= -\Lambda_{2i}^{-1}(\Phi_{0i}\Pi + \Phi_{1i})(I_{n_e} - \Lambda_{2i}^{-1}\Pi)^{-1}
\end{aligned}$$

Then, the solution for x_{t+1} is just

$$\begin{aligned}
x_{t+1} &= (P_{11}\Lambda_1 P_{11}^{-1})x_t + (P_{11}\Lambda_1 P^{12} + P_{12}\Lambda_2 P^{22})(P^{22})^{-1}\Psi z_t + R_x \Pi z_t + Q_x z_t \\
&= (P_{11}\Lambda_1 P_{11}^{-1})x_t + \left[(P_{11}\Lambda_1 P^{12} + P_{12}\Lambda_2 P^{22})(P^{22})^{-1}\Psi + R_x\Pi + Q_x\right] z_t \\
&= \Upsilon_{xx} x_t + \Upsilon_{xz} z_t \qquad (15)
\end{aligned}$$

The solution for λ_t is,

$$\begin{aligned}
\lambda_t &= -(P^{22})^{-1}P^{21}x_t + (P^{22})^{-1}\Psi z_t \\
&= \Upsilon_{\lambda x} x_t + \Upsilon_{\lambda z} z_t \qquad (16)
\end{aligned}$$

and the solution for the controls is obtained from (9)

$$\begin{aligned}
u_t &= M_{cc}^{-1} M_{cs} \begin{pmatrix} I \\ -(P^{22})^{-1}P^{21} \end{pmatrix} x_t + \left[M_{cc}^{-1} M_{cs} \begin{pmatrix} 0 \\ (P^{22})^{-1}\Psi \end{pmatrix} + M_{cc}^{-1} M_{ce} \right] z_t \\
&= \Upsilon_{ux} x_t + \Upsilon_{uz} z_t. \qquad (17)
\end{aligned}$$

4 Three Real Business Cycle Models

4.1 A Simple Growth Model

So far we have only looked at linear-quadratic models. However, for reasons of parsimony, and modelling flexibility it is often useful to look at non-LQ models. In general these models will have Euler equations which are not linear in the unknown solution functions. Take for example the Cass-Koopmans neoclassical growth model with a 100% depreciation rate for capital:

$$\max \quad E_0 \sum_{t=0}^{\infty} \beta^t \ln(c_t)$$

s.t. $c_t + k_{t+1} = A_t k_t^\alpha$. Setting up this problem as we set up the general LQ problem the Lagrangean is

$$\mathcal{L} = E_0 \sum_{t=0}^{\infty} \beta^t \left[\ln(c_t) + \lambda_t (A_t k_t^\alpha - k_{t+1} - c_t)\right].$$

The Euler equations for the choices of c_t and k_{t+1}, along with the constraint are

$$\begin{aligned} c_t^{-1} &= \lambda_t \\ \lambda_t &= \beta E_t(\lambda_{t+1}\alpha A_{t+1}k_{t+1}^{\alpha-1}) \\ c_t + k_{t+1} &= A_t k_t^{\alpha}. \end{aligned}$$

Clearly the trivial solution for c_t is $c_t = \lambda_t^{-1}$. You then have

$$\begin{aligned} \lambda_t &= \beta E_t(\lambda_{t+1}\alpha A_{t+1}k_{t+1}^{\alpha-1}) \\ \lambda_t^{-1} &= A_t k_t^{\alpha} - k_{t+1}. \end{aligned}$$

Then substitute the solution for λ_t into the first equation to get the fundamental difference equation in k (it is nonlinear).

$$\frac{1}{A_t k_t^{\alpha} - k_{t+1}} = \beta E_t \frac{\alpha A_{t+1}k_{t+1}^{\alpha-1}}{A_{t+1}k_{t+1}^{\alpha} - k_{t+2}}.$$

The solution for k_{t+1} is proportional to output, i.e. $k_{t+1} = dA_t k_t^{\alpha}$. To get the form of d substitute this guess into the difference equation for k

$$\begin{aligned} \frac{1}{(1-d)A_t k_t^{\alpha}} &= \beta E_t \frac{\alpha A_{t+1}k_{t+1}^{\alpha-1}}{(1-d)A_{t+1}k_{t+1}^{\alpha}} \\ &= \beta \frac{\alpha}{k_{t+1}(1-d)} \\ &= \beta \frac{\alpha}{dA_t k_t^{\alpha}(1-d)} \end{aligned}$$

Clearly this implies $d = \alpha\beta$ so that $k_{t+1} = \alpha\beta A_t k_t^{\alpha}$. This happens to be a fortuitous result. It is very special to the functional form. Notice that in logarithms the solution is linear $\ln(k_{t+1}) = \ln(\alpha\beta) + \ln(A_t) + \alpha\ln(k_t)$.

If we generalize the Cass-Koopmans model to the case where there is not complete depreciation we get

$$\max \quad E_0 \sum_{t=0}^{\infty} \beta^t \ln(c_t)$$

s.t. $c_t + k_{t+1} - (1-\delta)k_t = A_t k_t^{\alpha}$. It is straightforward to show that the fundamental difference equation in k is

$$\frac{1}{A_t k_t^{\alpha} + (1-\delta)k_t - k_{t+1}} = \beta E_t \left[\frac{\alpha A_{t+1}k_{t+1}^{\alpha-1} + (1-\delta)}{A_{t+1}k_{t+1}^{\alpha} + (1-\delta)k_{t+1} - k_{t+2}} \right].$$

There is no closed form solution for k_{t+1} in this case. The basic KPR method and similar methods involve obtaining an approximation to the solution which is linear in the logarithms of the variables.

4.2 The Divisible and Indivisible Labor Models

KPR present a quite general dynamic optimization problem in an environment which may involve growth. We will consider less general economies in which the solution to a social planner's problem is equivalent to the equilibrium of a decentralized competitive economy.[3] In particular, we consider a social planner's problem

$$\max \quad U = E_0 \sum_{t=0}^{\infty} \beta^t \left[\ln(C_t) + \theta V(1 - N_t) \right]$$

s.t.

$$C_t + K_{t+1} - (1-\delta)K_t = A_t K_t^{1-\alpha} (N_t X_t)^{\alpha}.$$

The function V defines utility over the fraction of the representative agent's time spent in leisure. In the case of the *divisible labor* model of KPR (1988a) the function V is given by

$$V(1 - N_t) = \ln(1 - N_t),$$

whereas, in the *indivisible labor* model of Hansen (1985) and Rogerson (1988) the function V is given by[4]

$$V(1 - N_t) = 1 - N_t.$$

Growth in these economies comes from the assumption of deterministic labor augmenting technical progress in the form of $X_t = \gamma_X X_{t-1}$, with $\gamma_X > 1$.

If you set up the Lagrangean for this problem you get

$$\mathcal{L} = E_0 \sum_{t=0}^{\infty} \left\{ \beta^t \left[\ln(C_t) + \theta V(1 - N_t) \right] + \Lambda_t \left[A_t K_t^{1-\alpha} (N_t X_t)^{\alpha} + (1-\delta)K_t - K_{t+1} - C_t \right] \right\}.$$

The efficiency conditions for this problem are

$$\begin{aligned}
\beta^t C_t^{-1} - \Lambda_t &= 0 \\
-\beta^t \theta V'(1 - N_t) + \Lambda_t \alpha A_t K_t^{1-\alpha} (N_t X_t)^{\alpha - 1} X_t &= 0 \\
-\Lambda_t + E_t \Lambda_{t+1} [(1-\alpha) A_{t+1} K_{t+1}^{-\alpha} (N_{t+1} X_{t+1})^{\alpha} + (1-\delta)] &= 0 \\
A_t K_t^{1-\alpha} (N_t X_t)^{\alpha} + (1-\delta)K_t - K_{t+1} - C_t &= 0
\end{aligned}$$

The transversality condition is $\lim_{t\to\infty} \Lambda_t K_t = 0$.

[3] The interested reader should refer to King, Plosser and Rebelo (1988b) for their discussion of a general set up. There are many models where the social planner's solution is not a competitive equilibrium. In these cases, the KPR method of linearizing can still be applied in the solution of the Euler equations for the competitive equilibrium.

[4] In the indivisible labor model the variable N_t actually represents the probability of employment. The underlying utility function is logarithmic in leisure, and if the fraction of their time endowment that the employed spend working is denoted h, the expected utility is $N_t \tilde{\theta} \ln(1-h) + (1-N_t)\tilde{\theta} \ln(1) = \theta \ln(1-h) N_t$. So, letting $V(1 - N_t) = 1 - N_t$ simply involves adding a constant to the utility function and defining the constant $\theta = -\tilde{\theta} \ln(1-h)$.

Now define transformed variables $c_t = C_t/X_t$, $k_t = K_t/X_t$, $y_t = Y_t/X_t$, and $\lambda_t = \beta^{-t}\Lambda_t X_t$. Clearly in terms of these variables the Euler equations are

$$\begin{aligned} c_t^{-1} - \lambda_t &= 0 \\ -\theta V'(1-N_t) + \lambda_t \alpha A_t k_t^{1-\alpha} N_t^{\alpha-1} &= 0 \\ -\lambda_t + \beta E_t \lambda_{t+1} \gamma_X^{-1}[(1-\alpha)A_{t+1}k_{t+1}^{-\alpha}N_{t+1}^{\alpha} + (1-\delta)] &= 0 \\ A_t k_t^{1-\alpha} N_t^{\alpha} + (1-\delta)k_t - k_{t+1}\gamma_X - c_t &= 0 \end{aligned}$$

with the transversality condition becomes $\lim_{t\to\infty} \beta^t \lambda_t k_t = 0$.

Notice that a nonstochastic steady state in terms of the transformed variables will exist, since they have been divided by their common growth component, X_t.[5] To find that steady state simply remove time subscripts from the Euler equations

$$\begin{aligned} c^{-1} - \lambda &= 0 && (18) \\ -\theta V'(1-N) + \lambda \alpha A k^{1-\alpha} N^{\alpha-1} &= 0 && (19) \\ -\lambda + \beta \lambda \gamma_X^{-1}[(1-\alpha)Ak^{-\alpha}N^{\alpha} + (1-\delta)] &= 0 && (20) \\ Ak^{1-\alpha}N^{\alpha} + (1-\delta)k - k\gamma_X - c &= 0 && (21) \end{aligned}$$

The steady state output-capital ratio is given by (20)

$$1 = \beta\gamma_X^{-1}[(1-\alpha)(y/k) + (1-\delta)]. \qquad (22)$$

Therefore,

$$(y/k) = [\beta^{-1}\gamma_X - (1-\delta)]/(1-\alpha).$$

Then, from (21)

$$(c/y) = 1 + [(1-\delta) - \gamma_X](k/y).$$

Then from (19) you get $\theta V'(1-N) = \alpha(y/c)/N$. Thus, in the *divisible labor* model

$$N = \alpha(y/c) \Big/ [\theta + \alpha(y/c)].$$

In the *indivisible labor* model

$$N = (\alpha/\theta)(y/c).$$

Notice that the production function implies that $y = Ak^{1-\alpha}N^{\alpha}$. Therefore, the capital-labor ratio is given by

$$(k/N) = (y/k)^{-1/\alpha}A^{1/\alpha}.$$

Given the solution for N we then have a full set of steady state values for k, then y, then c, and finally λ.

As in the Cass-Koopmans case there is no closed-form solution to the Euler equations for either model . Therefore, we need an alternate solution method. The method proposed by KPR approximates the Euler equations by a set of linear equations in the unknowns. The way they do this is to totally differentiate the Euler equations at the steady state values.

[5] KPR (1988b) give a detailed discussion of the stationary inducing transformation and of the existence of the nonstochastic steady state.

This provides a linear approximation to the Euler equations in the neighbourhood of the steady state. Totally differentiating the Euler equations above

$$-c^{-2}dc_t - d\lambda_t = 0 \tag{23}$$

$$\begin{aligned}\theta V''(1-N)dN_t + \alpha A k^{1-\alpha}N^{\alpha-1}d\lambda_t + \lambda\alpha k^{1-\alpha}N^{\alpha-1}dA_t + {}&\\ (1-\alpha)\lambda\alpha A k^{-\alpha}N^{\alpha-1}dk_t + (\alpha-1)\lambda\alpha A k^{1-\alpha}N^{\alpha-2}dN_t &= 0\end{aligned} \tag{24}$$

$$\begin{aligned}-d\lambda_t + E_t d\lambda_{t+1} + (\beta/\gamma_X)\Big(\lambda(1-\alpha)k^{-\alpha}N^{\alpha}E_t dA_{t+1} + {}&\\ \lambda[-\alpha(1-\alpha)Ak^{-\alpha-1}N^{\alpha}]E_t dk_{t+1} + \lambda\alpha(1-\alpha)Ak^{-\alpha}N^{\alpha-1}E_t dN_{t+1}\Big) &= 0\end{aligned} \tag{25}$$

$$\begin{aligned}k^{1-\alpha}N^{\alpha}dA_t + (1-\alpha)Ak^{-\alpha}N^{\alpha}dk_t + \alpha A k^{1-\alpha}N^{\alpha-1}dN_t + {}&\\ (1-\delta)dk_t - \gamma_X dk_{t+1} - dc_t &= 0\end{aligned} \tag{26}$$

In (25) I have used the fact that $(\beta/\gamma_X)[(1-\alpha)Ak^{-\alpha}N^{\alpha}+(1-\delta)] = 1$. Multiply (23) through by $c = \lambda^{-1}$. Divide (25) by λ. Divide (26) by $y = Ak^{1-\alpha}N^{\alpha}$.

$$-\frac{dc_t}{c} - \frac{d\lambda_t}{\lambda} = 0 \tag{27}$$

$$\begin{aligned}\theta N V''(1-N)\frac{dN_t}{N} + \lambda\alpha\frac{y}{N}\frac{d\lambda_t}{\lambda} + \lambda\alpha\frac{y}{N}\frac{dA_t}{A} + {}&\\ (1-\alpha)\lambda\alpha\frac{y}{N}\frac{dk_t}{k} + (\alpha-1)\lambda\alpha\frac{y}{N}\frac{dN_t}{N} &= 0\end{aligned} \tag{28}$$

$$\begin{aligned}-\frac{d\lambda_t}{\lambda} + E_t\frac{d\lambda_{t+1}}{\lambda} + \frac{\beta}{\gamma_X}\Bigg((1-\alpha)\frac{y}{k}E_t\frac{dA_{t+1}}{A} - {}&\\ \alpha(1-\alpha)\frac{y}{k}\frac{dk_{t+1}}{k} + \alpha(1-\alpha)\frac{y}{k}E_t\frac{dN_{t+1}}{N}\Bigg) &= 0\end{aligned} \tag{29}$$

$$\frac{dA_t}{A} + (1-\alpha)\frac{dk_t}{k} + \alpha\frac{dN_t}{N} + (1-\delta)\frac{k}{y}\frac{dk_t}{k} - \gamma_X\frac{k}{y}\frac{dk_{t+1}}{k} - \frac{c}{y}\frac{dc_t}{c} = 0 \tag{30}$$

Noting that $\lambda\alpha(y/N) = \theta V'(1-N)$ divide through (28) by $\theta V'(1-N)$ to get

$$\frac{NV''(1-N)}{V'(1-N)}\frac{dN_t}{N} + \frac{d\lambda_t}{\lambda} + \frac{dA_t}{A} + (1-\alpha)\frac{dk_t}{k} + (\alpha-1)\frac{dN_t}{N} = 0. \tag{31}$$

Defining $\mu = (\beta/\gamma_X)(1-\alpha)(y/k) = 1 - (\beta/\gamma_X)(1-\delta)$, rewrite (29) as

$$-\frac{d\lambda_t}{\lambda} + E_t\frac{d\lambda_{t+1}}{\lambda} + \mu E_t\frac{dA_{t+1}}{A} - \alpha\mu\frac{dk_{t+1}}{k} + \alpha\mu E_t\frac{dN_{t+1}}{N} = 0. \tag{32}$$

KPR define the variables $\hat{z}_t = dz_t/z$. These variables represent first-order approximations to percentage deviations from the steady state values. I.e. $dz_t/z \approx \ln(z_t/z)$. Rewriting equations (27), (31), (32) and (30) in terms of these variables we get

$$-\hat{c}_t - \hat{\lambda}_t = 0 \tag{33}$$

$$\frac{NV''(1-N)}{V'(1-N)}\hat{N}_t + \hat{\lambda}_t + \hat{A}_t + (1-\alpha)\hat{k}_t + (\alpha-1)\hat{N}_t = 0 \tag{34}$$

$$-\hat{\lambda}_t + E_t\hat{\lambda}_{t+1} + \mu E_t\hat{A}_{t+1} - \alpha\mu\hat{k}_{t+1} + \alpha\mu E_t\hat{N}_{t+1} = 0 \tag{35}$$

$$\hat{A}_t + (1-\alpha)\hat{k}_t + \alpha\hat{N}_t + (1-\delta)\frac{k}{y}\hat{k}_t - \gamma_X\frac{k}{y}\hat{k}_{t+1} - \frac{c}{y}\hat{c}_t = 0 \tag{36}$$

Notice that this system can be written in the form

$$\begin{pmatrix} -1 & 0 \\ 0 & 1-\alpha - NV''(1-N)/V'(1-N) \end{pmatrix} \begin{pmatrix} \hat{c}_t \\ \hat{N}_t \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1-\alpha & 1 \end{pmatrix} \begin{pmatrix} \hat{k}_t \\ \hat{\lambda}_t \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \hat{A}_t$$

$$\begin{pmatrix} -\alpha\mu & 1 \\ -\gamma_X(k/y) & 0 \end{pmatrix} E_t \begin{pmatrix} \hat{k}_{t+1} \\ \hat{\lambda}_{t+1} \end{pmatrix} + \begin{pmatrix} 0 & -1 \\ (1-\alpha)+(1-\delta)(k/y) & 0 \end{pmatrix} \begin{pmatrix} \hat{k}_t \\ \hat{\lambda}_t \end{pmatrix} =$$

$$\begin{pmatrix} 0 & -\alpha\mu \\ 0 & 0 \end{pmatrix} E_t \begin{pmatrix} \hat{c}_{t+1} \\ \hat{N}_{t+1} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ (c/y) & -\alpha \end{pmatrix} \begin{pmatrix} \hat{c}_t \\ \hat{N}_t \end{pmatrix} +$$

$$\begin{pmatrix} -\mu \\ 0 \end{pmatrix} E_t \hat{A}_{t+1} + \begin{pmatrix} 0 \\ -1 \end{pmatrix} \hat{A}_t.$$

Notice that these systems of equations conform to the form of the equations solved in Section 3, where

$$u_t = \begin{pmatrix} \hat{c}_t \\ \hat{N}_t \end{pmatrix} \qquad x_t = \hat{k}_t \qquad \lambda_t = \hat{\lambda}_t \qquad z_t = \hat{A}_t$$

$$M_{cc} = \begin{pmatrix} -1 & 0 \\ 0 & 1-\alpha - NV''(1-N)/V'(1-N) \end{pmatrix} \; M_{cs} = \begin{pmatrix} 0 & 1 \\ 1-\alpha & 1 \end{pmatrix} \; M_{ce} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$M_{ss}^0 = \begin{pmatrix} -\alpha\mu & 1 \\ -\gamma_X(k/y) & 0 \end{pmatrix} \quad M_{ss}^1 = \begin{pmatrix} 0 & -1 \\ (1-\alpha)+(1-\delta)(k/y) & 0 \end{pmatrix}$$

$$M_{sc}^0 = \begin{pmatrix} 0 & -\alpha\mu \\ 0 & 0 \end{pmatrix} \quad M_{sc}^1 = \begin{pmatrix} 0 & 0 \\ (c/y) & -\alpha \end{pmatrix} \quad M_{se}^0 = \begin{pmatrix} -\mu \\ 0 \end{pmatrix} \quad M_{se}^1 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}.$$

As a result, the *divisible* and *indivisible labor* models can be solved approximately by employing the methods provided in Section 3, as long as an appropriate law of motion is chosen for the exogenous variable, A_t. The typical assumption is that

$$\ln(A_t) = (1-\rho)\ln(A) + \rho\ln(A_{t-1}) + \epsilon_t,$$

where ϵ_t is i.i.d. with mean zero and variance σ^2. This suggests an approximation to the law of motion of the form

$$\hat{A}_t = \rho\hat{A}_{t-1} + \epsilon_t,$$

where $\hat{A}_t = \ln(A_t/A)$.

