Chapter 2
Recursive Methods for Computing Equilibria of Business Cycle Models

Gary D. Hansen and Edward C. Prescott

1. Introduction

In this chapter, we describe some computational methods for computing equilibria of business cycle models. The class of economies for which these methods can be applied is surprisingly large, as we illustrate with a number of examples. Generally, this class consists of economies that fluctuate around a steady-state or balanced-growth path and display local dynamics that are well approximated by a linear law of motion. The methods we describe are designed to economize on the time spent learning to use the techniques and to modify them for a particular application. They are not designed to minimize computing costs, which, in any case, are only a minute or two of personal computer time for most business cycle applications.

The fact that it is not generally possible to compute equilibria of business cycle models analytically led Kydland and Prescott (1982), and subsequently others, to consider a structure for which this is possible. Such a structure is one with a quadratic objective, linear constraints, and exogenous disturbances generated by a first-order, linear, vector-autoregressive process. The particular quadratic objective chosen is the second-order Taylor series expansion of the return function for the deterministic version of the model evaluated at its steady state.

An additional advantage of a linear-quadratic structure is that equilibria can be easily computed even when the dimension of the state variable is large. It does, however, have the added consequence that the resulting equilibrium law of motion is linear. This does not appear to be a serious limitation given that there is little evidence of major nonlinearities in aggregate data. In situations in which the behavior being modeled displays important nonlinearities, methods other than those reviewed here are needed. Some of these will be described in subsequent chapters.

We divide the class of applications where these methods have been used into two subclasses. In the first, competitive equilibria are Pareto optimal and hence solve a social planning problem. In Section 2, we examine the stochastic growth model augmented to have a labor-leisure allocation decision and show how it can be mapped into a basic recursive social planning problem.[1] We also consider a number of extensions of this basic environment to illustrate the flexibility of this mapping. These extensions include time to build, indivisible labor, and geometric growth. In Section 3, we describe computational algorithms for specifying the linear-quadratic social planning problem and for computing the soluton to the problem.

The second subclass consists of economies for which distortions due to taxes or externalities typically make it necessary to solve for an equilibrium directly. In Section 4, we deal with homogeneous-agent recursive economies in which the competitive equilibrium need not be Pareto optimal. The first application introduces taxes into our basic business cycle model. The second introduces a cash-in-advance constraint for the purchase of a subset of the consumption goods. In Section 5, we describe algorithms similar to those described in Section 3 for choosing a linear-quadratic recursive economy and for computing the equilibrium stochastic process for that economy. Finally, in Section 6, we demonstrate how these recursive computational methods can be extended to the study of heterogeneous-agent recursive economies. Examples applying these methods will be discussed in Chapter 4.

2. Social Planning Problems

In this section we describe some examples of economies for which competitive equilibrium allocations are identical to the allocations chosen by a social planner that acts to maximize the welfare of a representative agent. For these economies, the Second Welfare Theorem applies. In such situations the equilibrium allocations can be determined by solving a well-behaved concave optimization problem.[2] In addition, the marginal rates of substitution and transformation, evaluated at the optimal allocation, can be used to find equilibrium relative prices.

For each of our examples, the social-planning problem involves solving a dynamic programming problem of the following form (primes denote next-period values):

$$v(z, s) = \max\{r(z, s, d) + \beta E[v(z', s')|z, s]\} \tag{1}$$

$$\text{s.t.} \quad z' = A(z) + \epsilon' \tag{2}$$

$$s' = B(z, s, d). \tag{3}$$

The elements of this program are as follows: z is a vector of exogenous state variables; ϵ is a vector of random variables distributed independently over time with mean zero and finite variance (some components of ϵ may have zero variance); s is a vector of endogenous state variables; and d is a vector of decision variables.[3] Equation (2) is the law of motion for z, where A is a linear function.[4] The realization of z is observed at the beginning of the period. Equation (3) is the law of motion for the endogenous state variables, where B is also linear. Finally, $r(z, s, d)$ is the return function and $v(z, s)$ is the optimal value function for the problem.

An important feature of this problem is that A and B are linear functions, which means that any nonlinear constraints have been substituted into the return function. In Section 3, we describe a method for solving a linear-quadratic approximation of a problem of this form.

The equilibrium business cycle literature is full of applications where dynamic programming problems of this sort are formulated and solved.[5] The examples considered in this section, which are drawn from this literature, include a basic version of the stochastic growth model (the divisible labor model of Hansen [1985]) and a few variants of that model. In particular, we consider adding geometric growth, time to build, and indivisible labor to the basic model.