Often, we are interested in variables other than the states, controls, and exogenous variables. For example, we may be interested in the properties of output, y_t, average productivity, w_t, investment, i_t, the ex-post gross real rate of return on capital, R_t^k, or the gross real interest rate, R_t. These variables are all determined in terms of the others,

$$\begin{aligned} y_t &= A_t k_t^{1-\alpha} N_t^{\alpha} \\ w_t &= A_t k_t^{1-\alpha} N_t^{\alpha-1} \\ i_t &= A_t k_t^{1-\alpha} N_t^{\alpha} - c_t \\ R_t^k &= (1-\alpha) A_t K_t^{-\alpha} (N_t X_t)^{\alpha} = (1-\alpha) A_t k_t^{-\alpha} N_t^{\alpha} + (1-\delta) \\ R_t &= \left(\beta E_t \frac{C_t}{C_{t+1}}\right)^{-1} = \frac{\gamma_X}{\beta} \left(E_t \frac{c_t}{c_{t+1}}\right)^{-1} \end{aligned}$$

Notice that these expressions can be linearized.

$$\begin{aligned}
dy_t &= (y/A)dA_t + (1-\alpha)(y/k)dk_t + \alpha(y/N)dN_t && (37)\\
dw_t &= (y/NA)dA_t + (1-\alpha)(y/Nk)dk_t + (\alpha-1)(y/N^2)dN_t && (38)\\
di_t &= (y/A)dA_t + (1-\alpha)(y/k)dk_t + \alpha(y/N)dN_t - dc_t && (39)\\
dR_t^k &= (1-\alpha)(y/kA)dA_t - \alpha(1-\alpha)(y/k^2)dk_t + \alpha(1-\alpha)(y/kN)dN_t && (40)\\
dR_t &= \frac{\gamma_X}{\beta}\left[(1/c)E_t dc_{t+1} - (1/c)dc_t\right] && (41)
\end{aligned}$$

Divide equation (37) by y. Divide equation (38) by $w = (y/N)$. Divide equation (39) by $i = [\gamma_X - (1-\delta)]k$. Divide equation (40) by $R^k = (1-\alpha)(y/k) + (1-\delta)$. Divide equation (41) by $R = (\gamma_X/\beta)$. Notice from (22) that $R^k = R$.

$$\begin{aligned}
\hat{y}_t &= \hat{A}_t + (1-\alpha)\hat{k}_t + \alpha\hat{N}_t && (42)\\
\hat{w}_t &= \hat{A}_t + (1-\alpha)\hat{k}_t + (\alpha-1)\hat{N}_t && (43)\\
\hat{i}_t &= (y/i)\hat{A}_t + (1-\alpha)(y/i)\hat{k}_t + \alpha(y/i)\hat{N}_t - (y/i)(c/y)\hat{c}_t && (44)\\
\hat{R}_t^k &= \mu\hat{A}_t - \alpha\mu\hat{k}_t + \alpha\mu\hat{N}_t, && (45)\\
\hat{R}_t &= E_t\hat{c}_{t+1} - \hat{c}_t && (46)
\end{aligned}$$

where μ is defined as above. When R^k and R are approximately equal to 1, $\hat{R}_t^k$ and $\hat{R}_t$, which represent percentage deviations from R^k and R, will be approximately measured in percentage points which is useful in their interpretation. The interest rate is more complicated to compute than the other variables since it depends on a conditional expectation. Since $\hat{c}_t$ is a control variable we can use the notation of Section 3 so that

$$\hat{c}_{t+1} = \Upsilon^c_{ux}x_{t+1} + \Upsilon^c_{uz}z_{t+1}.$$

Substituting in equation (15) we have

$$\hat{c}_{t+1} = \Upsilon^c_{ux}(\Upsilon_{xx}x_t + \Upsilon_{xz}z_t) + \Upsilon^c_{uz}z_{t+1}.$$

Taking the conditional expectation, equation (46) becomes

$$\hat{R}_t = \Upsilon^c_{ux}\Upsilon_{xx}x_t + (\Upsilon^c_{ux}\Upsilon_{xz} + \Upsilon^c_{uz}\Pi)z_t - \hat{c}_t.$$

4.3 A Labor Hoarding Model

In this section we consider the labor hoarding model in Burnside, Eichenbaum and Rebelo (1993). This model is a good example of a case in which $n_s \neq n_{cs}$. The model is a modified version of the indivisible labor model presented in the last section, and the model in Christiano and Eichenbaum (1992) which we will examine in Section 8.

The model uses the same device as in the Hansen (1985) and Rogerson (1988) models to generate variability in employment. Agents face a lottery which determines whether they will work or not work. The probability of working is N_t. If an agent works, they work a fixed number of hours, f, of their time endowment, normalized to T hours. Furthermore, they lose a fixed cost of ξ hours by reporting for work. However, workers can adjust their level of effort, e_t, which means they will receive $T - \xi - fe_t$ effective hours of leisure. Assuming logarithmic preferences over leisure, the utility from leisure for the representative agent is therefore $\theta N_t \ln(T - \xi - fe_t) + \theta(1 - N_t)\ln(T)$.

A second feature of the model is that the lottery probability, and therefore, the level of employment, is predetermined, capturing the notion that employment is costly to adjust immediately in response to new information. This amounts to making N_t a state variable which, like k_t, is chosen at time $t-1$. However, there is no additional constraint in the model, so we have two state variables but one co-state variable. The social planner's problem for this economy is equivalent to various decentralizations so we can solve the planner's problem to find a competitive equilibrium.

A final feature of the model is that government expenditure acts as a form of lump-sum taxation. This expenditure is modelled as a purely exogenous process relative to the level of technology, and introduces a second source of uncertainty into the model.

The Lagrangean for the planner's problem is

$$\begin{aligned}\mathcal{L} = E_0 \sum_{t=0}^{\infty} \beta^t \left\{\ln(C_t) + \theta\left[N_t \ln(T - \xi - fe_t) + (1 - N_t)\ln(T)\right]\right\} + \\ E_0 \sum_{t=0}^{\infty} \Lambda_t \left[A_t K_t^{1-\alpha}(fe_t N_t X_t)^{\alpha} + (1-\delta)K_t - K_{t+1} - G_t - C_t\right]\Bigg\}.\end{aligned}$$

The transformed optimality conditions for this problem are

$$\begin{aligned}c_t^{-1} - \lambda_t &= 0 \\ -\theta f N_t (T - \xi - fe_t)^{-1} + \alpha\lambda_t A_t k_t^{1-\alpha}(fN_t)^{\alpha} e_t^{\alpha-1} &= 0 \\ \theta E_t \ln\left[(T - \xi - fe_{t+1})/T\right] + E_t \lambda_{t+1} \alpha A_{t+1} k_{t+1}^{1-\alpha}(fe_{t+1})^{\alpha} N_{t+1}^{\alpha-1} &= 0 \\ -\lambda_t + \beta E_t \lambda_{t+1} \gamma_X^{-1}\left[(1-\alpha)A_{t+1}k_{t+1}^{-\alpha}(fe_{t+1}N_{t+1})^{\alpha} + (1-\delta)\right] &= 0 \\ A_t k_t^{1-\alpha}(fe_t N_t)^{\alpha} + (1-\delta)k_t - k_{t+1}\gamma_X - g_t - c_t &= 0\end{aligned}$$

As in the previous models, it is easy to verify that in nonstochastic steady state

$$(y/k) = [\beta^{-1}\gamma_X - (1-\delta)]/(1-\alpha).$$

The budget constraint implies that

$$(c/y) = 1 + [(1-\delta) - \gamma_X](k/y) - (g/y),$$

where (g/y) is parameterized. Burnside, Eichenbaum and Rebelo chose the normalization $e = 1$, so that

$$N = (\alpha/\theta f)(T - \xi - f)(y/c).$$

Finally, by choosing $e = 1$, there is an implied relationship among the parameters from the optimality condition for N_{t+1}:

$$\ln\left[(T - \xi - f)/T\right] + f/(T - \xi - f) = 0.$$

Burnside, Eichenbaum and Rebelo (1993) use this to determine f in terms of T and ξ. Proceeding as in the previous sections it is straightforward to linearize the Euler equations to obtain the following set of equations.

$$\begin{aligned}
-\hat{c}_t - \hat{\lambda}_t &= 0 & (47)\\
\hat{\lambda}_t + \hat{A}_t + (1-\alpha)\hat{k}_t + (\alpha-1)\hat{N}_t + [\alpha - 1 - f/(T-\xi-f)]\,\hat{e}_t &= 0 & (48)\\
E_t\left[\hat{\lambda}_{t+1} + \hat{A}_{t+1} + (1-\alpha)\hat{k}_{t+1} + (\alpha-1)\hat{e}_{t+1} + (\alpha-1)\hat{N}_{t+1}\right] &= 0 & (49)\\
-\hat{\lambda}_t + E_t\hat{\lambda}_{t+1} + \mu E_t\hat{A}_{t+1} - \alpha\mu\hat{k}_{t+1} + \alpha\mu E_t\hat{N}_{t+1} + \alpha\mu E_t\hat{N}_{t+1} &= 0 & (50)\\
\hat{A}_t + (1-\alpha)\hat{k}_t + \alpha\hat{N}_t + \alpha\hat{e}_t + (1-\delta)\frac{k}{y}\hat{k}_t - \gamma_X\frac{k}{y}\hat{k}_{t+1} - \frac{c}{y}\hat{c}_t - \frac{g}{y}\hat{g}_t &= 0. & (51)
\end{aligned}$$

This system of equations can easily be written in the usual form using

$$u_t = \begin{pmatrix} \hat{c}_t \\ \hat{e}_t \end{pmatrix} \qquad x_t = \begin{pmatrix} \hat{k}_t \\ \hat{N}_t \end{pmatrix} \qquad \lambda_t = \hat{\lambda}_t \qquad z_t = \begin{pmatrix} \hat{A}_t \\ \hat{g}_t \end{pmatrix}.$$

Burnside, Eichenbaum and Rebelo close the model by assuming that $\hat{A}_t = \rho_A\hat{A}_{t-1} + \epsilon_{At}$, $\hat{g}_t = \rho_g\hat{g}_{t-1} + \epsilon_{gt}$, and ϵ_{At} and ϵ_{gt} are mutually independent.

5 Uniqueness, Boundedness and Sunspots

In the previous sections we have focused attention on cases where the number of eigenvalues of the matrix W which are less than one is exactly n_s, the number of endogenous state variables. Furthermore, even in this case we ignored the possibility that there might be multiple solutions to the difference equation

$$E_t\tilde{\lambda}_{t+1} = \Lambda_2\tilde{\lambda}_t + [(P^{21}R_x + P^{22}R_\lambda)\Pi + P^{21}Q_x + P^{22}Q_\lambda]z_t.$$

In this section we consider the possibility of multiple solutions to this difference equation, as well as the possibilities that arise when the number of eigenvalues inside the unit circle, denoted n_i is not equal to n_s.

5.1 Uniqueness When $\mathbf{n_i = n_s}$

In this section we need only verify the conditions under which the solution we obtained in Section 3 is unique. It turns out that uniqueness relies on side conditions put on the process x_t. Without these side conditions there are many explosive solutions for x_t that satisfy the difference equations governing the model. Returning to the original difference equation (12) we see that

$$x_{t+1} = W_{11}x_t + W_{12}\lambda_t + (R_x\Pi + Q_x)z_t,$$

where I have used, for simplicity of notation, the assumption that $z_{t+1} = \Pi z_t + \epsilon_{t+1}$. As before, we solve this equation by substituting in $\lambda_t = -(P^{22})^{-1}P^{21}x_t + (P^{22})^{-1}\tilde{\lambda}_t$, which gives us

$$x_{t+1} = (P_{11}\Lambda_1 P_{11}^{-1})x_t + W_{12}(P^{22})^{-1}\tilde{\lambda}_t + (R_x\Pi + Q_x)z_t.$$

The difference equation for $\tilde{\lambda}_t$ is

$$E_t\tilde{\lambda}_{t+1} = \Lambda_2\tilde{\lambda}_t + (\Phi_0\Pi + \Phi_1)z_t, \tag{52}$$

where I am using again the AR(1) assumption for z_t and Φ_0 and Φ_1 are defined as in Section 3. We found a solution to this difference equation which was denoted $\tilde{\lambda}_t = \Psi z_t$. This was obtained by inverting (52) and iterating forward

$$\begin{aligned}\tilde{\lambda}_t &= \Lambda_2^{-1}E_t\tilde{\lambda}_{t+1} - \Lambda_2^{-1}(\Phi_0\Pi + \Phi_1)z_t \\ &= -\sum_{j=0}^{\infty}\Lambda_2^{-(j+1)}(\Phi_0\Pi + \Phi_1)\Pi^j z_t.\end{aligned}$$

This last equation implicitly defined Ψ. However, it is easily verified by substitution that a more general solution to (52) is denoted $\tilde{\lambda}_t = \Psi z_t + \xi_t$, where ξ_t is any process for which $E_t\xi_{t+1} = \Lambda_2\xi_t$.

As a result, a general solution for x_{t+1} is given by

$$x_{t+1} = (P_{11}\Lambda_1P_{11}^{-1})x_t + \left[W_{12}(P^{22})^{-1}\Psi + R_x\Pi + Q_x\right]z_t + W_{12}(P^{22})^{-1}\xi_t.$$

Thus, an infinite variety of solutions exists, because there is only one restriction on the form ξ_t should take.

Typically, it is assumed that the solution of interest is bounded in some way. Notice that since Λ_2 consists of eigenvalues greater than one in absolute value, the process ξ_t is expected to explode. I.e. $\lim_{j\to\infty} E_t\xi_{t+j} = \lim_{j\to\infty} E_t\Lambda_2^j\xi_t = \pm\infty$, depending on the sign of ξ_t.

A simple condition which leaves only the solution we found earlier is to require

$$\lim_{j\to\infty} E_t x_{t+j} = 0, \quad \forall\ t.$$

This is only true for the particular solution we found earlier, i.e. for $\xi_t = 0$, for all t. A different method of rendering the solution unique would be to impose a transversality condition on the original optimization problem. The problem in Section 3 would have the transversality condition

$$\lim_{t\to\infty} E_0\beta^t[x_t'A_{xx}x_t + 2x_t'A_{xu}u_t + 2x_t'A_{xz}z_t + 2x_t'B_x'\lambda_t] = 0.$$

If at least one of the eigenvalues in Λ_2 is greater than $\beta^{-1/2}$ in absolute value, then the explosiveness in ξ_t will dominate the term β^t and the transversality condition will not be satisfied. Therefore, requiring that the condition holds eliminates all but the solution with $\xi_t = 0$ for all t.

5.2 Sunspots and Boundedness When $\mathbf{n_i} > \mathbf{n_s}$

In this section we consider a second possibility, that the number of eigenvalues which are less than one in modulus exceeds the number of state variables. We use the usual definitions of Λ_1 and Λ_2, except that in this case Λ_2 has $n_i - n_s$ elements which are less than one in absolute value. This suggests a further partition of Λ_2 as

$$\Lambda_2 = \begin{pmatrix} \Lambda_{2\ell} & 0 \\ 0 & \Lambda_{2g} \end{pmatrix},$$

where the elements of $\Lambda_{2\ell}$ are less than one in modulus and the elements of Λ_{2g} are greater than one in modulus. In this case, it is convenient to write (52) in two parts

$$\begin{aligned} E_t\tilde{\lambda}_{\ell t+1} &= \Lambda_{2\ell}\tilde{\lambda}_{\ell t} + \Omega_\ell(\Phi_0\Pi + \Phi_1)z_t, & (53)\\ E_t\tilde{\lambda}_{gt+1} &= \Lambda_{2g}\tilde{\lambda}_{gt} + \Omega_g(\Phi_0\Pi + \Phi_1)z_t, & (54)\end{aligned}$$

where $\tilde{\lambda}_{\ell t}$ is simply the first $n_i - n_s$ elements of $\tilde{\lambda}_t$ and $\tilde{\lambda}_{gt}$ is the last $n_{cs} + n_s - n_i$ elements. Furthermore, we have

$$\Omega_\ell = \begin{pmatrix} I_{n_i-n_s} & 0_{n_i-n_s\times n_{cs}+n_s-n_i} \end{pmatrix} \text{ and } \Omega_g = \begin{pmatrix} 0_{n_{cs}+n_s-n_i\times n_i-n_s} & I_{n_{cs}+n_s-n_i} \end{pmatrix}.$$

In this case, we iterate backward on (53) and forward on (54) to obtain the general solutions

$$\begin{aligned} \tilde{\lambda}_{\ell t} &= \sum_{j=0}^{\infty}(\Lambda_{2\ell})^j\Omega_\ell(\Phi_0\Pi + \Phi_1)z_{t-1-j} + \xi_{\ell t} & (55)\\ \tilde{\lambda}_{gt} &= -\sum_{j=0}^{\infty}(\Lambda_{2g})^{-(j+1)}\Omega_g(\Phi_0\Pi + \Phi_1)\Pi z_t + \xi_{gt} \\ &= \Psi_g z_t + \xi_{gt} & (56)\end{aligned}$$

where we have the restrictions $E_t\xi_{\ell t+1} = \Lambda_{2\ell}\xi_{\ell t}$ and $E_t\xi_{gt+1} = \Lambda_{2g}\xi_{gt}$. This gives us a general solution for x_{t+1} of the form

$$x_{t+1} = (P_{11}\Lambda_1 P_{11}^{-1})x_t + W_{12}(P^{22})^{-1}\begin{pmatrix} \sum_{j=0}^{\infty}(\Lambda_{2\ell})^j\Omega_\ell(\Phi_0\Pi + \Phi_1)z_{t-1-j} + \xi_{\ell t} \\ \Psi_g z_t + \xi_{gt} \end{pmatrix} + (R_x\Pi + Q_x)z_t.$$

In this case a boundedness condition on x_{t+1} will allow us to assume that $\xi_{gt} = 0$ for all t. However, the same cannot be said for $\xi_{\ell t}$ since it is not explosive. Therefore, even with boundedness we get left with a multitude of solutions,

$$x_{t+1} = (P_{11}\Lambda_1 P_{11}^{-1})x_t + W_{12}(P^{22})^{-1}\begin{pmatrix} \sum_{j=0}^{\infty}(\Lambda_{2\ell})^j\Omega_\ell(\Phi_0\Pi + \Phi_1)z_{t-1-j} + \xi_{\ell t} \\ \Psi_g z_t \end{pmatrix} + (R_x\Pi + Q_x)z_t.$$

The variables $\xi_{\ell t}$ are referred to as sunspots as they need bear no relationship to the variables in the model (i.e. they are nonfundamental), and need only satisfy the restriction that $E_t\xi_{\ell t+1} = \Lambda_{2\ell}\xi_{\ell t}$.

5.3 Explosiveness When $\mathbf{n_i < n_s}$

If there are fewer eigenvalues less than one in modulus than there are state variables, it is not, in general, possible to derive a solution that satisfies the typical boundedness condition. The general solution to the difference equation governing x_t is

$$x_{t+1} = P_{11}\Lambda_1 P_{11}^{-1}x_t + \left[W_{12}(P^{22})^{-1}\Psi + R_x\Pi + Q_x\right]z_t + W_{12}(P^{22})^{-1}\xi_t,$$

where Ψ is defined as in section 5.1, since in this case, as in that section, it makes sense to solve forward for all elements of $\tilde{\lambda}_t$. Simple iteration on this equation indicates that

$$\begin{aligned} E_t x_{t+j} \;=\;& P_{11}\Lambda_1^j P_{11}^{-1}x_t + \sum_{i=0}^{j-1} P_{11}\Lambda_1^i P_{11}^{-1}\left\{\left[W_{12}(P^{22})^{-1}\Psi + R_x\Pi + Q_x\right]\Pi^{j-1-i}z_t + \right.\\ &\left. W_{12}(P^{22})^{-1}\Lambda_2^{j-1-i}\xi_t\right\}.\end{aligned}$$

Regardless of the assumptions made concerning ξ_t, this expression has, in general, no limiting value because Λ_1 contains eigenvalues which are greater than 1 in absolute value.

6 Computing Impulse Response Functions

This section discusses impulse response functions. An impulse response function is a device which is used to describe the response of endogenous variables in a model to innovations in the exogenous variables in the model. In the context of the real business cycle models discussed in Section 4, we might be interested in the response of output, over time, to an innovation in the level of technology. One way of describing that response is to derive the moving average representation of output as a function of current and past innovations in the level of technology.

In the more setting of the general LQ problem of Section 3, the relevant exercise is to derive expressions for all variables of interest in terms of current and lagged innovations in the exogenous shock vector z_t. Recall that the solution for the endogenous state variable x_{t+1} in that problem is given by

$$x_{t+1} = \Upsilon_{xx}x_t + \Upsilon_{xz}z_t.$$

The solution for λ_t is,

$$\lambda_t = \Upsilon_{\lambda x}x_t + \Upsilon_{\lambda z}z_t$$

and the solution for the controls is given by

$$u_t = \Upsilon_{ux}x_t + \Upsilon_{uz}z_t.$$

This means we can write

$$s_{t+1} = Ms_t + \hat{\epsilon}_{t+1}$$

where

$$s_t = \begin{pmatrix} x_t \\ z_t \end{pmatrix} \qquad \hat{\epsilon}_t = \begin{pmatrix} 0 \\ \epsilon_t \end{pmatrix}$$

and

$$M = \begin{pmatrix} \Upsilon_{xx} & \Upsilon_{xz} \\ 0 & \Pi \end{pmatrix}.$$

The vector ϵ_t is the vector of innovations to the exogenous variables. Furthermore, we have

$$\lambda_t = \begin{pmatrix} \Upsilon_{\lambda x} & \Upsilon_{\lambda z} \end{pmatrix} s_t \tag{57}$$

$$u_t = \begin{pmatrix} \Upsilon_{ux} & \Upsilon_{uz} \end{pmatrix} s_t \tag{58}$$

There may be other variables in our model of interest, denoted f_t (such as output, average productivity, investment and the rate of return in the real business cycle models described in Section 4) which are expressible as a linear combination of the controls, states and exogenous variables

$$\begin{aligned} f_t &= F_c u_t + F_x x_t + F_e z_t \\ &= \left[F_c \begin{pmatrix} \Upsilon_{ux} & \Upsilon_{uz} \end{pmatrix} + \begin{pmatrix} F_x & F_e \end{pmatrix}\right] s_t \end{aligned} \tag{59}$$

Therefore it is possible to represent all the variables except the state and exogenous variables in the form

$$\begin{pmatrix} \lambda_t \\ u_t \\ f_t \end{pmatrix} = Hs_t \tag{60}$$

To generate moving average representations for each of the variables notice that

$$\begin{aligned} s_t &= Ms_{t-1} + \hat{\epsilon}_t \\ &= M^2 s_{t-2} + \hat{\epsilon}_t + M\hat{\epsilon}_{t-1} \\ &\vdots \\ &= \hat{\epsilon}_t + M\hat{\epsilon}_{t-1} + M^2\hat{\epsilon}_{t-2} + \cdots \end{aligned}$$

Impulse response functions usually refer to the response of variable i to an innovation in exogenous variable j at time $t-k$, which we might denote $\text{IRF}_{ij}(k)$. So, to measure the response of the ith element of s_t to an innovation in the j element of $\hat{\epsilon}_{t-k}$, we would simply let $\text{IRF}_{ij}(k)$ be the ijth element of M^k. Similarly, to measure the response of the ith element of $\begin{pmatrix} \lambda_t & u_t & f_t \end{pmatrix}$ to the same innovation we would let $\text{IRF}_{ij}(k)$ be the ijth element of HM^k.

Figures 1 and 2 illustrate the impulse response functions for the divisible and indivisible labor models described in Section 4. Each diagram represents the response of the indicated variable to a 1% innovation in the level of technology. MATLAB procedures to compute the impulse response functions are included in the programs that accompany these notes.

7 Computing Autocovariances

In the previous section we derived a representation of the solution to the general LQ problem given by

$$s_{t+1} = Ms_t + \hat{\epsilon}_{t+1} \tag{61}$$

and

$$\begin{pmatrix} \lambda_t \\ u_t \\ f_t \end{pmatrix} = Hs_t$$

To compute the autocovariances of s_t notice that

$$s_t = M^i s_{t-i} + \hat{\epsilon}_t + \ldots + M^{i-1}\hat{\epsilon}_{t-i+1}.$$

Therefore

$$\begin{aligned} \Gamma_i &= E(s_t s'_{t-i}) \\ &= E\left[(M^i s_{t-i} + \hat{\epsilon}_t + \ldots + M^{i-1}\hat{\epsilon}_{t-i+1})s'_{t-i}\right] \\ &= M^i\Gamma_0 \end{aligned}$$

To compute Γ_0, in some cases we can diagonalize $M = VDV^{-1}$, where D is a diagonal matrix with the eigenvalues of M on the diagonal and V is a matrix whose columns are the corresponding $n_s + n_e$ linearly independent eigenvectors. Then pre-multiplying (61) by V^{-1} we get

$$\begin{aligned} V^{-1}s_{t+1} &= DV^{-1}s_t + V^{-1}\hat{\epsilon}_{t+1} \\ \tilde{s}_{t+1} &= D\tilde{s}_t + \tilde{\epsilon}_{t+1} \end{aligned}$$

where the covariance matrix of $\tilde{\epsilon}$ is

$$\tilde{\Sigma} = V^{-1}\begin{pmatrix} 0 & 0 \\ 0 & \Sigma \end{pmatrix} V^{-1\prime}.$$

Notice that the ith element of $\tilde{s}_t$ is given by

$$\begin{aligned} \tilde{s}_{it} &= d_i\tilde{s}_{it-1} + \tilde{\epsilon}_{it} \\ &= \sum_{j=0}^{\infty} d_i^j \tilde{\epsilon}_{it-j} \end{aligned}$$

Therefore $\tilde{\Gamma}_0$, the covariance matrix of $\tilde{s}_t$, is a matrix whose ij element is given by

$$\frac{1}{1 - d_i d_j}\tilde{\Sigma}_{ij}.$$

The matrix $\Gamma_0 = V\tilde{\Gamma}_0 V'$. In some cases, M may be singular and the diagonalization cannot be performed. For example, if one of the shocks in z_t is white noise, Π, will be singular, rendering M singular. In cases for which M is singular, Γ_0 can be computed by some finite approximation to $\Sigma + M\Sigma M' + M^2\Sigma(M')^2 + \cdots$.

We also have the covariance

$$\begin{aligned} E\left[s_t \begin{pmatrix} \lambda_t \\ u_t \\ f_t \end{pmatrix}' \right] &= E(s_t s_t' H') \\ &= \Gamma_0 H'. \end{aligned}$$

These simple calculations show that it is straightforward to compute any of the autocovariances among x_t, z_t, λ_t, u_t or f_t.

7.1 The Hodrick-Prescott Filter

The literature contains many papers which compare the predictions of a model for the properties of Hodrick-Prescott (HP) filtered time series to the properties of corresponding time series in the data. This filter was proposed by Hodrick and Prescott (1997). It decomposes the logarithm of a time series, denoted y_t, into two components, the growth component, g_t, and the cyclical component, c_t. That is $c_t + g_t = y_t$. The way they determine this decomposition is to choose $\{g_t\}_{t=1}^{T}$ to minimize

$$\sum_{t=1}^{T}(y_t - g_t)^2 + \lambda\sum_{t=2}^{T-1}(g_{t+1} - 2g_t + g_{t-1})^2.$$

Notice that the first order conditions, for $g_1, g_2, \ldots$, from this minimization are

$$\begin{aligned}
-y_1 + g_1 + \lambda(g_3 - 2g_2 + g_1) &= 0 \\
-y_2 + g_2 + \lambda[(-2g_3 + 4g_2 - 2g_1) + (g_4 - 2g_3 + g_2) &= 0 \\
-y_t + g_t + \lambda[(g_t - 2g_{t-1} + g_{t-2}) + (-2g_{t+1} + 4g_t - 2g_{t-1}) + (g_{t+2} - 2g_{t+1} + g_t)] &= 0 \\
t = 3, \ldots T-2 & \\
-y_{T-1} + g_{T-1} + \lambda[(-2g_T + 4g_{T-1} - 2g_{T-2}) + (g_{T-1} - 2g_{T-2} + g_{T-3}) &= 0 \\
-y_T + g_T + \lambda(g_T - 2g_{T-1} + g_{T-2}) &= 0
\end{aligned}$$

If you write the time series $\{y_t\}_{t=1}^T$ as a vector, y, and the growth components $\{g_t\}_{t=1}^T$ as a vector, g, the first order conditions for the growth components are given by $Ag = y$, or $g = A^{-1}y$, and the cyclical component is $c = y - g$. The matrix A is equal to

$$\begin{pmatrix}
1+\lambda & -2\lambda & \lambda & 0 & 0 & \cdots & 0 & 0 & 0 \\
-2\lambda & 1+5\lambda & -4\lambda & \lambda & 0 & \cdots & 0 & 0 & 0 \\
\lambda & -4\lambda & 1+6\lambda & -4\lambda & \lambda & \cdots & 0 & 0 & 0 \\
0 & \lambda & -4\lambda & 1+6\lambda & -4\lambda & \cdots & 0 & 0 & 0 \\
0 & 0 & \lambda & -4\lambda & 1+6\lambda & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \cdots & 1+6\lambda & -4\lambda & \lambda \\
0 & 0 & 0 & 0 & 0 & \cdots & -4\lambda & 1+5\lambda & -2\lambda \\
0 & 0 & 0 & 0 & 0 & \cdots & \lambda & -2\lambda & 1+\lambda
\end{pmatrix}$$

Notice that the filter lets you write $g_t = G_t^T(L)y_t$, where $G_t^T(L)$ is a polynomial (possibly double-sided) in the lag-operator. Notice that the coefficients in the polynomial are time-varying and depend on the sample size.

The series created by the HP filter are similar (in large samples) to the time series obtained by choosing the time series $\{g_t\}_{t=-\infty}^{\infty}$ to minimize

$$\sum_{t=-\infty}^{\infty} (y_t - g_t)^2 + \lambda \sum_{t=-\infty}^{\infty} (g_{t+1} - 2g_t + g_{t-1})^2.$$

This minimization problem leads to a time-invariant form of the HP filter. I.e. we get a form $g_t = G(L)y_t$, where $G(L)$ is independent of time. The second moments of cyclical components obtained using the time-varying filter are asymptotically equivalent to the second moments of cyclical components obtained using the time-invariant filter. Therefore, we focus on the time-invariant filter.

The bigger is λ the smoother is the series g_t that you get out of the filter. The number usually used for quarterly data is $\lambda = 1600$. King and Rebelo (1989) describe the form of the time-invariant filter. They show that the filtered series c_t is given by

$$\begin{aligned}
[\lambda^{-1}L^2 + (1-L)^4]c_t &= (1-L)^4 y_t \\
R(L)c_t &= (1-L)^4 y_t
\end{aligned}$$

King and Rebelo show that this can be written as

$$c_t = H(L)(1-L)^4 y_t$$

where

$$\begin{aligned} H(L) &= (|r|^2L^{-2})\left[(1-rL)^{-1}(1-\bar{r}L)^{-1}(1-rL^{-1})^{-1}(1-\bar{r}L^{-1})^{-1}\right] \\ &= (|r|^2L^{-2})\left[1-2\mathrm{Re}(r)L+|r|^2L^2\right]^{-1}\left[1-2\mathrm{Re}(r)L^{-1}+|r|^2L^{-2}\right]^{-1}. \end{aligned}$$

The parameter r is the reciprocal of a stable root of $R(Z)=0$ (these are all complex as shown by King and Rebelo 1989). Re(r) is the real part of r, while $\bar{r}$ is its complex conjugate. King and Rebelo (1989) show that this can be written in the form of an infinite symmetric double-sided filter,

$$c_t = B(L)y_t = (\ldots + b_{-2}L^{-2} + b_{-1}L^{-1} + b_0 + b_1L + b_2L^2 + \ldots)y_t,$$

where $b_0 = 1-[.0561\cos(0)+.0558\sin(0)]$, $b_{-j} = b_j = -(0.894^j)[.0561\cos(.112j)+.0558\sin(.112j)]$ when $\lambda = 1600$.