The Basic Model

This model is an extension of the Brock and Mirman (1972) optimal stochastic growth model upon which much of the equilibrium business cycle literature is based. A representative agent maximizes the utility fmction

$$E\sum_{t=0}^{\infty} \beta^t U(c_t, \ell_t), \quad 0 < \beta < 1, \tag{4}$$

where U is concave, strictly increasing, and twice continuously differentiable in both arguments. The variables c_t and ℓ_t are consumption and leisure in period t, respectively. The household is endowed with one unit of time, which is divided between work, h_t, and leisure, so that $h_t + \ell_t = 1$.

The representative agent has access to a technology that produces output, y_t, from capital, k_t, and labor:

$$y_t = z_t F(k_t, h_t). \tag{5}$$

The production function, F, is concave, twice continuously differentiable, increasing in both arguments, and displays constant returns to scale. The variable z_t is a technology shock, which is observed at the beginning of the period and follows a first-order linear Markov process:

$$z_{t+1} = A(z_t) + \epsilon_{t+1}, \tag{6}$$

where the ϵ's are i.i.d. random variables with mean zero and finite variance and A is a linear function.

Total output can be freely allocated to either investment, i_t, or consumption:

$$y_t = c_t + i_t, \tag{7}$$

where investment this period becomes productive capital next period. In particular, the capital stock evolves according to the law of motion:

$$k_{t+1} = (1-\delta)k_t + i_t, \quad 0 < \delta < 1. \tag{8}$$

The problem solved by the social planner is to maximize (4) subject to (5) through (8) with z_0 and k_0 given. We, however, need to express this problem as a dynamic programming problem that is a special case of (1) in order to apply the solution method described in the next section. This is accomplished by combining constraints (5) and (7) to eliminate y_t, solving the resulting equation for c_t, and substituting it into the utility function. This yields a version of problem (1) where $s = k$, $d = (h, i)$, the return function is

$$r(z, k, h, i) = U[zF(k, h) - i, 1 - h],$$

and the law of motion for s is, $B(z, k, h, i) = (1-\delta)k + i$.

We now consider three additional examples that are simply elaborations of this basic model.

Geometric Growth

As a second example, we add labor-augmenting technological growth to the basic model.[6] We do this by replacing (5) with

$$y_t = z_t F(k_t, \lambda^t h_t), \quad \text{where } \lambda > 1. \tag{9}$$

In addition, we require that the elasticity of substitution between consumption and leisure equal one so that hours worked is constant on the balanced growth path. An example of such a utility function is the following:

$$U(c_t, \ell_t) = (c_t^{\alpha}\ell_t^{1-\alpha})^{\rho}/\rho, \quad 0 < \alpha < 1, \rho < 1 \quad \text{and} \quad \rho \neq 0 \tag{10}$$

All other aspects of the model are the same as for the basic model. The equilibrium allocation is obtained by maximizing (4) subject to (9) and (6)–(8). A property of the solution to this problem is that consumption, investment, output, and capital all grow at the same rate: $\lambda - 1$. In order to solve this problem using our method, we transform the problem so that the solution is stationary over time. The following change of variables achieves this purpose:

$$\hat{y} = y_t/\lambda^t, \quad \hat{c}_t = c_t/\lambda^t, \quad \hat{i} = i_t/\lambda^t \quad \text{and} \quad \hat{k} = k_t/\lambda^t. \tag{11}$$

After this transformation, the social planner's problem is a special case of (1) if $s = \hat{k}$, $d = (h, \hat{i})$,

$$r(z, \hat{k}, h, \hat{i}) = ([zF(\hat{k}, h) - \hat{i}]^{\alpha}(1-h)^{1-\alpha})^{\rho}/\rho,$$

and

$$B(z, \hat{k}, h, \hat{i}) = [(1-\delta)/\lambda]\hat{k} + (1/\lambda)\hat{i}.$$

In addition, the discount factor after the transformation, call it $\hat{\beta}$, equals $\lambda^{\alpha\rho}\beta$. We assume that the parameters are such that $\hat{\beta}$ is less than one.

Time to Build

Kydland and Prescott (1982) studied a version of the basic model in which multiple periods are required to build productive capital. This requires that the state vector include stocks of capital goods j periods from completion, u_j, in addition to finished capital. Thus, if it takes J periods to build productive capital, the state and decision variables are $s = (k, u_1, \ldots, u_{J-1})$ and $d = (h, u_J)$.