To get a good understanding of what the HP filter does to data you need a basic understanding of spectral analysis. The spectrum of a time series is a function which depends on the autocovariances of that time series. Suppose we have a covariance stationary, mean zero, time series x_t. The ith autocovariance is defined as

$$\gamma_i = E(x_t x_{t-i}) \qquad i = 0, 1, \ldots \infty.$$

The spectrum of a time series is a function of the frequency ω

$$f(\omega) = \frac{1}{2\pi}\left[\gamma_0 + 2\sum_{i=1}^{\infty}\gamma_i\cos(\omega i)\right], \qquad \omega \in [0,\pi].$$

Frequencies are mapped into periods using the mapping $p = 2\pi/\omega$. The spectrum roughly tells you how much of the variance of the time series occurs in components with different frequencies. If you take any covariance stationary time series it has a spectrum given by the formula above. In Figure 3, the spectra of four time series are plotted. These time series are

- a *white noise* process, $x_t = \epsilon_t$, where $E\epsilon_t^2 = \sigma^2$ and $E(\epsilon_t\epsilon_{t-i}) = 0$, for $i \neq 0$.
- a *first order autoregressive*, AR(1), process $x_t = \rho x_{t-1} + \epsilon_t$, where $\rho = 0.8$,
- a *first order moving average*, MA(1), process $x_t = \epsilon_t + \theta\epsilon_{t-1}$, where $\theta = 0.5$,
- a seasonal AR(4) process, $x_t = \phi x_{t-4} + \epsilon_t$, with $\phi = 0.8$.

In each case, the spectra are plotted for $\sigma^2 = 1$.

Now suppose a filter $B(L)$ is applied to x_t. That is, define

$$y_t = B(L)x_t.$$

The spectrum of y_t is given by

$$f_y(\omega) = |B(\omega)|^2 f_x(\omega),$$

where $B(\omega) = \sum_{j=-\infty}^{\infty} b_j e^{-i\omega j}$ and $i = \sqrt{-1}$. Using this information you can simply plot the *transfer function*, $|B(\omega)|^2$, that goes with any filter in order to see how different components of the variation in x_t are affected by the application of the filter. The transfer function of the time-invariant HP filter is given in Figure 4 for different values of λ. For the sake of comparison, Figure 4 also contains a graph of the transfer function for the first difference filter, $\Delta = (1 - L)$.

A useful property of the time-invariant HP filter is that the cyclical component does not change when anything up to a quadratic deterministic trend is added to the series being filtered, y_t. To see this, define $x_t = y_t + \tau_t$, where $\tau_t = \alpha + \beta t + \gamma t^2$. The growth component of x_t is determined by

$$\left[\lambda L^2 - 4\lambda L + (1+6\lambda) - 4\lambda L^{-1} + \lambda L^{-2}\right] g_{xt} = x_t = y_t + \tau_t,$$

while the growth component of y_t is determined by

$$\left[\lambda L^2 - 4\lambda L + (1+6\lambda) - 4\lambda L^{-1} + \lambda L^{-2}\right] g_{yt} = y_t.$$

These equations combine to yield

$$\left[\lambda L^2 - 4\lambda L + (1+6\lambda) - 4\lambda L^{-1} + \lambda L^{-2}\right] (g_{xt} - g_{yt}) = \tau_t.$$

Since $x_t = y_t + \tau_t = c_{xt} + g_{xt}$ and $y_t = c_{yt} + g_{yt}$, we have $g_{xt} - g_{yt} = \tau_t + c_{yt} - c_{xt}$. Therefore,

$$\left[\lambda L^2 - 4\lambda L + (1+6\lambda) - 4\lambda L^{-1} + \lambda L^{-2}\right] (c_{yt} - c_{xt}) + \left[\lambda L^2 - 4\lambda L + 6\lambda - 4\lambda L^{-1} + \lambda L^{-2}\right] \tau_t = 0.$$

Showing that $c_{xt} = c_{yt}$ depends on showing that

$$\left[\lambda L^2 - 4\lambda L + 6\lambda - 4\lambda L^{-1} + \lambda L^{-2}\right] \tau_t = 0,$$

for all t. Some algebra shows that this requires showing that

$$\begin{aligned}&[\alpha + \beta(t-2) + \gamma(t-2)^2] - 4[\alpha + \beta(t-1) + 4\gamma(t-1)^2] + 6(\alpha + \beta t + \gamma t^2) \\ &\qquad -4[\alpha + \beta(t+1) + 4\gamma(t+1)^2] + [\alpha + \beta(t+2) + \gamma(t+2)^2] \;=\; 0.\end{aligned}$$

More algebra confirms that these equalities hold. This property is not shared by the time-varying filter, although it is invariant to anything up to a linear trend.

This property of the HP filter is useful because it allows a simple mapping from the moments implied by the model for the hatted variables to the moments implied by the model for HP filtered data. Take for example, $\hat{y}_t$. It was argued in Section 4 that $\hat{y}_t \approx \ln(y_t/y) = \ln[(Y_t/X_t)/y] = \ln(Y_t) - \ln(y) - \ln(X_t) = \ln(Y_t) - \ln(y) - \ln(X_0) - t\ln(\gamma_X)$. Therefore, the moments of the hatted variables are directly comparable to the moments of linearly detrended data. The invariance of the cyclical component to the inclusion or exclusion of a trend suggests that the moments of HP filtered hatted variables can be compared directly to the moments of HP filtered data. The next step is to determine the mapping from the moments of the hatted variables to the moments of HP filtered data.

Let x_t denote a vector of hatted time series and let $E(x_t x_{t-i}) = \Gamma_i$. Then HP filtered series corresponding to x_t are given by $y_t = \sum_{j=-\infty}^{+\infty} b_j x_{t-j}$, where the b_j are defined above. The autocovariances of y_t are given by

$$\begin{aligned} E(y_t y'_{t-i}) &= E\left[\sum_{j=-\infty}^{+\infty} b_j x_{t-j} \sum_{j=-\infty}^{+\infty} b_j x'_{t-i-j}\right] \\ &= \Gamma_0 \sum_{j=-\infty}^{+\infty} b_j b_{j-i} + \sum_{k=1}^{\infty} \Gamma_k \sum_{j=-\infty}^{+\infty} b_j b_{j+k-i} + \sum_{k=1}^{\infty} \Gamma'_k \sum_{j=-\infty}^{+\infty} b_j b_{j-k-i} \end{aligned}$$

Suppose we used a finite number of the b_j in approximating this sum, say b_j for $j = -N$ to $j = N$ and a finite number of the autocovariances, Γ_k, $k = 0, \ldots, M$. The autocovariances of y_t could be approximated by

$$\begin{aligned} \sum_{j=-N+i}^{N} b_j b_{j-i} \Gamma_0 \quad &+ \quad \sum_{k=1}^{i} \sum_{j=-N+i-k}^{N} b_j b_{j+k-i} \Gamma_k + \sum_{k=1}^{i} \sum_{j=-N+i+k}^{N} b_j b_{j-k-i} \Gamma'_k + \\ & \sum_{k=i+1}^{M} \sum_{j=-N+k-i}^{N} b_j b_{j+i-k} \Gamma_k + \sum_{k=i+1}^{M} \sum_{j=-N+k+i}^{N} b_j b_{j-i-k} \Gamma'_k . \end{aligned}$$

In order for this formula to be accurate, N should be considerably larger than M which should be considerably larger than i.

It is much easier to derive the autocovariances of first-differenced series. For example, $\Delta\hat{y}_t \approx \Delta\ln(y_t/y) = \ln(Y_t) - \ln(Y_{t-1}) - \ln(\gamma_X)$. If the autocovariances of x_t are given by Γ_i, then it is easy to show that the autocovariances of Δx_t are given by

$$\begin{aligned} E(\Delta x_t \Delta x_{t-i}) &= E(x_t x_{t-i}) - E(x_t x_{t-i-1}) - E(x_{t-1} x_{t-i}) + E(x_{t-1} x_{t-i-1}) \\ &= 2\Gamma_i - \Gamma_{i-1} - \Gamma_{i+1} \qquad i > 0 \\ &= 2\Gamma_0 - \Gamma_1 - \Gamma'_1 \qquad i = 0 \end{aligned}$$

These are directly comparable to the autocovariances of first-differenced data.

8 Models With Stochastic Growth

Some RBC models specify the labour augmenting growth component, X_t, as a stochastic process. Such stochastic trends can be accommodated in a similar fashion to the way deterministic trends are handled. However, some differences arise, especially when computing moments of interest. Take, for example, one of Christiano and Eichenbaum's (1992) models. They considered a model with *indivisible labor* in which the government removes a stochastic quantity of resources in a lump-sum manner. Government expenditure yields no utility to private consumers. The social planner's problem in their model is to maximize

$$\mathcal{L} = E_0 \sum_{t=0}^{\infty} \left\{ \beta^t \left[\ln(C_t) + \theta(T - N_t)\right] + \Lambda_t \left[K_t^{1-\alpha}(N_t X_t)^{\alpha} + (1-\delta)K_t - K_{t+1} - C_t - G_t\right] \right\}.$$

Government expenditure is exogenous, as is X_t. The Euler equations are

$$\begin{aligned}
\beta^t C_t^{-1} - \Lambda_t &= 0 \\
-\beta^t \theta + \Lambda_t \alpha K_t^{1-\alpha} N_t^{\alpha-1} X_t^{\alpha} &= 0 \\
-\Lambda_t + E_t \Lambda_{t+1} \Big[(1-\alpha) K_{t+1}^{-\alpha} (N_{t+1} X_{t+1})^{\alpha} + (1-\delta) \Big] &= 0 \\
K_t^{1-\alpha} (N_t X_t)^{\alpha} + (1-\delta) K_t - K_{t+1} - C_t - G_t &= 0
\end{aligned}$$

Christiano and Eichenbaum assume that $X_t = \exp(\gamma + v_t) X_{t-1}$. I.e. the common growth component is stochastic rather than deterministic. The shock v_t is assumed to be a stationary mean zero AR(1) process. I.e. $v_t = \rho_v v_{t-1} + \epsilon_{vt}$. We write the model in terms of detrended series using the following transformations: $c_t = C_t / X_t$, $k_t = K_t / X_{t-1}$, $y_t = Y_t / X_t$, $g_t = G_t / X_t$, $\lambda_t = \beta^{-t} \Lambda_t X_t$. K_t is divided by X_{t-1} so that k_t remains predetermined. The stochastic process for g_t is assumed to be stationary, with $\ln(g_t) = (1-\rho_g)\ln(g) + \rho_g \ln(g_{t-1}) + \epsilon_{gt}$.[6] Rewritten in these terms the Euler equations are

$$\begin{aligned}
c_t^{-1} - \lambda_t &= 0 \\
-\theta + \lambda_t \alpha k_t^{1-\alpha} N_t^{\alpha-1} \exp[(\alpha-1)(\gamma+v_t)] &= 0 \\
-\lambda_t + \beta E_t \lambda_{t+1} \Big[(1-\alpha) k_{t+1}^{-\alpha} N_{t+1}^{\alpha} \exp[(\alpha-1)(\gamma+v_{t+1})] + (1-\delta)\exp(-\gamma - v_{t+1}) \Big] &= 0 \\
k_t^{1-\alpha} N_t^{\alpha} \exp[(\alpha-1)(\gamma+v_t)] + (1-\delta)\exp(-\gamma - v_t) k_t - k_{t+1} - c_t - g_t &= 0
\end{aligned}$$

The nonstochastic steady state is easily obtained. Simply remove all time subscripts and expectations from the Euler equations and set $v_t = 0$. The equations become

$$\begin{aligned}
c^{-1} - \lambda &= 0 \\
-\theta + \lambda \alpha k^{1-\alpha} N^{\alpha-1} \exp[(\alpha-1)\gamma] &= 0 \\
-\lambda + \beta\lambda \Big[(1-\alpha) k^{-\alpha} N^{\alpha} \exp[(\alpha-1)\gamma] + (1-\delta)\exp(-\gamma) \Big] &= 0 \\
k^{1-\alpha} N^{\alpha} \exp[(\alpha-1)\gamma] + (1-\delta)\exp(-\gamma) k - k - c - g &= 0
\end{aligned}$$

Clearly, the steady state detrended output-capital ratio[7] is given by

$$1 = \beta[(1-\alpha)(y/k) + (1-\delta)\exp(-\gamma)]. \tag{62}$$

Therefore,

$$(y/k) = [\beta^{-1} - (1-\delta)\exp(-\gamma)]/(1-\alpha).$$

Similarly, the capital-labor ratio is given by

$$(k/N) = \left[\frac{\beta^{-1} - (1-\delta)\exp(-\gamma)}{(1-\alpha)\exp[(\alpha-1)\gamma]} \right]^{-1/\alpha}.$$

From the Euler equation for N this determines λ which determines c. The resource constraint implies that

$$y = [1 - (1-\delta)\exp(-\gamma)]k + c + g$$

[6] The accompanying program for this model assumes that all state variables are rescaled by dividing by X_{t-1} and that all exogenous shocks are defined relative to the common trend X_t.

[7] The non-detrended output-capital ratio is $(y/k)\exp(\gamma)$ because $Y_{t+1}/K_{t+1} = (y_{t+1}X_{t+1})/(k_{t+1}X_t)$.

or

$$y\left(1-[1-(1-\delta)\exp(-\gamma)]\,\frac{k}{y}\right)=c+g.$$

Since g is a parameter, this determines y, which determines k, which then determines N.

To linearize the Euler equations proceed as in Section 4. Defining $\hat{v}_t = dv_t$, since v_t is already measured in logarithms, and $\mu = \beta(1-\alpha)(y/k)$, we obtain

$$\begin{aligned}
-\hat{c}_t-\hat{\lambda}_t &= 0\\
\hat{\lambda}_t+(1-\alpha)\hat{k}_t+(\alpha-1)\hat{N}_t+(\alpha-1)\hat{v}_t &= 0\\
-\hat{\lambda}_t+\hat{\lambda}_{t+1}-\alpha\mu\hat{k}_{t+1}+\alpha\mu\hat{N}_{t+1}+(\alpha\mu-1)\hat{v}_{t+1} &= 0\\
(1-\alpha)\hat{k}_t+\alpha\hat{N}_t-[(1-\alpha)+(1-\delta)\exp(-\gamma)(k/y)]\hat{v}_t+ &\\
(1-\delta)\exp(-\gamma)(k/y)\hat{k}_t-(k/y)\hat{k}_{t+1}-(c/y)\hat{c}_t-(g/y)\hat{g}_t &= 0
\end{aligned}$$

The additional variables y_t, w_t, i_t, R^k_t and R_t are given by

$$\begin{aligned}
y_t &= k_t^{1-\alpha}N_t^{\alpha}\exp[(\alpha-1)(\gamma+v_t)]\\
w_t &= k_t^{1-\alpha}N_t^{\alpha-1}\exp[(\alpha-1)(\gamma+v_t)]\\
i_t &= k_t^{1-\alpha}N_t^{\alpha}\exp[(\alpha-1)(\gamma+v_t)]-c_t-g_t\\
R^k_t &= (1-\alpha)K_t^{-\alpha}(N_tX_t)^{\alpha}=(1-\alpha)k_t^{-\alpha}N_t^{\alpha}\exp[\alpha(\gamma+v_t)]+(1-\delta)\\
R_t &= \left(\beta E_t\frac{C_t}{C_{t+1}}\right)^{-1}=\frac{exp(\gamma)}{\beta}\left[E_t\exp(-v_{t+1}\frac{c_t}{c_{t+1}}\right]^{-1}
\end{aligned}$$

The steady state value of R^k_t is $R^k=(1-\alpha)(y/k)\exp(\gamma)+(1-\delta)$. The steady state value of R_t is given by $R=\exp(\gamma)/\beta$. From (62) it follows that $R^k=R$. These equations linearized are given by

$$\begin{aligned}
\hat{y}_t &= (1-\alpha)\hat{k}_t+\alpha\hat{N}_t+(\alpha-1)\hat{v}_t\\
\hat{w}_t &= (1-\alpha)\hat{k}_t+(\alpha-1)\hat{N}_t+(\alpha-1)\hat{v}_t\\
\hat{i}_t &= (1-\alpha)(y/i)\hat{k}_t+\alpha(y/i)\hat{N}_t+(\alpha-1)(y/i)\hat{v}_t-(y/i)(c/y)\hat{c}_t-(y/i)(g/y)\hat{g}_t\\
\hat{R}^k_t &= -\alpha\mu\hat{k}_t+\alpha\mu\hat{N}_t+\alpha\mu\hat{v}_t\\
\hat{R}_t &= E_t\hat{c}_{t+1}-\hat{c}_t,
\end{aligned}$$

where we have used the fact that $i=[1-(1-\delta)\exp(-\gamma)]k$.

Once we have the linearized equations in this form, it is straightforward to solve the linearized model using the methods of Section 3. However, given the definition of the transformed variables, the mapping from the moments of the hatted variables to the moments of HP filtered, or first-differenced series is not the same as the mapping used in Section 7.

As an example, consider the first difference of output, given by $\Delta\ln(Y_t)=\Delta\ln(y_tX_t)=\gamma+\Delta\ln(y_t)+v_t$. Given this, the autocovariances of $\Delta\ln(Y_t)$ can be approximated by the moments of $\Delta\hat{y}_t+\hat{v}_t$.

The HP-filtered series for output would be $c_{y,t}=B(L)\ln(Y_t)=\tilde{B}(L)\Delta\ln(Y_t)$, where $\tilde{b}_i=\sum_{j=-\infty}^{i}b_j$. Therefore, both the moments of log first-differenced data implied by the model and the moments of HP-filtered data implied by the model can be constructed using the autocovariances of the hatted variables.

Suppose we were interested in the moments of HP filtered data from the model. A simple approach is to first derive the moments of the first-differenced variables. Recall the solution for the state variable given by equation (15)

$$x_t = \Upsilon_{xx} x_{t-1} + \Upsilon_{xz} z_{t-1}.$$

Notice that in the Christiano and Eichenbaum model the variable x_t is interpreted as $\ln(K_t/X_{t-1}) - \ln(k)$. I.e. it is the percentage deviation of the capital stock, relative to the trend, from its nonstochastic steady state value. Therefore, the second moments of $\Delta \ln(K_t)$ would correspond to the second moments of $\Delta x_t + \Delta \ln(X_{t-1})$. Since $\Delta \ln(X_{t-1})$ is simply $\gamma + v_{t-1}$, it is clear that these second moments would correspond to the second moments of $\Delta x_t + v_{t-1}$. To compute these moments for general models it is convenient to define

$$\bar{s}_t = \begin{pmatrix} \Delta x_t + z_{1,t-1} \\ z_t \\ z_{t-1} \end{pmatrix},$$

where I am assuming that the shock to the stochastic trend (in our example v_t) is always defined as the first element in the vector of exogenous variables. There are two useful sets of equations involving $\bar{s}_t$. First, define a $n_s \times n_e$ matrix Θ_x whose first column is a vector of ones, and whose other elements are 0. I.e. define,

$$\Theta_x = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \dots & 0 \end{pmatrix}.$$

Then

$$\begin{aligned} \bar{s}_t &= \begin{pmatrix} x_t - x_{t-1} + \Theta_x z_{t-1} \\ z_t \\ z_{t-1} \end{pmatrix} \\ &= \begin{pmatrix} I & 0 \\ 0 & I \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_t \\ z_t \end{pmatrix} + \begin{pmatrix} -I & \Theta_x \\ 0 & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} x_{t-1} \\ z_{t-1} \end{pmatrix} \\ &= B_0 s_t + B_1 s_{t-1} \end{aligned} \tag{63}$$

Second, note that

$$\begin{aligned} \bar{s}_t &= \begin{pmatrix} \Upsilon_{xx} & \Upsilon_{xz} + \Theta_x & -(\Upsilon_{xz} + \Upsilon_{xx}\Theta_x) \\ 0 & \Pi & 0 \\ 0 & I & 0 \end{pmatrix} \bar{s}_{t-1} + \begin{pmatrix} 0 \\ \epsilon_t \\ 0 \end{pmatrix} \\ &= \bar{M} \bar{s}_{t-1} + \bar{\epsilon}_t. \end{aligned} \tag{64}$$

Notice that (63) implies that

$$\bar{\Gamma}_0 = E(\bar{s}_t \bar{s}_t') = B_0 \Gamma_0 B_0' + B_1 \Gamma_0 B_1' + B_0 M \Gamma_0 B_1' + B_1 \Gamma_0' M' B_0', \tag{65}$$

where Γ_0 and M are defined as in Section 7. Furthermore, (64) can be used to show that

$$\bar{\Gamma}_i = E(\bar{s}_t \bar{s}'_{t-i}) = \bar{M}^i \bar{\Gamma}_0. \tag{66}$$

Some care has to be taken, in order to compute covariances involving the other variables of interest. These might include, $\Delta \ln(K_t)$, $\Delta \ln(X_t)$, $\Delta \ln(G_t)$, $\Delta \ln(\Lambda_t)$, $\Delta \ln(C_t)$, $\Delta \ln(N_t)$, $\Delta \ln(Y_t)$, $\Delta \ln(Y_t/N_t)$, $\Delta \ln(I_t)$, $\Delta \ln(R^k_t)$ and $\Delta \ln(R_t)$. Define

$$\Delta \bar{F}_t = \begin{pmatrix} \Delta \ln(K_t) \\ \Delta \ln(X_t) \\ \Delta \ln(G_t) \\ \Delta \ln(\Lambda_t) \\ \Delta \ln(C_t) \\ \Delta \ln(N_t) \\ \Delta \ln(Y_t) \\ \Delta \ln(Y_t/N_t) \\ \Delta \ln(I_t) \\ \Delta \ln(R^k_t) \\ \Delta \ln(R_t) \end{pmatrix} = \begin{pmatrix} \Delta \hat{k}_t + \Delta \ln(X_{t-1}) \\ \Delta \ln(X_t) \\ \Delta \hat{g}_t + \Delta \ln(X_t) \\ \Delta \hat{\lambda}_t - \Delta \ln(X_t) \\ \Delta \hat{c}_t + \Delta \ln(X_t) \\ \Delta \hat{N}_t \\ \Delta \hat{y}_t + \Delta \ln(X_t) \\ \Delta \hat{w}_t + \Delta \ln(X_t) \\ \Delta \hat{i}_t + \Delta \ln(X_t) \\ \Delta \hat{R}^k_t \\ \Delta \hat{R}_t \end{pmatrix} = \begin{pmatrix} \Delta \hat{k}_t + \gamma + v_{t-1} \\ 0 \\ \Delta \hat{g}_t \\ \Delta \hat{\lambda}_t \\ \Delta \hat{c}_t \\ \Delta \hat{N}_t \\ \Delta \hat{y}_t \\ \Delta \hat{w}_t \\ \Delta \hat{i}_t \\ \Delta \hat{R}^k_t \\ \Delta \hat{R}_t \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 1 \\ -1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} (\gamma + v_t)$$

Ignoring the constant γ, this can be written as

$$\Delta \bar{F}_t = \begin{pmatrix} \Delta \hat{k}_t + v_{t-1} \\ v_t \\ \Delta \hat{g}_t + v_t \\ \Delta f_t + \zeta v_t \end{pmatrix},$$

where $\zeta = \begin{pmatrix} -1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}'$ and $f_t = \begin{pmatrix} \hat{\lambda}_t & \hat{c}_t & \hat{N}_t & \hat{y}_t & \hat{w}_t & \hat{i}_t & \hat{R}^k_t & \hat{R}_t \end{pmatrix}'$. Our solution method implies that f_t can be written as $f_t = H_1 x_t + H_2 z_t$. Therefore,

$$\Delta \bar{F}_t = \begin{pmatrix} \Delta x_t + \Theta_x z_{t-1} \\ \Xi_1 z_t \\ \Xi_2 z_t + \Xi_3 z_{t-1} \\ H_1 \Delta x_t + (H_2 + \Theta_f) z_t - H_2 z_{t-1} \end{pmatrix},$$

where Ξ_1 is a $1 \times n_e$ vector whose first element is 1 and whose other elements are 0, Ξ_2 is a $(n_e - 1) \times n_e$ matrix whose first column is a vector of ones and whose remaining block is an identity, Ξ_3 is a $(n_e - 1) \times n_e$ matrix whose first column is a vector of zeros and whose remaining block is the negative of an identity and Θ_f is a $n_f \times n_e$ matrix whose first column is the vector ζ and whose remaining elements are zero. I.e.

$$\Xi_1 = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix} \qquad \Xi_2 = \begin{pmatrix} 1 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

$$\Xi_3 = \begin{pmatrix} 0 & -1 & 0 & \cdots & 0 \\ 0 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -1 \end{pmatrix} \qquad \Theta_f = \begin{pmatrix} \zeta & 0 \end{pmatrix}.$$

This enables us to write

$$\begin{aligned} \Delta\bar{F}_t &= \begin{pmatrix} I & 0 & 0 \\ 0 & \Xi_1 & 0 \\ 0 & \Xi_2 & \Xi_3 \\ H_1 & (H_2+\Theta_f) & -(H_2+H_1\Theta_x) \end{pmatrix} \begin{pmatrix} \Delta\bar{x}_t \\ z_t \\ z_{t-1} \end{pmatrix} \\ &= \bar{H}\bar{s}_t \end{aligned} \tag{67}$$

Thus we can compute the autocovariances of $\Delta\bar{F}_t$ quite easily as

$$E\left[\Delta\bar{F}_t\Delta\bar{F}'_{t-i}\right] = \bar{H}\Gamma_i\bar{H}'.$$

In order to obtain the moments of HP filtered data, we can simply obtain the impact of applying the filter $\tilde{B}(L)$ described above to the vector $\Delta\bar{F}_t$.[8] This will be the straightforward analog of the method described in Section 7 for the HP filter $B(L)$.

9 Estimation and Hypothesis Testing Using GMM

9.1 Estimation

There are many methods which can be applied to the estimation of real business cycle models. These notes focus on GMM estimation as described by Hansen (1982) and as implemented for RBC models by Christiano and Eichenbaum (1992) among others. GMM estimators are quite general, and include many of the common estimators we are familiar with. Parameter estimates are obtained by exploiting moment restrictions implied by the model to be estimated. In particular, assume that the model in question implies that the following moment conditions hold

$$E\left[f(x_t,\psi)\right] = 0,$$

where ψ is a $k\times 1$ vector, x_t is a vector of stationary time series, and $f(x_t,\psi)$ is $m\times 1$ where $m \geq k$. Since these moment conditions hold in population when the model is true, a natural way to choose an estimate of ψ is to set the sample moment condition to 0. I.e. choose $\hat{\psi}_T$ in order to set

$$\frac{1}{T}\sum_{t=1}^{T} f(x_t,\psi) = 0.$$

Clearly if none of the moment conditions are redundant this is only possible if $m = k$. If $m > k$ we must generalize the estimator. One way to do this is to minimize a quadratic form in the sample moments. I.e. choose $\hat{\psi}_T$ to minimize

$$J_T(\psi) = \left[\frac{1}{T}\sum_{t=1}^{T} f(x_t,\psi)\right]' W_T \left[\frac{1}{T}\sum_{t=1}^{T} f(x_t,\psi)\right],$$

[8] We apply $\tilde{B}(L)$ since we have already first differenced the logarithms of the variables of interest.

where W_T is a $m \times m$ symmetric positive definite *weighting* matrix. The first-order condition for the minimization problem is

$$\frac{\partial J_T(\psi)}{\partial \psi} = \left[\frac{1}{T}\sum_{t=1}^{T}\frac{\partial f(x_t,\psi)}{\partial \psi'}\right]' W_T \left[\frac{1}{T}\sum_{t=1}^{T} f(x_t,\psi)\right] = 0.$$

You can see that the result of satisfying the first-order condition is to set k linear combinations of the m sample moment conditions equal to zero.

Let the true value of the parameter vector be given by ψ_0. Define the $m \times k$ matrix

$$D_0 = E\left[\frac{\partial f(x_t,\psi_0)}{\partial \psi'}\right]$$

and let W_T converge almost surely to the symmetric positive definite matrix W_0. Then Hansen (1982) shows that $\hat{\psi}_T$ is consistent and that

$$\sqrt{T}(\hat{\psi}_T - \psi_0) \xrightarrow{d} N\left[0, (D_0'W_0D_0)^{-1}D_0'W_0S_0W_0D_0(D_0'W_0D_0)^{-1}\right].$$

The matrix S_0 is given by

$$S_0 = E\left[\sum_{j=-\infty}^{\infty} f(x_t,\psi_0)f(x_{t+j},\psi_0)'\right].$$

Hansen (1982) shows that across all GMM estimators which exploit the same moment restrictions, the asymptotic variance-covariance matrix is smallest when W_T converges almost surely to S_0^{-1}. This suggests the following two step procedure.

- Estimate ψ using any positive definite weighting matrix, for example the identity matrix. Since this estimated parameter vector, $\hat{\psi}_T^1$, is consistent it can be used to construct a consistent estimator of S_0. There are many methods that can be used to estimate S_0. Several of these methods are roughly of the form,

$$\hat{S}_T = \sum_{j=-n}^{n} k(j,n)\frac{1}{T}\sum_{t=1}^{T}\left[f(x_t,\hat{\psi}_T^1)f(x_{t+j},\hat{\psi}_T^1)'\right]$$

  where $k(j,n)$ takes several forms depending on the method.[9]

- Setting $W_T = \hat{S}_T^{-1}$ re-estimate ψ. The estimator $\hat{\psi}_T$ has the following asymptotic distribution

$$\sqrt{T}(\hat{\psi}_T - \psi_0) \xrightarrow{d} N\left[0, (D_0'S_0^{-1}D_0)^{-1}\right].$$

  The asymptotic variance-covariance matrix can be estimated by computing

$$V_T = (D_T'\hat{S}_T^{-1}D_T)^{-1},$$

  where

$$D_T = \frac{1}{T}\sum_{t=1}^{T}\frac{\partial f(x_t,\psi)}{\partial \psi'}.$$

[9] See the Hansen, Heaton and Ogaki (1992) GAUSS GMM programs for the details.

This procedure can be iterated on several times, but it will not change the *asymptotic* distribution of $\hat{\psi}_T$. Hansen (1982) also shows that when this two step procedure is used, the value of the objective function at the optimum has the following asymptotic distribution

$$TJ_T(\hat{\psi}_T) \xrightarrow{d} \chi^2(m-k).$$

This provides a test of the over-identifying restrictions when $m > k$.

In order to estimate a RBC model using GMM, moment restrictions which are sufficient to identify the parameters are required. Consider the *indivisible labor* model. Assume first that we know that $\beta = 1.03^{-.25}$, so that it does not need to be estimated. The parameter vector ψ is given by

$$\psi = \begin{pmatrix} \theta & \delta & \alpha & A_y & \ln(\gamma_X) & \ln(A) & \rho & \sigma \end{pmatrix}'.$$

Consider a simple, exactly identified GMM estimator based on the following moment conditions

$$\begin{aligned}
E\left[\theta - \alpha\frac{Y_t}{N_t C_t}\right] &= 0 && (68)\\
E\left[\delta - 1 + \frac{K_{t+1}}{K_t} - \frac{I_t}{K_t}\right] &= 0 && (69)\\
E\left[1 - \beta\frac{C_t}{C_{t+1}}\left((1-\alpha)\frac{Y_{t+1}}{K_{t+1}} + (1-\delta)\right)\right] &= 0 && (70)\\
E\left[\ln(Y_t) - A_y - \ln(\gamma_X)t\right] &= 0 && (71)\\
E\left[(\ln(Y_t) - A_y - \ln(\gamma_X)t)\frac{t}{T}\right] &= 0 && (72)\\
E\left[\ln(A_t) - \ln(A)(1-\rho) - \rho\ln(A_{t-1})\right] &= 0 && (73)\\
E\left[(\ln(A_t) - \ln(A)(1-\rho) - \rho\ln(A_{t-1}))\ln(A_{t-1})\right] &= 0 && (74)\\
E\left[(\ln(A_t) - \ln(A)(1-\rho) - \rho\ln(A_{t-1}))^2 - \sigma^2\right] &= 0. && (75)
\end{aligned}$$

Data for the series N_t are obtained by using a series on per capita hours per quarter and dividing it by a measure of the time endowment.[10] Data for the series A_t can be created for each value of α using data on Y_t, K_t and N_t and the Cobb-Douglas assumption.

Equation (68) is derived from the Euler equation for employment. Equation (69) is derived from the identity that relates gross investment to capital stocks.[11] Equation (70) is derived from the Euler equation for the capital stock. Equations (71) and (72) identify the parameter γ_X with the average growth rate of output over the sample. The parameter A_y is estimated, but is not used to solve the model. Equations (73), (74) and (75) are derived from the law of motion for technology shocks, A_t.

Since there are 8 moment conditions and 8 parameters, the estimator is exactly identified. The value of the objective function at the optimum will not provide a test of the model. However, it is possible to imagine estimators which are based on over-identifying restrictions.

[10] In the examples estimated in this paper, the time endowment is normalized to 1369 hours, as in Christiano and Eichenbaum (1992).

[11] Formally, neither (68) nor (69) should be employed in estimation. Since these two equations hold without the expectations operator, under the null that the model is true, the matrices S_T and S_0 will be singular.

9.2 Hypothesis Testing

Much of the real business cycle literature focuses on the ability of these models to replicate the time series properties of macroeconomic variables. For the most part, the time series properties of interest have been the autocovariances of output, consumption, investment, average productivity, hours worked, government expenditure and interest rates. In an informal approach, the researcher obtains parameters for a model, solves that model, then either simulates the model or uses the methods suggested above to calculate the autocovariances of the variables in question. The model is judged based on how well these 'model' autocovariances match the corresponding moments from the data.

Christiano and Eichenbaum (1992) suggest a formal statistical metric for measuring how close the model moments are to the data moments. The problem with informal metrics is twofold

- they do not take into account the effect of parameter uncertainty on the judgments being made, and
- they do not take into account sampling uncertainty in the data moments.

The method Christiano and Eichenbaum (1992) suggest is essentially a Wald type testing procedure. They note that the second moments implied by the model can be expressed implicitly as a nonlinear function of the parameter vector ψ. This is clearly true. In Section 4 we showed that the linearized solution depends only on the model's structure, steady state values which are functions of the underlying parameters, and the parameters of the law of motion of the exogenous shocks. Furthermore, in Sections 7 and 8 we showed that the solutions, along with more parameters of the law of motion of the exogenous shocks could be used to determine second moments of linearly detrended, HP filtered or first-differenced data. Therefore, those second moments are just complicated nonlinear functions of the data.

Let the mapping from the parameter vector to the model moments be given by $m(\psi)$. Notice that the function m can be given different definitions depending on how many model moments we are interested in calculating.

In order to compare the model moments to the data moments it is necessary to augment the parameter vector. Redefine the model parameters as ψ_1, and define a new set of parameters, ψ_2, whose true values are the population values of the sample moments which correspond to the model moments in question. For example, suppose the only moments of interest are the mean and the standard deviation of the growth rate of output. Then the parameter vector ψ_2 is given by

$$\psi_2 = \begin{pmatrix} E\Delta\ln(Y_t) \\ \left[E\left([\Delta\ln(Y_t)]^2 - [E\Delta\ln(Y_t)]^2\right)\right]^{1/2} \end{pmatrix}.$$

In general we can imagine that the sample moments are identifiable from a condition of the form $E\left[g(x_t,\psi_2)\right] = 0$, of which the above expression for ψ_2 is an example. To see this, write the expression for ψ_2 as

$$\begin{aligned} E\left[\Delta\ln(Y_t) - \psi_{21}\right] &= 0 \\ E\left[[\Delta\ln(Y_t)]^2 - \psi_{21}^2 - \psi_{22}^2\right] &= 0 \end{aligned}$$

The parameter ψ_{21} represents the mean of $\Delta \ln(Y_t)$, while ψ_{22} represents its standard deviation. The combined parameter vector $\psi = \begin{pmatrix} \psi_1' & \psi_2' \end{pmatrix}'$ can be estimated by GMM by exploiting the moment conditions

$$\begin{aligned} E\left[f(x_t, \psi_1)\right] &= 0 \\ E\left[g(x_t, \psi_2)\right] &= 0 \end{aligned}$$

Suppose the entire parameter vector ψ is estimated by GMM, and its asymptotic variance covariance matrix is given by V. The distance between the 'model' moments and the 'data' moments is given by

$$h(\hat{\psi}_T) = m(\hat{\psi}_{T1}) - \hat{\psi}_{T2}.$$

Since this distance depends on both $\hat{\psi}_{T1}$ and $\hat{\psi}_{T2}$ we would probably be advised to make judgements about whether this distance is large or small depending on how much uncertainty we have about the true values of ψ_1, the model parameters, and ψ_2, the 'data' moments. Assuming that $\sqrt{T}h(\hat{\psi}_T)$ is a $\ell \times 1$ vector, with $\ell < k$, its variance-covariance matrix is given by the $\ell \times \ell$ matrix

$$V_0^h = \frac{\partial h(\psi_0)}{\partial \psi'} V \frac{\partial h(\psi_0)}{\partial \psi'}'.$$

This matrix can be estimated consistently by substituting $\hat{\psi}_T$ for ψ_0 and V_T for V_0. I.e.,

$$V_T^h = \frac{\partial h(\hat{\psi}_T)}{\partial \psi'} V_T \frac{\partial h(\hat{\psi}_T)}{\partial \psi'}'.$$

This suggests that we use the following test statistic

$$\mathcal{W}_T = Th(\hat{\psi}_T)'(V_T^h)^{-1}h(\hat{\psi}_T).$$

Under the null hypothesis that the 'model' moments and the 'data' moments are equal in population $\mathcal{W}_T \xrightarrow{d} \chi^2(\ell)$.