The laws of motion for these state variables are the following:

$$k' = (1-\delta)k + u_1,$$

and

$$u'_j = u_{j+1}, \quad \text{for } j = 1, \ldots, J-1.$$

Letting ϕ_j, for $j = 1, \ldots, J$, be the fraction of resources allocated to the investment project in the jth stage from the last, total investment in the current period is $i = \sum_{j=1}^{J} \phi_j u_j$. With investment defined in this way, the return function for the version of problem (1) corresponding to this economy is given by

$$r(z, k, u_1, \ldots, u_{J-1}) = U[zF(k, h) - i, 1 - h].$$

Indivisible Labor

We now consider an example drawn from Hansen (1985), where indivisible labor, along with Rogerson's (1988) employment lotteries, are introduced into the basic model. For this example, instead of a single representative agent, there is a continuum of ex ante identical agents. All quantities must be interpreted as per capita values in this case. The technology is the same as in the basic model, but the utility function is of the form $u(c) + g(\ell)$, where u and g are increasing, concave, and twice continuously differentiable. Indivisible labor implies that ℓ can take on only two values, $(1 - \hat{h})$ and 1, corresponding to working full time or not at all. An additional difference between this and the basic model is that the competitive equilibrium involves agents' trading employment lotteries that specify a probability of working, rather than hours of work directly.

Letting n equal the probability of working $\hat{h}$ hours, the expected utility of a representative household is

$$\begin{aligned} &n[u(c) + g(1-\hat{h})] + (1-n)[u(c) + g(1)] \\ &= u(c) + ng(1-\hat{h}) + (1-n)g(1).\end{aligned}$$[7]

Since there is a continuum of households, the equilibrium value of n is also equal to the fraction of households that work. This implies that total hours worked, h, is given by $n\hat{h}$. The utility of the stand-in agent, which enters the objectve function of the social planner, is the following (ignoring a constant term):[8]

$$U(c, h) = u(c) + \phi h,$$

where

$$\phi = [g(1-\hat{h}) - g(1)]/\hat{h}.$$

Therefore, the version of problem (1) for this example is the same as for the basic model except that the return function is linear in hours worked: $r(z, k, h, i) = u(zF(k,h)-i)+\phi h$. Although individual households do not choose hours worked under the competitive equilibrium interpretation of this economy, the decision variables for the social planner are the same as for the basic model.

It is also possible to solve this problem if the utility function is not additively separable in consumption and leisure and takes the form of equation (10). In this case, the commodity traded is an employment lottery that specifies consumption compensation contingent on the employment status. Letting c_1 be consumption when working and c_2 be consumption when not working, the appropriate utility function is

$$nU(c_1, 1-\hat{h}) + (1-n)U(c_2, 1).$$

The resource constraint, using the fact that $h = n\hat{h}$, is

$$nc_1 + (1-n)c_2 + i = zF(k, n\hat{h}).$$

In this case, the planning problem is mapped into the notation employed in problem (1) by setting $s = k$ and $d = (n, c_2, i)$. The return function is

$$r(z, k, n, c_2, i) = nU[\frac{zF(k, n\hat{h}) - (1-n)c_2 - i}{n}, 1-\hat{h}] + (1-n)U(c_2, 1)$$

3. Solving a Social Planning Problem

In this section, we describe a method for solving problems of the form (1) when the return function is quadratic. However, the applications we have considered typically do not deliver quadratic return functions. Therefore, we describe a procedure for approximating a general return function by one that is quadratic. The advantage of solving a linear-quadratic planning problem is that it is possible to solve for an explicit linear policy function, $d_t = d(z_t, s_t)$, which when substituted into (3) yields a linear law of motion for the state variables, $s_{t+1} = g(z_t, s_t)$.

In this discussion, we employ the following convention to refer to the dimension of a particular vector: let $\eta(x)$ equal the dimension of a column vector x, and $\eta(x, y)$ equal the dimension of the stacked vector (x, y). This implies that the vector z of exogenous state variables is of dimension $\eta(z) \times 1$. The dimensions

of s (the endogenous state variables) and d (the decision variables) are defined analogously. In addition, $\eta(\epsilon)$ is equal to $\eta(z)$.

Another important convention we employ is to define the first component of z to be constant over time (equal to one, without loss of generality). This assumption will help to simplify accounting later.

Forming the Quadratic Approximation

The quadratic approximation of r corresponds to the first three terms of a Taylor series expansion of this function at the steady state values for (z, s, d), corresponding to the certainty version of problem (1), denoted $(\bar{z}, \bar{s}, \bar{d})$. The vector $\bar{z}$ is the solution to the equation $\bar{z} = A(\bar{z})$. Given $\bar{z}$, the following $\eta(s, d)$ equations are solved for the $\eta(s, d)$ unknowns, $\bar{s}$ and $\bar{d}$:

$$r_d(\bar{z}, \bar{s}, \bar{d}) + \beta r_s(\bar{z}, \bar{s}, \bar{d})[I - \beta B_s(\bar{z}, \bar{s}, \bar{d})]^{-1} B_d(\bar{z}, \bar{s}, \bar{d}) = 0;$$
$$\bar{s} = B(\bar{z}, \bar{s}, \bar{d}). \tag{12}$$

In (12), r_d is the vector of partial derivatives with respect to the elements of d and is of dimension $1 \times \eta(d)$. Similarly, r_s is of dimension $1 \times \eta(s)$. Since $B(z, s, d)$ is actually $\eta(s)$ linear equations, B_s and B_d are of dimension $\eta(s) \times \eta(s)$ and $\eta(s) \times \eta(d)$, respectively. In practice, the first equation in (12) can be made much simpler if one begins by substituting the laws of motion (3) into the return function, eliminating some elements of d. The idea is to rewrite the problem so that next-period state variables, s', are current period decision variables. In this case, all of the elements of B_s are zero.

Let y be the stacked vector (z, s, d) and a superscript T denote the transpose of a vector. The Taylor series expansion of $r(y)$ at the steady-state $\bar{y}$ is

$$\tilde{r}(y) = r(\bar{y}) + Dr(\bar{y})^T(y-\bar{y}) + (1/2)(y-\bar{y})^T D^2r(\bar{y})(y-\bar{y}), \tag{13}$$

where $Dr(\bar{y})$ is the $\eta(y) \times 1$ vector of first partial derivatives of r and $D^2r(\bar{y})$ is the $\eta(y) \times \eta(y)$ matrix of second partial derivatives of r, where $\eta(y) = \eta(z, s, d)$. Both are evaluated at the steady state. The first element of $Dr(\bar{y})$ and the elements in the first row and column of $D^2r(\bar{y})$ are zero, since the first component of y is a constant term and not a variable.

Rather than computing $Dr(\bar{y})$ and $D^2r(\bar{y})$ algebraically, we approximate the components of these matrices numerically. Let h^i be an $\eta(y) \times 1$ vector, all of the components of which are zero except for the ith component, h^i_i, which is set equal to a small positive number, $\tilde{h}$. The value of $\tilde{h}$ should be chosen to be as small as possible, subject to avoiding computer accuracy problems.[9] The following formulas are used to obtain numerical approximations of the components of $Dr(\bar{y})$ and $D^2r(\bar{y})$ (recall that the first component of y is constant over time):

$$D_i r(\bar{y}) = [r(\bar{y}+h^i) - r(\bar{y}-h^i)]/(2\tilde{h})$$
$$D^2_{ii} r(\bar{y}) = [r(\bar{y}+h^i) + r(\bar{y}-h^i) - 2r(\bar{y})]/(\tilde{h}^2)$$

and

$$D_{ij}^2 r(\bar{y}) = [r(\bar{y} + h^i + h^j) - r(\bar{y} + h^i - h^j) - r(\bar{y} - h^i + h^j) + r(\bar{y} - h^i - h^j)]/(4\tilde{h}^2)$$

for $i \neq j (i, j = 2, \dots, \eta(y))$.

Exploiting the fact that the first component of y is equal to one, we can rearrange equation (13) so that $\tilde{r}(y) = y^T Q y$, where Q is a symmetric matrix of dimension $\eta(y) \times \eta(y)$. The elements of Q are given by the following expressions:

$$Q_{1i} = Q_{i1} = [D_i r(\bar{y}) - \sum_{j=2}^{\eta(y)} (D_{ij}^2 r(\bar{y}) \cdot \bar{y}_j)]/2, \quad \text{for } i = 2, \dots, \eta(y)$$

$$Q_{ij} = Q_{ji} = (1/2) D_{ij}^2 r(\bar{y}), \quad \text{for } i, j = 2, \dots, \eta(y)$$

and

$$Q_{11} = r(\bar{y}) - \sum_{i=2}^{\eta(y)} D_i r(\bar{y}) \cdot \bar{y}_i + (1/2) \sum_{i=2}^{\eta(y)} \sum_{j=2}^{\eta(y)} D_{ij}^2 r(\bar{y}) \cdot \bar{y}_i \cdot \bar{y}_j$$

For reasons that will be made clear below, it is important for our method that the ordering of y, and hence the ordering of the elements of Q, be exactly as described here. The following is the linear-quadratic dynamic programming problem obtained from this approximation:

$$v(z, s) = \max\{y^T Q y + \beta E[v(z', s')|z]\} \tag{14}$$

subject to (2) and (3).[10]

Solving the Dynamic Program by Successive Approximations

Problem (14) is a standard linear-quadratic dynamic programming problem. Under suitable conditions, the optimal value function, v, exists, solves this functional equation, and is quadratic. Given this, the associated policy functions are linear. In this section, we do not attempt to survey the extensive literature (see, e.g., Hansen and Sargent [forthcoming]) describing efficient techniques for solving such a problem. Instead, we describe a simple algorithm that is easy to implement and understand. A computer program designed to carry out these computations is easy to write and debug. One advantage of this is that it saves time for the researcher. A second advantage is that it will be easy to modify the method to solve for equilibria that are not solutions to social planner's problems, as in economies with taxes or other distortions, or for studying an important class of heterogeneous-agent economies, including those with n-period-lived overlapping generations. These will be described in later sections.