Therefore, by computing $\mathcal{W}_T$ for any pair of 'model' and 'data' moments we wish to compare, we have a simple way of deciding whether the model is consistent with that aspect of the data. If the test statistic is large, we will reject the null hypothesis that the model is capable of reproducing the 'data' moment in question.

10 The Programs

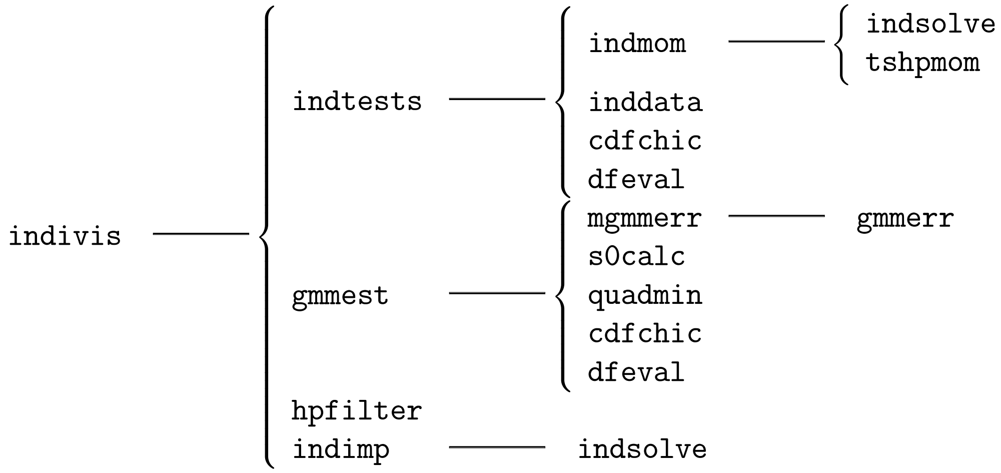
In this section I describe computer programs written in MATLAB that can be used to estimate and test the indivisible labor model, described in Section 4.[12] These programs are located on the disk that accompanies these notes. The user should copy the files and subdirectories on the disk into a directory which can be given any name the user chooses, say `C:\notes`. To run the indivisible labor programs, open MATLAB, and type

[12] The programs were tested in MATLAB 5.2 for Windows 95/NT. They may not run correctly in other versions of MATLAB. Older programs that I wrote for GAUSS, along with notes that describe them, are available upon request.

```
cd c:\notes\indivis
path(path,'c:\notes')
indivis
```

The first line changes directory to the one in which the files for the indivisible labor model are located.[13] The second line adds to MATLAB's internal search path the directory in which files that are necessary for the estimation of any of the models are located. The last line runs the program indivis.m which controls the estimation.

The next several pages dissect the files used in estimating the indivisible labor model to provide the reader with guidance to their use. Therefore, the files themselves do not include many comment lines. In order to differentiate the code from explanatory text, I have set all computer code in the `typewriter font`. The following diagram describes the dependence of various files on each other with m-files being called by a program located to the right of it.



The files `cdfn.m`, `cdfnc.m`, and `cdfchi.m` are not used by any of the other m-files. They are included for the reader's convenience. They compute the left-tail of a normal, the right-tail of a normal, and the the left-tail of a χ^2 distribution respectively. The file `dshpmom.m` is not called by any of the routines used for the indivisible labor example. It is similar to `tshpmom.m` and will be described briefly below.

We begin by describing the m-files to the extreme right, and we will work backward toward indivis.m. The first file is `indsolve.m`, which defines a function taking 5 parameters of the model as arguments and providing output equal to the linearized decision rules. A careful examination of that model shows that the only parameters which matter for the linearized solution are β (`beta_`), α (`alpha_`), $\ln(\gamma_X)$ (`lngamm`), δ (`deltak`) and ρ (`rho`). The next part of the code defines some parameters and some steady state values. `nc` is the number of control variables (c and N), `ns` is the number of state variables (k), `ncs` is the number of co-state variables (λ), `nex` is the number of exogenous variables (A) and `nf` is

[13] Notice that the files for the indivisible labor model are located in a subdirectory called indivis. There are similar files located for the divisible labor model in the subdirectory `divisib`. The directory `ce1992` contains files for Christiano and Eichenbaum's (1992) indivisible labor model with government purchases, and stochastic growth. The directory `ber1993` contains files for Burnside, Eichenbaum and Rebelo's (1993) labor hoarding model. The main directory contains files that are used in estimating any of the models.

the number of additional variables (y, $y-n$, i, r and the interest rate). `mu` is μ defined in Section 4.

```
function mh=indsolve(beta_,alpha_,lngamm,deltak,rho) ;

nc=2 ;
ns=1 ;
ncs=1 ;
nex=1 ;
nf=5 ;

gammax=exp(lngamm) ;
kyratio=(1-alpha_)/(gammax/beta_-(1-deltak)) ;
iyratio=(gammax-(1-deltak))*kyratio ;
cyratio=1-iyratio ;

mu=(gammax-beta_*(1-deltak))/gammax;
muk=-alpha_*mu ;
mun=alpha_*mu ;
```

After this section it is possible to define the $M_{..}$ matrices defined in Section 4, which are used in the solution of the model.

```
mcc = [ -1    0
         0 1-alpha_ ] ;
mcs = [ 0       1
       1-alpha_ 1 ] ;
mce = [ 0
        1 ] ;
mss0 = [ muk            1
        -gammax*kyratio 0 ]  ;
mss1 = [ 0                          -1
        1-alpha_+(1-deltak)*kyratio  0 ] ;
msc0 = [ 0 -mun
         0   0  ] ;
msc1 = [  0        0
        cyratio -alpha_ ] ;
mse0 = [ -mu
          0  ] ;
mse1 = [ 0
        -1 ] ;

fc = [ 0                alpha_
       0                alpha_-1
     -cyratio/iyratio alpha_/iyratio
```

```
       0                mun
      -1                 0   ] ;
fx = [ 1-alpha_
       1-alpha_
       (1-alpha_)/iyratio
       muk
       0  ] ; % to be updated below
fe = [ 1
       1
      1/iyratio
       mu
       0  ] ; % to be updated below
```

In the next section the fundamental difference equation is put in the form

$$\begin{pmatrix} x_{t+1} \\ \lambda_{t+1} \end{pmatrix} = W \begin{pmatrix} x_t \\ \lambda_t \end{pmatrix} + Rz_{t+1} + Qz_t.$$

The eigenvalues and eigenvectors of W are computed. The last line in this section normalizes the eigenvectors so that they have norm 1. The solution is invariant to this normalization, but it helps to scale things.

```
w = -inv(mss0 - msc0*inv(mcc)*mcs)*(mss1 - msc1*inv(mcc)*mcs);
r =  inv(mss0 - msc0*inv(mcc)*mcs)*(mse0 + msc0*inv(mcc)*mce);
q =  inv(mss0 - msc0*inv(mcc)*mcs)*(mse1 + msc1*inv(mcc)*mce);

[pr,lambr]=eig(w);
```

In the next section the eigenvalues and eigenvectors are sorted in ascending order as suggested by the solution method of Section 3. Then the matrices, Λ, P, P^{-1} (`ps`), R and Q are broken up into appropriate sub-blocks.

```
alamb=abs(diag(lambr)) ;
[lambs,lambz]=sort(alamb) ;

lambda=lambr(lambz,lambz) ;
p=pr(:,lambz) ;

lamb1=lambda(1:ns,1:ns) ;
lamb2=lambda(ns+1:ns+ncs,ns+1:ns+ncs) ;

p11=p(1:ns,1:ns) ;
p12=p(1:ns,ns+1:ns+ncs) ;
p21=p(ns+1:ns+ncs,1:ns) ;
p22=p(ns+1:ns+ncs,ns+1:ns+ncs) ;
```

```
ps=inv(p) ;
ps11=ps(1:ns,1:ns) ;
ps12=ps(1:ns,ns+1:ns+ncs) ;
ps21=ps(ns+1:ns+ncs,1:ns) ;
ps22=ps(ns+1:ns+ncs,ns+1:ns+ncs) ;

rxe=r(1:ns,1:nex) ;
rle=r(ns+1:ns+ncs,1:nex) ;

qxe=q(1:ns,1:nex) ;
qle=q(ns+1:ns+ncs,1:nex) ;
```

In the next section the matrices Φ_0 and Φ_1 are defined. Then the matrix Ψ is computed row by row by looping over the rows.

```
phi0=ps21*rxe+ps22*rle ;
phi1=ps21*qxe+ps22*qle ;

psi=zeros(ncs,nex) ;

for i=1:ncs
 psi(i,:)=-(phi0(i,:)*rho+phi1(i,:))*inv(eye(nex)-rho/lamb2(i,i))/lamb2(i,i) ;
end
```

Next, the matrices which express x_{t+1} as a function of x_t and z_t are computed. These are denoted xx and xe respectively. These are combined into the matrix solx which lets you express x_{t+1} as a function of $s_t = \left(\begin{array}{cc} x_t' & z_t' \end{array} \right)'$. Similarly, the solution for λ_t in terms of s_t is given by soll while the solution for the controls is given by solc and the solution for the other variables of interest is given by solf. The interest rate equation is also updated here. The user must ensure that this part of the code remains valid with alternative models.

```
xx=p11*lamb1*inv(p11) ;
xe=(p11*lamb1*ps12+p12*lamb2*ps22)*inv(ps22)*psi+rxe*rho+qxe ;
solx=[ xx xe ] ;

lx=-inv(ps22)*ps21 ;
lex=inv(ps22)*psi ;
soll=[lx lex] ;

cxl=inv(mcc)*mcs ;
ce=inv(mcc)*mce ;
solc=[ cxl*[ eye(ns) ; lx ] cxl*[ zeros(ns,nex) ; lex ]+ce ] ;

% update the interest rate equation
```

```
fx(5,:)=solc(1,1:ns)*xx ;
fe(5,:)=solc(1,1:ns)*xe+solc(1,ns+1:ns+nex)*rho ;

solf=[ fx fe ]+fc*solc ;
```

Finally, the stacked matrices referred to in Section 6, M and H are defined. Then control is returned to the program that called this m-file.

```
m = [ solx ; [ zeros(nex,ns) rho ] ] ;
h = [ soll ; solc ; solf ] ;
mh=[ m ; h ] ;
```

The next m-file `tshpmom.m` computes the implications of a trend stationary model for the moments of HP filtered data. This procedure takes as arguments the covariance matrix of ϵ_t, Σ or `sigma`, `drules` which is simply a large matrix containing M and H, and `ncorr`, the number of autocovariances you want to calculate. At the beginning of the procedure it determines which part of `drules` is M and which is H. Then it defines `sigh` which is just

$$\begin{pmatrix} 0 & 0 \\ 0 & \Sigma \end{pmatrix}.$$

```
function [facv,facr]=tshpmom(sigma,drules,ncorr) ;

[nex,nex2]=size(sigma) ;
[ny,ns]=size(drules) ;

ns=ns-nex ;

m=drules(1:nex+ns,:) ;
h=drules(nex+ns+1:ny,:) ;
sigh=zeros(ns+nex,ns+nex) ;
sigh(ns+1:ns+nex,ns+1:ns+nex)=sigma ;
```

Then, M is diagonalized. Then the matrix $\tilde{\Sigma}$, (`sigt`) is calculated. This quickly leads to the solution for Γ_0, (gamm0).

```
[vr,dr]=eig(m) ;
dr=diag(dr) ;

sigt=inv(vr)*sigh*inv(vr') ;
gammt0=(ones(ns+nex,ns+nex)./(ones(ns+nex,ns+nex)-kron(dr,dr'))).*sigt ;
gamm0=vr*gammt0*(vr') ;
```

The first matrix in the next section, (hbig), defines the stacked vector

$$\left(\begin{array}{cccc} s_t' & \lambda_t' & u_t' & f_t' \end{array} \right)'$$

in terms of s_t. The parameters (base and maxf) are the same as N and M in Section 7 on the HP filter. The vector hpa contains the weights in the HP cyclical filter.

```
hbig=[ eye(ns+nex) ; h ] ;

base=181;
maxf=101;

jp=(1:1:maxf)' ;
jm=(maxf:-1:1)' ;
hpap=-(0.894.^jp).*(0.0561*cos(jp*0.112)+0.0558*sin(jp*0.112)) ;
hpam=-(0.894.^jm).*(0.0561*cos(jm*0.112)+0.0558*sin(jm*0.112)) ;
hpa=[ hpam ; 1-(0.0561*cos(0)+0.0558*sin(0)) ; hpap ] ;
```

The next section does the large double summations used in Section 7 to approximate the autocovariances of HP filtered data. Ultimately the autocovariances for $i = 0, \ldots, \texttt{ncorr}$ are stored in the matrix facv. You can think of this matrix as being of the form

$$\texttt{facv} = \left(\begin{array}{cccc} \Gamma_{y0} & \Gamma_{y1} & \cdots & \Gamma_{y,\texttt{ncorr}} \end{array} \right)$$

where the y subscript just signifies that these are the autocovariances of the stacked vector, $\left(\begin{array}{cccc} s_t' & \lambda_t' & u_t' & f_t' \end{array} \right)'$. As MATLAB passes through these nested loops it is evaluating the autocovariances for lag k. It starts by getting the kth autocovariance of HP filtered s_t with itself. The loop over j effectively accomplishes this. Once the autocovariance of s_t is known it is straightforward to get the autocovariance of the HP filtered stacked vector since it is just a linear function of s_t.

```
gammf=kron(zeros(1,ncorr+1),zeros(ns+nex,ns+nex)) ;
facv=zeros(ny,(ncorr+1)*ny) ;

for k=0:ncorr
  for j=0:base
    if j==0
      gammj=gamm0 ;
      gammf(:,k*(ns+nex)+1:(k+1)*(ns+nex))=
         gammf(:,k*(ns+nex)+1:(k+1)*(ns+nex))+
         gammj*(hpa(k+1:2*maxf+1,1)'*hpa(1:2*maxf+1-k,1)) ;
    else
      gammj=m*gammj ;
      if j<=k
        gammf(:,k*(ns+nex)+1:(k+1)*(ns+nex))=
           gammf(:,k*(ns+nex)+1:(k+1)*(ns+nex))+
            gammj*(hpa(k+1-j:2*maxf+1,1)'*hpa(1:2*maxf+1-k+j,1))+
            gammj'*(hpa(k+j+1:2*maxf+1,1)'*hpa(1:2*maxf+1-k-j,1)) ;
       else
         gammf(:,k*(ns+nex)+1:(k+1)*(ns+nex))=
            gammf(:,k*(ns+nex)+1:(k+1)*(ns+nex))+
            gammj*(hpa(1:2*maxf+1+k-j,1)'*hpa(1-k+j:2*maxf+1,1))+
            gammj'*(hpa(k+j+1:2*maxf+1,1)'*hpa(1:2*maxf+1-k-j,1)) ;
       end
     end
   end
   facv(:,k*ny+1:(k+1)*ny)=
      hbig*gammf(:,k*(ns+nex)+1:(k+1)*(ns+nex))*(hbig') ;
end
```

Finally the procedure gets the standard deviations of the HP filtered data and uses these to define the autocorrelations which are stored in a matrix with the same shape as the autocovariance matrix. The results for both are returned to the calling m-file.

```
sd=sqrt(diag(facv(1:ny,1:ny))) ;
tcorr=kron(sd,sd') ;
facr=facv./kron(ones(1,ncorr+1),tcorr) ;
```

Next we have the m-file `gmmerr.m`. This file defines the GMM errors. The name of this file cannot be changed by the user. Formally, this procedure defines the $T \times m$ matrix whose tth row is the GMM error term, $f(x_t, \psi)'$. The user can think of this procedure as defining the moment restrictions used to estimate the parameters. With the first 8 parameters ordered as θ, $\ln(A)$, ρ, σ, A_y, $\ln(\gamma_X)$, δ and α, it should be clear that the vectors `u1` through `u8` define the GMM error terms for the eight Euler equations listed in Section 9. The moment restrictions used to estimate the data moments are as follows

$$
\begin{aligned}
E\left[y_{hp,t}^2 - \sigma_y^2\right] &= 0 \\
E\left[c_{hp,t}^2 - \left(\frac{\sigma_c}{\sigma_y}\right)^2 y_{hp,t}^2\right] &= 0 \\
E\left[i_{hp,t}^2 - \left(\frac{\sigma_i}{\sigma_y}\right)^2 y_{hp,t}^2\right] &= 0 \\
E\left[n_{hp,t}^2 - \left(\frac{\sigma_n}{\sigma_y}\right)^2 y_{hp,t}^2\right] &= 0 \\
E\left[n_{hp,t}^2 - \left(\frac{\sigma_n}{\sigma_{y-n}}\right)^2 (y-n)_{hp,t}^2\right] &= 0.
\end{aligned}
$$