The optimal value function for problem (14) is identical, save for a constant, for any covariance matrix of ϵ. As a result, the optimal policy function is independent of this covariance matrix. Given this, we solve the programming problem for the certainty case, where the covariance matrix has been set equal to zero. In other words, we solve the version of (14) in which the expectations operator has been dropped and ϵ' in (2) has been replaced with its mean, zero.

The method of successive approximations is used to compute the optimal value function, v. Following this method, we generate a sequence of approximations to v that for well-behaved problems will converge to the optimal value function.[11] To solve this problem, an initial quadratic approximation for the value function, v^0, is selected and the standard Bellman mapping is used to obtain the sequence of approximations. In particular, given the nth element of this sequence, the $n+1$st element is obtained as follows:

$$\begin{aligned} v^{n+1}(z, s) &= \max\{y^T Q y + \beta v^n(z', s')\} \\ \text{s.t.} \quad [z', s']_i &= \sum_{j \leq \eta(y)} B_{ij} y_j \quad \text{for } i = 1, \dots, \eta(z, s). \end{aligned} \tag{15}$$

The B_{ij}'s in the above constraints are taken directly from equations (2) and (3). To obtain v^{n+1}, we first substitute the constraint into the right side of (15) in order to eliminate z' and s' from the problem. This yields a quadratic expression in (z, s, d). Next, the first order-conditions are used to solve for the vector d as a linear function of z and s. Substituting these into (15), we obtain the next approximation, which is a quadratic function of (z, s). If the problem is well behaved, this procedure is repeated until $\|v^{n+1} - v^n\| < \xi$, where ξ is some small positive real number.

We now describe these iterations in greater detail:

Step 1. Choose some arbitrary negative semidefinite matrix, v^0, of size $\eta(z, s) \times \eta(z, s)$. A possible candidate is a matrix with small negative numbers on the diagonal and zeros for the off-diagonal elements. Once again, the ordering of the columns of this matrix is very important: the first $\eta(z)$ columns contain coefficients corresponding to terms involving elements of z (thus the first column contains the linear terms), and the last $\eta(s)$ columns contain coefficients corresponding to terms involving elements of s.

Steps 2 through 5 describe how to generate successive approximations of the optimal value function. In particular, we describe how to compute v^{n+1} given an approximation v^n. These four steps are repeated until the sequence of approximations has converged.

Step 2. Let x be the stacked vector (y, z', s'), which is equal to (z, s, d, z', s'). Construct a matrix $R^{[\eta(x)]}$, which is of dimension $\eta(x) \times \eta(x)$, that contains the matrix Q (with its elements in the order described above) in the top left corner and the matrix βv^n in the lower right corner. The remaining elements of $R^{[\eta(x)]}$ are set equal to zero. This enables us to write the expression $y^T Qy + \beta v(z', s')$ from (15) as a single quadratic form, $x^T R^{(\eta(x))} x$.

The next two steps describe how to compute $v^{n+1}(z, s)$ by eliminating the variables s', z', and d from $x^T R^{[\eta(x)]} x$, using the constraints in (15) and the first-order conditions. We begin by eliminating the last element of x, and then proceed to eliminate the second-to-last element, and so on, until only the elements of z and s remain. To eliminate a particular element of x, say, x_j, we must be able to express x_j as a linear function of the variables x_i, where i is less than j. This requirement will be satisfied given the particular way in which we have ordered the elements of x.

Each time a linear expression is used to eliminate a component of x, the form of the quadratic objective is altered. For example, when the first variable, which is the last component of x, is eliminated, the quadratic objective becomes $x^T R^{[\eta(x)-1]} x$, where $R^{[\eta(x)-1]}$ is the same array as $R^{[\eta(x)]}$, with the entries changed to reflect the substitution. In particular, the last row and column are now filled with zeros.[12]

To make this more precise, suppose that after some substitutions, we are left with the quadratic form, $x^T R^{(j)} x$, where $j > \eta(z, s)$. The jth component of x must be eliminated next. Using (2), (3), or a first order condition makes it possible to express x_j in terms of x_i, $i < j$, as follows:

$$x_j = \sum_{i<j} \gamma_i x_i. \tag{16}$$

Substituting (16) into the quadratic objective yields a new quadratic objective, $x^T R^{(j-1)} x$, where the components in the first $j-1$ rows and columns of $R^{(j-1)}$ are given by

$$R^{(j-1)}_{ih} = R^{(j)}_{ih} + R^{(j)}_{jh}\gamma_i + R^{(j)}_{ji}\gamma_h + R^{(j)}_{jj}\gamma_i\gamma_h, \quad \text{for } i, h = 1, \dots, j-1 \tag{17}$$

The remaining elements of this $\eta(x) \times \eta(x)$ array are equal to zero.