The *hp* subscript simply denotes the HP cyclical component of the time series in question. The data are recovered from the global variable `xdata`. As we will see below, its columns are, in order, the levels of consumption (1), investment (2), output (3), hours (4) and capital (5), followed by HP filtered consumption (6), investment (7), output (8), hours (9) and average productivity (10).

```
function u=gmmerr(b)
global xdata ;
dep=1-(xdata(2:115,5)-xdata(1:114,2))./xdata(1:114,5) ;
t=(1:1:113)' ;
t0=(0:1:114)' ;
beta_=1.03^(-.25) ;

loga=log(xdata(1:115,3))-log(xdata(1:115,5))*(1-b(8,1))-
   log(xdata(1:115,4))*b(8,1)-t0*b(6,1)*b(8,1) ;
u1=ones(113,1)*b(1,1)-b(8,1)*xdata(2:114,3)./(xdata(2:114,1).*xdata(2:114,4)) ;
u2=loga(2:114,1)-b(2,1)-loga(1:113,1)*b(3,1) ;
u3=u2.*loga(1:113,1) ;
u4=u2.^2-b(4,1)^2 ;
u5=log(xdata(2:114,3))-b(5,1)-t*b(6,1) ;
u6=u5.*t/113 ;
u7=dep(2:114,1)-b(7,1) ;
u8=1-beta_*(xdata(2:114,1)./xdata(3:115,1)).*((1-b(8,1))*xdata(3:115,3)./
   xdata(3:115,5)+1-b(7,1)) ;
u9=xdata(2:114,8).^2-b(9,1)^2 ;
u10=xdata(2:114,6).^2-b(10,1)^2*xdata(2:114,8).^2 ;
u11=xdata(2:114,7).^2-b(11,1)^2*xdata(2:114,8).^2 ;
u12=xdata(2:114,9).^2-b(12,1)^2*xdata(2:114,8).^2 ;
u13=xdata(2:114,9).^2-b(13,1)^2*xdata(2:114,10).^2 ;
u=[ u1 u2 u3 u4 u5 u6 u7 u8 u9 u10 u11 u12 u13 ] ;
```

The file `cdfchic.m` is used to compute the right-hand tail of a χ^2 distribution. It's first argument is the statistic for which the tail area is to be evaluated. It's second argument is the number of degrees of freedom.

```
function y=cdfchic(x,d)
y=1-gammainc(x/2,d/2) ;
```

The file `dfeval.m` computes numerical gradients of vector-valued functions with vector arguments. It's first argument, `f`, is the name of the m-file that defines the function to be differentiated. It's second argument, `f0`, is the value of the function at the point of differentiation, `x0`. The fourth argument, `dx`, is a steplength to use in computing the derivatives. If this is set to 0, a default steplength will be used.

```
function df=dfeval(f,f0,x0,dx)

[n,nc]=size(f0) ;
[k,nc]=size(x0) ;
```

```
df=zeros(n,k) ;

if dx==0
   ax=abs(x0) ;

   dx=(max([ ax ones(k,1)*(1e-2) ]')')*(1e-5) ;
end

dxm=eye(k).*(dx*ones(1,k)) ;

for j=1:k
   dxj=dxm(:,j) ;
   df(:,j)=feval(f,x0+dxj)-feval(f,x0-dxj) ;
end

df=df./(ones(n,1)*(dx')*2) ;
```

The next m-file `mgmmerr.m`, computes the means of the GMM errors.

```
function m=mgmmerr(b0)
m=mean(gmmerr(b0))' ;
```

The m-file `s0calc.m` computes $\hat{S}_T$ as part of the GMM procedure. The parameters `wt` and `wl` control the computation of $\hat{S}_T$. When `wt`=0, the user is imposing the assumption that the GMM errors are orthogonal to lagged information. For each of the other methods the user is computing

$$\begin{aligned}\hat{S}_T &= \sum_{t=1}^{T}\left[f(x_t,\hat{\psi}_T)f(x_t,\hat{\psi}_T)'\right]+\sum_{j=1}^{N}L(j,N)\sum_{t=1+j}^{T}\left[f(x_t,\hat{\psi}_T)f(x_{t-j},\hat{\psi}_T)'\right]\\ &\quad+\sum_{j=1}^{N}L(j,N)\sum_{t=1}^{T-j}\left[f(x_t,\hat{\psi}_T)f(x_{t+j},\hat{\psi}_T)'\right]\end{aligned}$$

where `wl`$=N$ and $L(j,N)$ is a particular type of lag window. When `wt`=1, $L(j,N)=1$, and the user is effectively imposing that the GMM errors have an MA(N) structure. This approach is not guaranteed to lead to a positive semi-definite $\hat{S}_T$. The other methods are appropriate for cases where the GMM errors may have an infinite order and are described by Newey and West (1987). The lag length, N, should be increased with the sample size, T, for consistency, but in finite samples there are no hard and fast rules about how to choose N. For `wt`=2, $L(j,N)=1-j/(N+1)$, while for `wt`=3,

$$L(j,N)=\begin{cases}1-6\left(\frac{j}{N+1}\right)^2+6\left(\frac{j}{N+1}\right)^3 & \text{for } j<(N+1)/2\\ 2\left(1-\frac{j}{N+1}\right)^3 & \text{otherwise.}\end{cases}$$

For the type of estimation described in Section 9, the GMM errors will have arbitrary serial correlation so that one of the latter methods should be used.

```
function s0=s0calc(u,wt,wl)

% wt indicates the window type
%   0: no lags used in computing s0
%   1: flat window used-may not be psd
%   2: Bartlett window used
%   3: Parzen window used
% wl indicates the lag length for the window (largest non-zero lag)

[T,k]=size(u) ;
s0=(u'*u)/T ;
if wt~=0
   if wt==1
      for j=1:wl
         s0j=(u(1+j:T,:)'*u(1:T-j,:))/T ;
         s0=s0+s0j+(s0j') ;
      end
   elseif wt==2
      for j=1:wl
         s0j=(u(1+j:T,:)'*u(1:T-j,:))/T ;
         s0=s0+(s0j+(s0j'))*(1-j/(wl+1)) ;
      end
   else
      for j=1:wl
         s0j=(u(1+j:T,:)'*u(1:T-j,:))/T ;
         if j<((wl+1)/2)
            fac=1-6*(j/(wl+1))^2+6*(j/(wl+1))^3 ;
         else
            fac=2*((1-j/(wl+1))^3) ;
         end
         s0=s0+(s0j+(s0j'))*fac ;
      end
   end
end
```

The file `quadmin.m` is the m-file that minimizes the GMM objective function for a given weighting matrix, W_T. It mimics the GMM code written by Hansen, Heaton and Ogaki (1992) for GAUSS. It has several arguments. First, is `func`, which is the name of the m-file that defines the means of the GMM errors. Second is `x0`, which is the initial value of the parameter vector. Third is `xtol`, a convergence criterion defined on the scaled gradients of the GMM objective function. This has no default value so it must be set by the user. Similarly, `nitermax` is the maximum number of iterations over the parameter vector allowed, and it has no default value. Finally, on each iteration over the parameters, the minimizer tries to take a step in a direction suggested by the gradients of the objective function. If a full step does not go downhill, then the step length is cut in half. This process continues until a lower point is found. The parameter `sstol` puts a limit on how small a step is permitted. The output of the function is the minimizing point, the value of the GMM objective at that point, and the number of iterations it took to obtain the minimum.

The first part of the code defines the weighting matrix, W_T, as a global variable. It finds the size of the initial parameter vector, `x0`. It then evaluates the means of the GMM errors, at `x0`, denoted `g0`. It also computes the derivatives of these means, denoted `dg0`, with respect to the parameter vector.

```
function [x1,f1,niter]=quadmin(func,x0,xtol,nitermax,sstol) ;
global wmatrix_ ;

[k,nc]=size(x0) ;

g0=feval(func,x0) ;
dg0=dfeval(func,g0,x0,0) ;
```

The next part of the code computes the initial value of the GMM objective function, `f0`, and its derivatives with respect to the parameters, `df0`. An initial outer-product method approximation to the inverse Hessian is then computed: `h0`. The initial vector of directions in which to try to move the parameter vector is given by `xi`, and is `df0` scaled by `h0`. The parameters which keep track of whether the convergence criterion has been satisified or not and the number of iterations are also initialized.

```
f0=g0'*wmatrix_*g0 ;
df0=g0'*wmatrix_*dg0 ;

h0=inv(dg0'*wmatrix_*dg0) ;

xi=-h0*(df0') ;
convcrit=0 ;
niter=1 ;
```

The next part of the code is executed if the convergence criterion has not yet been satisified. This is true as long as `convcrit`=0. The next part of the code initializes `s`, the size of the step to take in the directions suggested by `xi`. When `s`=1 a full step is taken. As long as `s` does not become too small, smaller and smaller steps are taken until a lower value of the GMM objective is obtained. Then `test_` is computed to see if the scaled gradients satisfy their convergence criterion, `xtol`. If they do, or if the maximum number of iterations has been exceeded, the process stops. If not, the function values, derivatives, approximate inverse Hessian and suggested directions for minimization are recomputed, and the process is repeated.

```
while convcrit==0

  s=1 ;
  while s>sstol
```

```
    dx=s*xi ;
    x1=x0+dx ;
    g1=feval(func,x1) ;
    f1=g1'*wmatrix_*g1 ;
    if f1<f0
      g0=g1 ;
      f0=f1 ;
      x0=x1 ;
      s=0 ;
    end

    s=s*0.5 ;
  end
  test_=max( abs(xi)./( max([ abs(x0) ones(k,1) ]')' ) ) ;
  if niter==nitermax
    convcrit=1 ;
  elseif test_<xtol
    convcrit=1 ;
  else
    dg0=dfeval(func,g0,x0,0) ;
    df0=g0'*wmatrix_*dg0 ;
    h0=inv(dg0'*wmatrix_*dg0) ;
    xi=-h0*(df0') ;
  end
  niter=niter+1 ;
end
```

The file gmmest.m controls the GMM estimation. It has as arguments, b0, the initial parameter vector, nstep, the number of steps over the weighting matrix to take, nmax, the maximum number of iterations on any one step over the weighting matrix, xtol, the convergence criterion on the gradients on each step over the weighting matrix, w0flag, the type of initial weighting matrix, w0, the initial weighting matrix if the defaults are not desired, wtype and wlgth, the parameters wt and wl of the s0calc procedure, and sstol, the tolerance on step length in the hill climbing algorithm (see quadmin.m). If w0flag=0 an identity matrix is used for the initial weighting matrix. If w0flag=1, the weighting matrix is initialized to equal the inverse of $\hat{S}_T$ evaluated at the starting parameter vector. Otherwise, the user must pass a full weighting matrix w0 to gmmest.m. Notice that the weighting matrix is always defined as a global variable. The output of gmmest.m is a vector of parameters estimates, b, a variance covariance matrix, v, and a value of the GMM objective function, q.

```
function [b,v,q]=gmmest(b0,nstep,nmax,xtol,w0flag,w0,wtype,wlgth,sstol)

global wmatrix_ ;

[nb,nb1]=size(b0) ;
```

```
u0=gmmerr(b0) ;
[T,k]=size(u0) ;

if w0flag==0
   wmatrix_=eye(k) ;
elseif w0flag==1
   wmatrix_=inv(s0calc(u,wtype,wlgth)) ;
else
   wmatrix_=w0 ;
end

% 1st Step of GMM

[b,q,niter]=quadmin('mgmmerr',b0,xtol,nmax,sstol) ;

if niter>nmax
   'warning: maximum iterations reached'
end

u=gmmerr(b) ;
g0=mean(u)' ;
dg=dfeval('mgmmerr',g0,b,0) ;
idwd=inv(dg'*wmatrix_*dg) ;
s0=s0calc(u,wtype,wlgth) ;

'Iteration & Number of Steps'
[ 1 niter]

'Parameters'

[ b ]

iter=1 ;

while iter<nstep

 wmatrix_=inv(s0) ;
 b0=b ;

 [b,q,niter]=quadmin('mgmmerr',b0,xtol,nmax,sstol) ;

 if niter>nmax
  'warning: maximum iterations reached'
 end
 g0=feval('mgmmerr',b) ;
 dg=dfeval('mgmmerr',g0,b,0) ;
 v=(inv(dg'*wmatrix_*dg))/T ;
 sd=sqrt(diag(v)) ;

 iter=iter+1 ;

 'Iteration & Number of Steps'
 [iter niter]
 'GMM T*Q'
 q=T*q ;
 if k>nb
    [ q cdfchic(q,k-nb) ]
 else
    q
 end
 'Parameters & Std. Errors'
 [ b sd ]

end
```

Working our way in, the next m-file `indmom.m` takes the parameter vector as an argument. It then calls the file `indsolve.m` to determine the decision rules. Then, given the decision rules, it calls `tshpmom.m` to calculate the autocovariances and autocorrelations of HP filtered data implied by the model. The moments we are interested in are

$$\sigma_y \quad \frac{\sigma_c}{\sigma_y} \quad \frac{\sigma_i}{\sigma_y} \quad \frac{\sigma_n}{\sigma_y} \quad \frac{\sigma_n}{\sigma_{y-n}} \quad \rho_{n,y-n}$$

Since these are all based on contemporaneous relationships the parameter `ncorr` is set to 0. These moments are then returned to the calling program.

```
function mm=indmom(b)

beta_=1.03^(-.25) ;
sigma=b(4,1) ; sigma=sigma*sigma ;

drules=indsolve(beta_,b(8,1),b(6,1),b(7,1),b(3,1)) ;

[facv,facr]=tshpmom(sigma,drules,0) ;

sd=sqrt(diag(facv)) ;

mm=[ sd(6,1) ; sd(4,1)/sd(6,1) ; sd(8,1)/sd(6,1) ; sd(5,1)/sd(6,1) ;
      sd(5,1)/sd(7,1) ; facr(5,7) ] ;
```

The next m-file, `inddata.m`, takes the parameter vector as an argument and pulls off the elements of the parameter vector which represent the corresponding 'data' moments. Notice that the last moment $\rho_{n,y-n}$ is defined in terms of the others. If this parameter were estimated separately the matrix S_0, and the matrix $\hat{S}_T$, would be singular.

```
function dm=inddata(b)

dm=[ b(9:13,1) ;
    (b(9,1)^2-b(12,1)^2*b(9,1)^2-b(13,1)^(-2)*b(12,1)^2*b(9,1)^2)/
       (2*b(13,1)^(-1)*b(12,1)^2*b(9,1)^2) ] ;
```

Moving in again we have `indtests.m`. It uses the estimated parameter vector, `b`, and its variance-covariance matrix, `varb`, to compute the test statistics and their associated p-values. The methods outlined in Section 7 are used to compute the model moments (generated by `indmom.m`), the data moments (generated by `inddata.m`), and their standard errors. Then it calculates the test statistic $\mathcal{W}$ and p-value associated with the difference between *each* 'model' moment and the corresponding 'data' moment. These results are then printed in tabular form.

```
function tsts=indtests(b,varb)

mm=indmom(b) ;
[nm,nm2]=size(mm) ;
gm=dfeval('indmom',mm,b,0) ;

dm=inddata(b) ;
gd=dfeval('inddata',dm,b,0) ;

vm=zeros(nm,1) ;
vd=zeros(nm,1) ;

test=zeros(nm,1) ;
pv=zeros(nm,1) ;

for k=1:nm
  vm(k,1)=gm(k,:)*varb*(gm(k,:)') ;
  vd(k,1)=gd(k,:)*varb*(gd(k,:)') ;
  testn=mm(k,1)-dm(k,1) ;
  testd=(gm(k,:)-gd(k,:))*varb*((gm(k,:)-gd(k,:))') ;
  test(k,1)=(testn^2)/testd ;
  pv(k,1)=cdfchic(test(k,1),1) ;
end

sqvm=sqrt(vm) ;
sqvd=sqrt(vd) ;

'Model Moments   s.e.      Data Moments     s.e       Test       P-Value'
```

```
tsts=[mm sqvm dm sqvd test pv] ;
tsts
```

Next we have `hpfilter.m`, which is located in the main directory. It is designed to HP filter data. It takes as arguments a matrix of data of dimension $d \times k$, where d is the sample size and k is the number of variables, and a parameter `lamb` which is the λ parameter of the filter. The user can leave this procedure unchanged.

```
function y=hpfilter(x,lamb)
[d,k]=size(x) ;
a=zeros(d,d) ;
for i=3:d-2
 a(i,i)=6*lamb+1;
 a(i,i+1)=-4*lamb;
 a(i,i+2)=lamb;
 a(i,i-2)=lamb;
 a(i,i-1)=-4*lamb;
end
a(2,2)=1+5*lamb;
a(2,3)=-4*lamb;
a(2,4)=lamb;
a(2,1)=-2*lamb;
a(1,1)=1+lamb;
a(1,2)=-2*lamb;
a(1,3)=lamb ;
a(d-1,d-1)=5*lamb+1;
a(d-1,d)=-2*lamb;
a(d-1,d-2)=-4*lamb;
a(d-1,d-3)=lamb;
a(d,d)=1+lamb;
a(d,d-1)=-2*lamb;
a(d,d-2)=lamb;
y=(eye(d)-inv(a))*x ;
```