There is a matrix algebraic alternative to (17) that may be easier to implement on the computer, especially if one is using a matrix programming language:

$$R^{(j-1)} = \Gamma^T R^{(j)} \Gamma, \quad \text{where } \Gamma = \begin{bmatrix} I_{j-1} \\ \gamma_1 \dots, \gamma_{j-1} \end{bmatrix} \tag{18}$$

and I_{j-1} is a $j-1$ dimensional identity matrix. Note that this formula is written under the assumption that $R^{(j)}$ is of dimension $j \times j$, meaning that the last $\eta(y) - j$ rows and columns of $R^{(j)}$ (which are all zeros) have been eliminated. In all other parts of this chapter, the R matrices are assumed to be padded with zeros so that they are of dimension $\eta(x) \times \eta(x)$.

After repeated application of this procedure, we obtain the quadratic form $x^T R^{\eta(z,s)]} x$. The matrix v^{n+1}, defined by the mapping (15), is simply the first $\eta(z, s)$ rows and columns of $R^{[\eta(z,s)]}$. We now describe more precisely the particular substitutions that are made in order to obtain v^{n+1}.

Step 3. In this step, we substitute expressions for s' and z', given by the constraints in equation (15), into the objective. These constraints, which determine the last $\eta(z, s)$ elements of x, are the following:

$$x_i = \sum_{j \le J} B_{ij} x_j,$$

where

$$i = \eta(z, s, d) + 1, \dots, \eta(x)$$

and

$$J = \eta(z, s, d).$$

As explained above, we first eliminate $x_{\eta(x)}$, the last element of s'. Using equation 17 with the coefficients $B_{\eta(x),j}$ in place of the γ's, we obtain the matrix $R^{[\eta(x)-1]}$. After all components of s' and z' have been eliminated, we are left with the quadratic form $x^T R^{[\eta(z,s,d)]} x$.

Step 4. The next $\eta(d)$ variables, which are the components of d, are eliminated by using the first-order conditions for the maximization problem on the right side of (15), beginning with $x_{\eta(z,s,d)} = d_{\eta(d)}$. The following is the first-order condition with respect to the jth component of x, assuming that all components of x with index greater than j have already been eliminated:

$$x_j = -\sum_{i=1}^{j-1} (R^{(j)}_{ji} / R^{(j)}_{jj}) x_i, \quad j = \eta(z, s) + 1, \dots, \eta(z, s, d). \tag{19}$$

At this stage, it is important to examine whether the second-order conditions are satisfied by checking whether $R^{(j)}_{jj}$ is less than zero. In cases where the return function, r, is strictly concave, these conditions will be satisfied if no errors have been made in implementing the algorithm. However, in other cases, say, where the return function is not bounded from above, a violation of the second-order conditions at some stage indicates that a maximum does not exist, and hence indicates a failure of this method to find the optimal value function.

As before, (17) is used to compute $R^{(j-1)}$ where the γ's are given by the coefficients in (19). After all of the decision variables have been eliminated, we are left with the matrix $R^{[\eta(z,s)]}$.

Step 5. Set v^{n+1} equal to the matrix formed by the first $\eta(z, s)$ rows and columns of $R^{[\eta(z,s)]}$. If all the elements of v^{n+1} are sufficiently close to the corresponding elements of v^n (for example, if the biggest difference is less than .00001), stop the iterations.[13] If not, repeat these steps again beginning with Step 2, using v^{n+1} in place of v^n.

Step 6. Once this sequence of successive approximations has converged, the first-order conditions from the last iteration, given by (19), can be used to derive the equilibrium policy functions. Equation (19) can be rewritten as follows:

$$d_j = \sum_{i<K} C_{ij} x_i, \quad j = 1, \ldots, \eta(d),$$

where

$$C_{ij} = \frac{-R_{Ki}^{(K)}}{R_{KK}^{(K)}},$$

and $K = \eta(z, s) + j$.

In this form, the expression for d_j is a function not only of the state variables, z and s, but also of the decision variables with indices from 1 to $j-1$. These policy functions can be expressed in terms of the state variables alone as follows:

$$d_j = \sum_{i=1}^{\eta(z,s)} D_{ij} x_i, \tag{20}$$

where for each i,

$$D_{i1} = C_{i1},$$

$$D_{i2} = C_{i2} + C_{\eta(z,s)+1,2} D_{i1},$$

and

$$D_{ij} = C_{ij} + \sum_{h<j} [C_{\eta(z,s)+h,j} D_{ih}], \quad j = 3, \ldots, \eta(d).$$

Step 7. Finally, it is wise to check whether the steady state implied by (20) is the same as the steady state for the original nonlinear planner's problem, $(\bar{z}, \bar{s}, \bar{d})$, defined by (12). The simplest way to do this is to substitute $\bar{z}$ and $\bar{s}$ into the right side of (20), and then check whether the resulting vector of decisions equals $\bar{d}$ to, say, six decimal places.