The last m-file called by `indivis.m` is `indimp.m`. It computes the impulse response functions of technology, consumption, hours, output, average productivity and investment in response to technology innovations up to `nimp` periods in the past. It takes the parameter vector and `nimp` as arguments. A large part of the code is concerned with beautifying the graphics output. Computing the decision rules is done by calling `indsolve.m`. The impulse responses are generated recursively as described in Section 6.

```
function irf=indimp(b,nimp)

beta_=1.03^(-.25) ;
```

```
sigma=b(4,1) ; sigma=sigma*sigma ;

drules=indsolve(beta_,b(8,1),b(6,1),b(7,1),b(3,1)) ;
[nex,nex2]=size(sigma) ;
[ny,ns]=size(drules) ;
ns=ns-nex ;
m=drules(1:nex+ns,:) ;
h=drules(nex+ns+1:ny,:) ;
irf=zeros(nimp,9) ;

for k=1:nimp
  if k==1
    irfs=eye(ns+nex) ;
  else
    irfs=m*irfs ;
  end
  irff=h*irfs ;
  irf(k,1)=irfs(ns+1,ns+1) ;
  irf(k,2)=irfs(1,ns+1) ;
  irf(k,3)=irff(2,ns+1) ;
  irf(k,4)=irff(3,ns+1) ;
  irf(k,5)=irff(4,ns+1) ;
  irf(k,6)=irff(5,ns+1) ;
  irf(k,7)=irff(6,ns+1) ;
  irf(k,8)=irff(7,ns+1) ;
  irf(k,9)=irff(8,ns+1) ;
end

t=(1:1:nimp)' ;
zers=zeros(nimp,1) ;

figure(1)

subplot(221)
plot(t,[ irf(:,1) zers] )
title('A')
set(gca,'XLim',[0 nimp+1]) ;

subplot(222)
plot(t,[ irf(:,4) irf(:,6) zers ])
title('H and Y/N')
set(gca,'XLim',[0 nimp+1]) ;

subplot(223)
plot(t,[ irf(:,3) irf(:,7) zers ])
```

```
title(’C and I’)
set(gca,’XLim’,[0 nimp+1]) ;

subplot(224)
plot(t,[ irf(:,5) irf(:,2) zers ])
title(’Y and K’)
set(gca,’XLim’,[0 nimp+1]) ;
```

The main program that controls estimation and testing is `indivis.m`. At the beginning are several sections the user should change depending on the model being estimated. First, the starting values of the parameter vector are defined in `bstart`. The parameters being estimated in our indivisible labor example (see above) are

$$\left(\theta, \ln A, \rho_A, \sigma_\epsilon, A_y, \ln\gamma, \delta, \alpha, \sigma_y, \frac{\sigma_c}{\sigma_y}, \frac{\sigma_i}{\sigma_y}, \frac{\sigma_n}{\sigma_y}, \frac{\sigma_n}{\sigma_{y-n}}\right)$$

```
’Indivisible Labor Model Deterministic Trend’
’Parameters: Theta,lnA,rhoA,sigA,Ay,lnGamm,Delta,Alpha,5 Data Mom’

% SET INITIAL PARAMETER VALUES

bstart=[ 1.0 1.0 0.9 0.01 1.0 0.01 0.02 0.6 0.01 1.0 1.0 1.0 1.0 ]’ ;
```

Next comes a section in which various parameters controlling the GMM estimation procedure are defined. `w0` and `w0flag` determine the weighting matrix on the first GMM iteration. `nstep` determines the number of steps over the weighting matrix that will be taken before the estimation procedure is terminated. The parameters `miter` and `xtol` control the maximum number of iterations on each GMM step, and the convergence tolerance, respectively.

The parameters `s0method` and `lags` define the window type and lag length to be used in computing $\hat{S}_T$. (See the description of s0calc.m, above.)

Finally `sstol` determines a step length tolerance used in the estimation procedure. See the file `quadmin.m` for details.

```
% SET GMM PARAMETERS

w0flag=0 ;

% w0flag=0 : W0=I is used as the initial weighing matrix
% w0flag=1 : bstart is used to calculate initial W0
% w0flag=2 : W0 passed to gmmest is used as initial W0.

w0=0 ; % initial W0, set to 0 if w0flag=0 or 1

nstep=2 ;
```

```
% Sets # of steps over weighting matrix, W0.
% Set w0flag=0 and nstep=2 to execute usual 2-Stage GMM.

miter=400 ;

% Sets maximum number of iterations on each GMM step

xtol=1e-6 ;

% gradient tolerance for convergence on each step

s0method=2;

% This variable is used to choose the method to calculate S0
% s0method=0  No lags used in computing s0
% s0method=1  Flat window used when computing s0 (may not be psd)
% s0method=2  Bartlett window used when computing s0
% s0method=3  Parzen window used when computing s0

lags=5 ;  % number of lags used in window methods for calculating s0

sstol=1e-12 ;

% step length tolerance - see quadmin
```

The next step involves reading in the data which are contained in the file HOARDING.DAT. These data are the same as the data from Christiano and Eichenbaum (1992). They represent quarterly observations from 1955Q3–1984Q1. The columns represent the per-capita levels of C, I, G, Y, H^h, K and H^e respectively. H^h and H^e are the *household* and *establishment* hours series discussed by Christiano and Eichenbaum (1992). The data are HP filtered using the m-file `hpfilter.m`. Finally, the data are stored in a global variable `xdata`.

```
% LOAD THE DATA

load hoarding.dat ;

c=hoarding(:,1) ;
dk=hoarding(:,2) ;
y=hoarding(:,4) ;
n=hoarding(:,5)/1369 ;
k=hoarding(:,6) ;
hc=hpfilter(log(c),1600) ;
hi=hpfilter(log(dk),1600) ;
hy=hpfilter(log(y),1600) ;
```

```
hn=hpfilter(log(n),1600) ;
hapl=hy-hn ;

global xdata ;

xdata=[ c dk y n k hc hi hy hn hapl ] ;
```

The final part of the program calls the m-file `gmmest.m` to perform the estimation. The variables `b`, `v`, and `q` represent the parameter estimates, their variance-covariance matrix and the Q statistic for testing over-identifying restrictions, if any. The routine `indtests.m` performs statistical tests regarding the data and model moments. Finally, `indimp.m` plots impulse response functions from the model out to 47 periods after a technology shock.

```
% ESTIMATE THE MODEL, PERFORM TESTS, GRAPH IMPULSE RESPONSES
[b,v,q]=gmmest(bstart,nstep,miter,xtol,w0flag,w0,s0method,lags,sstol) ;
dmmy=indtests(b,v) ;
dmmy=indimp(b,47) ;
```

This completes the description of the programs. The comfortable user should examine the files in the directories `divisib`, `ce1992` and `ber1993`, to see which minor changes are necessary to the above code in order to estimate different models. The file `dshpmom.m` replaces `tshpmom.m` for the programs in `ce1992`, because the model is one in which technology is difference stationary rather than trend stationary.

References

Burnside, C., M. Eichenbaum and S. Rebelo (1993) “Labor Hoarding and the Business Cycle”, *Journal of Political Economy*, 101, pp.

Christiano, L. (1998) “Solving Dynamic Equilibrium Models by a Method of Undetermined Coefficients,” manuscript, Northwestern University.

Christiano, L. and M. Eichenbaum (1992) “Current Real Business Cycle Theories and Aggregate Labor-Market Fluctuations,” *American Economic Review*, 82, pp 430–450.

Hansen, G. (1985) “Indivisible Labor and the Business Cycle,” *Journal of Monetary Economics*, 16, pp 309–328.

Hansen, L.P. (1982) “Large Sample Properties of Generalized Method of Moments Estimators,” *Econometrica*, 50, pp 1029–1054.

Hansen, L.P., J. Heaton and M. Ogaki (1992) GMM Software for GAUSS.

Hodrick, R.J. and E.C. Prescott (1997) “Postwar U.S. Business Cycles: An Empirical Investigation,” *Journal of Money, Credit and Banking*, 29, pp 1–16.

Judd, K.L. (1998) *Numerical Methods in Economics*. Cambridge, MA: MIT Press.

King, R., C. Plosser and S. Rebelo (1988a) “Production, Growth and Business Cycles: I & II,” *Journal of Monetary Economics*, 21, pp 195–232, 309–341.

King, R., C. Plosser and S. Rebelo (1988b) “Production, Growth and Business Cycles: Technical Appendix,” manuscript.

King, R. and S. Rebelo (1989) “Low Frequency Filtering and Real Business Cycles,” Rochester Center for Economic Research Working Paper No. 205.

King, R.G. and M.W. Watson (1997) “System Reduction and Solution Algorithms for Singular Linear Difference Systems Under Rational Expectations,” manuscript, University of Virginia.

Marimon, R. and A. Scott, eds. (1999) *Computational Methods for the Study of Dynamic Economies*. Oxford: Oxford University Press.

Newey, W.K. and K.D. West (1987) “A Simple, Positive Semi-definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix,” *Econometrica*, 55, pp 703–708.

Rogerson, R. (1988) “Indivisible Labor, Lotteries and Equilibrium,” *Journal of Monetary Economics*, 21, pp 3–16.

Sargent, T. (1987) *Macroeconomic Theory*. New York: Academic Press.

Uhlig, H. (1997) “A Toolkit for Analyzing Nonlinear Dynamic Stochastic Models Easily,” manuscript, CentEr, University of Tilburg.

Figure 1

Impulse Response Functions from the Divisible Labor Model

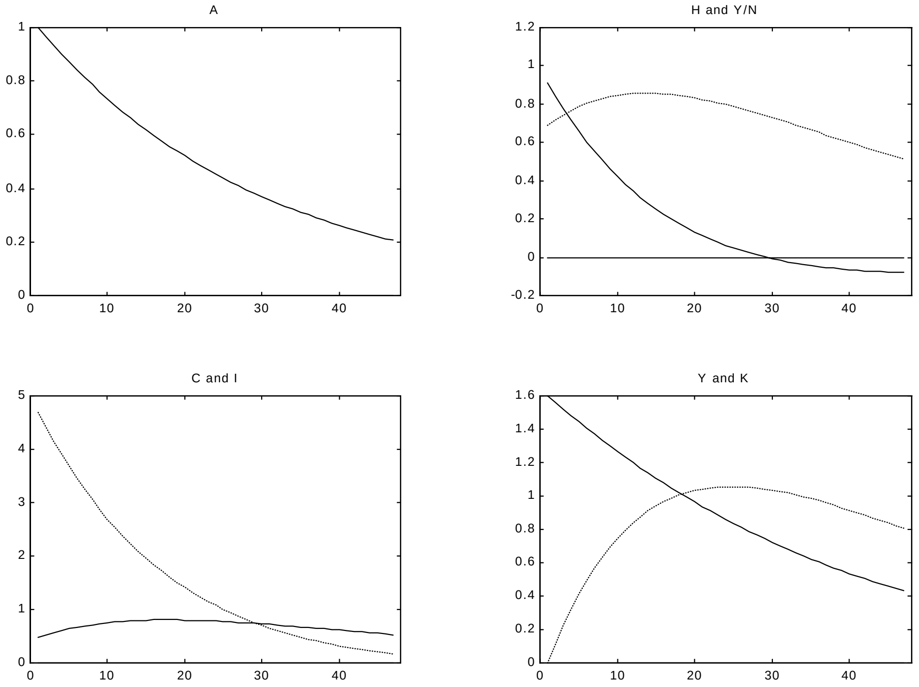


Figure 2

Impulse Response Functions from the Indivisible Labor Model

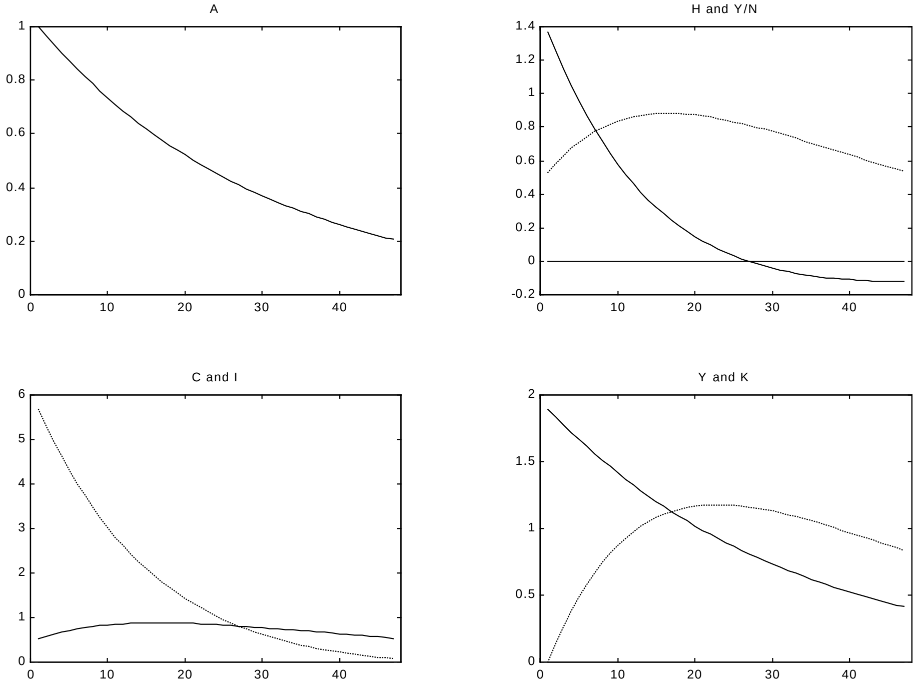


Figure 3

Spectra of Four Time Series

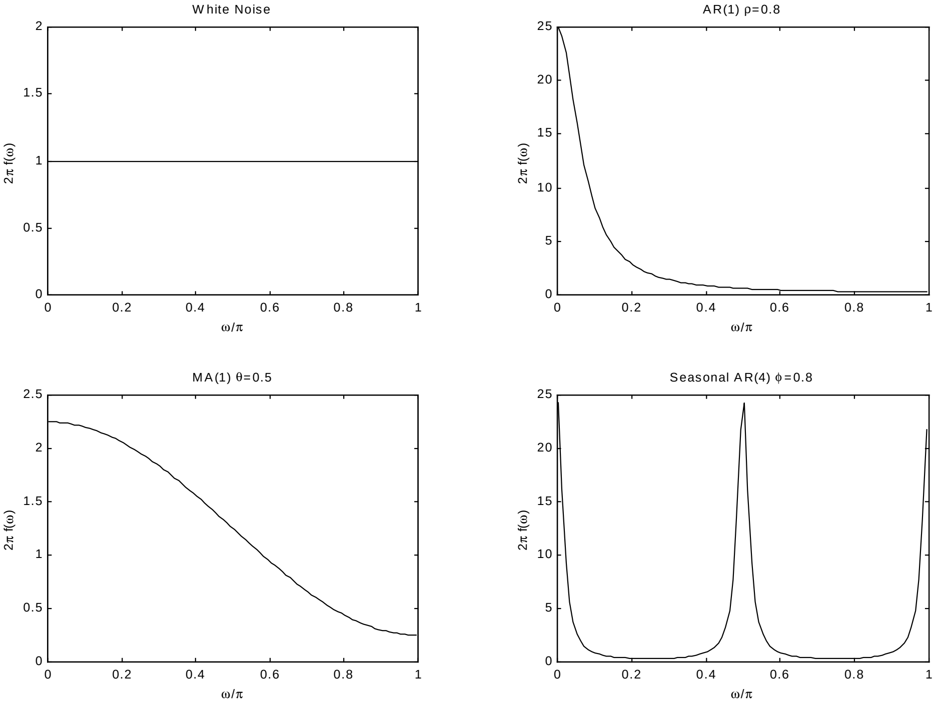
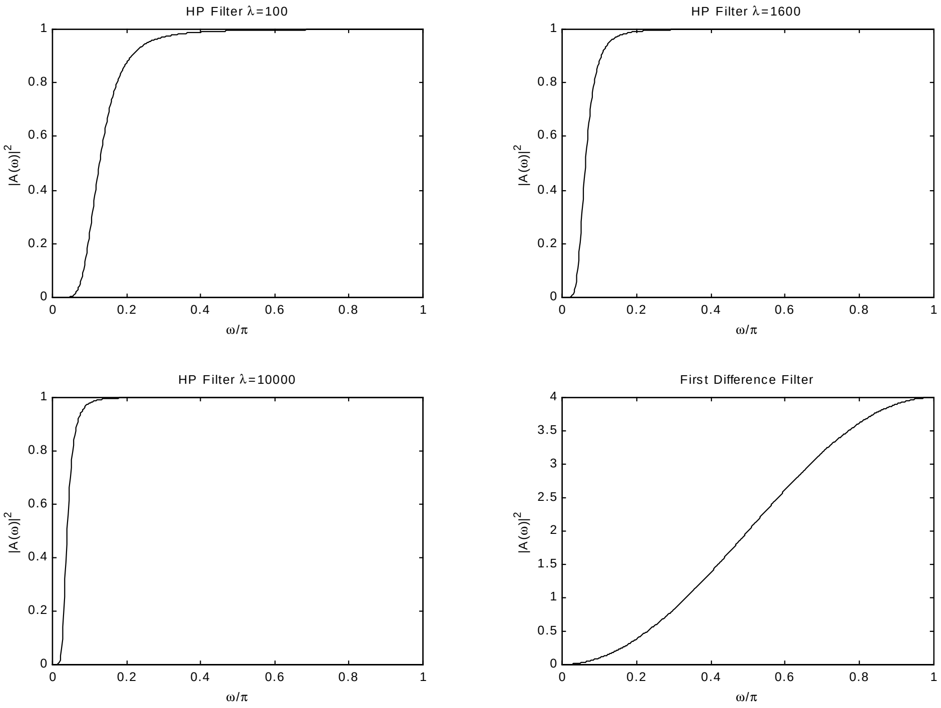


Figure 4

The Transfer Function of the HP Filter


HP Filter λ=100
HP Filter λ=1600
HP Filter λ=10000
First Difference Filter
$|A(\omega)|^2$
0
0.2
0.4
0.6
0.8
1
1.5
2
2.5
3
3.5
4
0.5
ω/π