4. Recursive Competitive Equilibrium for Homogeneous-Agent Economies

For many applications, including many in public finance and monetary economics, it is not possible to find equilibrium allocations by solving a planning problem. Instead, it is necessary to solve for equilibrium allocations directly by solving a

In this section, we describe two applications where the methods discussed in the previous two sections cannot be applied directly. Our first example is an economy with distorting taxes, and the second is an economy with money introduced by imposing a cash-in-advance constraint. Methods for solving linear-quadratic versions of these two examples are described in Section 5.

The problem faced by households in our tax example, as well as in many other applications involving nonmonetary distortions, is a dynamic programming problem of the following form:

$$v(z, S, s) = \max\{r(z, S, s, D, d) + \beta E[v(z', S', s')|z]\}, \tag{21}$$

$$\text{s.t.} \quad z' = A(z) + \epsilon' \tag{22}$$

$$s' = B(z, S, s, D, d) \tag{23}$$

$$S' = B(z, S, S, D, D) \tag{24}$$

$$D = \mathbf{D}(z, S). \tag{25}$$

As in section 2, z is a vector of exogenous state variables, possibly stochastic, that evolves according to the first-order Markov process (22), where A is a linear function. The variable ϵ is a mean zero random vector with finite variance. In addition, s is a vector of endogenous household-specific state variables, and S is a vector containing their economywide (per capita) values.[14] Similarly, d is a vector of household decision variables, and D is the vector of per capita values of these same variables. Equations (23) and (24) describe the evolution of s and S, where B is a linear function. Note that (24) is obtained from (23) by aggregating over all households.

The function $\mathbf{D}$ in equation (25) expresses the relationship between the per capita values of the decision variables, D, and the state variables, z and S. This function does not describe a feature of the environment but is instead determined as part of the equilibrium. The primary goal of the next section is to describe a computational method for finding a function $\mathbf{D}$ that satisfies our definition of equilibrium.

More specifically, we wish to find a *recursive competitive equilibrium* (RCE), which consists of decision rules for the households, $d = d(z, S, s)$; a rule determining the per capita values of these variables, $D = \mathbf{D}(z, S)$; and a value function, $v(z, S, s)$, such that[15]

1) given the aggregate decision rules, D, the value function, v, satisfies equation (21) and d are the associated decision rules; and

2) the function $\mathbf{D}$ satisfies the relationship $\mathbf{D}(z, S) = d(z, S, S)$.

The Basic Model with Taxes

Our first example is a version of the basic model from Section 2 with taxes on labor and capital income. The particular decentralized economy that we consider consists of a large number N of identical households ...

capital in period 0 and one unit of time in each period, which is spent either working or enjoying leisure. The households receive income in each period from capital and labor, which is used to finance consumption and investments in new capital. Consumption, leisure, and investment are chosen to maximize (4), subject to the following sequence of budget constraints (one for each t from zero to infinity):

$$c_t + i_t = (1 - \tau_h)w_t h_t + (1 - \tau_k)r_t k_t + \tau_k \delta k_t + TR_t. \tag{26}$$

In this equation, the variables w_t and r_t denote the wage rate and rental rate, respectively. The parameters τ_h and τ_k are the tax rate on labor income and the tax rate on capital income net of depreciation, $(r_t - \delta)k_t$. The capital stock owned by a given household evolves according to (8). The last term, TR_t, is a per capita lump sum transfer from the government to the households.

A firm in this economy purchases labor and capital services from the households and uses these to produce output, y_t^f, according to the technology given by (5) and (6). (The superscript f indicates quantities chosen by the firm.) Given that the technology displays constant returns to scale, no loss in generality is incurred by assuming that there is only one firm. The first-order conditions for the firm's profit maximization problem are

$$w_t = z_t F_2(k_t^f, h_t^f)$$

and

$$r_t = z_t F_1(k_t^f, h_t^f),$$

where k_t^f is the amount of capital that the firm rents from the households. Market clearing requires that $k_t^f = K_t N$ and $h_t^f = H_t N$, where N is the number of households, K_t is the per capita stock of capital, and H_t is per capita hours worked. Substituting this into the above first-order conditions, and using the fact that constant returns imply that the marginal products are homogeneous of degree zero, we obtain the following equilibrium expressions:

$$w_t = w(z_t, K_t, H_t) = z_t F_2(K_t, H_t), \tag{27}$$

and

$$r_t = r(z_t, K_t, H_t) = z_t F_1(K_t, H_t). \tag{28}$$

Constant returns also imply that in equilibrium, payments to factors of production fully exhaust revenues and, as a result, dividends are zero.

The role of the government in this economy is simply to collect tax revenue and return it to the households as a lump sum transfer. This implies that the government budget constraint is

$$TR_t = \tau_h w_t H_t + \tau_k (r_t - \delta) K_t. \tag{29}$$

The problem faced by a particular household can be expressed in the form of (21) by making a series of substitutions. First, (29) is substituted into (26) by eliminating TR_t. Next, (27) and (28) are substituted into (26), eliminating w_t and r_t, respectively. Finally, (26) is solved for c_t, and the result is substituted into the utility function (4). After these substitutions, the household's optimization problem can be written as the following dynamic programming problem:

$$\begin{aligned} v(z, K, k) &= \max\{r(z, K, k, I, H, i, h) + \beta E[v(z', K', k')|z]\} \\ \text{s.t.}\quad & z' = A(z) + \epsilon' \\ & K' = (1 - \delta)K + I \\ & k' = (1 - \delta)k + i \\ & I = \mathbf{I}(z, K) \text{ and } H = \mathbf{H}(z, K). \end{aligned} \tag{30}$$

The function $r(z, K, k, I, H, i, h)$ is equal to $U(c, 1 - h)$, where c is given by $w(z, K, H)[h + \tau_h(H - h)] + r(z, K, H)[k + \tau_k(K - k)] + \tau_k^\delta(k - K) - i$.

The functions $\mathbf{I}$ and $\mathbf{H}$ describe the relationship perceived by households between the aggregate decision variables and the state of the economy. We are interested in finding functional forms for $\mathbf{I}$ and $\mathbf{H}$ that satisfy the definition of a recursive competitive equilibrium applied to this example.

This problem can easily be mapped into the framework described at the beginning of the section. The only exogenous state variable is the technology shock, z, and the only endogenous state variable is the capital stock, K.[16] The decision variables are $d = (h, i)$, and the function $B(z, K, k, I, H, i, h)$ for this example is $(1 - \delta)k + i$. Finally, the analog to the function $\mathbf{D}$ in (25) is the pair of functions $\mathbf{I}$ and $\mathbf{H}$.

The Basic Model with Money

Leaving the preferences and technology of our basic model unchanged, fiat money will not be valued in equilibrium.[17] This follows from the fact that money would be dominated in rate of return by privately issued assets. The two most common ways of overcoming this obstacle are to include money as an argument in the utility function or to assume that previously accumulated cash balances are required for the purchase of some consumption goods (cash-in-advance).[18] In this section we will describe an example that illustrates the second approach.

Households choose consumption and leisure to maximize

$$E \sum_{t=0}^{\infty} \beta^t U(c_{1t}, c_{2t}, \ell_t), \quad 0 < \beta < 1, \tag{31}$$

where c_1 is consumption of the "cash good," c_2 is consumption of the "credit good," and ℓ is leisure. The period utility function, U, is bounded, continuously differentiable, strictly increasing, and strictly concave. In addition, Inada conditions are required to ensure that agents consume positive quantities of both consumption goods.

The period budget constraint is

$$c_{1t} + c_{2t} + i_t + \frac{m_{t+1}}{p_t} \leq w_t h_t + r_t k_t + \frac{m_t}{p_t} + \frac{TR_t}{p_t}. \tag{32}$$

This constraint reflects the assumption that c_1 and c_2, in addition to investment, are perfect substitutes in production, and hence sell at the same relative price. Households enter the period with nominal balances equal to m_t, which is augmented with a lump sum transfer of newly printed money, TR_t. Using our notational convention, M_t denotes beginning-of-period (pretransfer) per capita money balances, and m_t denotes the money holdings of a particular household. Thus, $TR_t = M_{t+1} - M_t$. The price level is denoted by p_t.

Purchases of the cash good, c_{1t}, must be financed with nominal cash holdings at the beginning of the period (post-transfer). This requirement is formalized by the cash-in-advance constraint,

$$p_t c_{1t} \leq m_t + TR_t. \tag{33}$$

The resource constraint is $c_{1t} + c_{2t} + i_t \leq y_t$, where y_t is produced according to the production function (4). This implies that the equilibrium wage rate and rental rate are given in equations (27) and (28), respectively.

The money supply, M_t, evolves according to the following rule:

$$M_{t+1} = gM_t. \tag{34}$$

The monetary growth factor, g, is constant over time, but, as in the case of the tax rates in the previous example, a natural extension is to model g as an exogenous state variable or as depending on the economy-wide state.

In this example, as well as in most applications involving cash-in-advance models, we impose conditions on the money growth rate such that (33) holds with equality (that is, the Lagrange multiplier associated with this constraint is positive in equilibrium). The precise form that this restriction takes depends on the form of the utility function. In general, this restriction is equivalent to requiring that an appropriately defined nominal interest rate be positive.

Our solution method requires that all variables fluctuate around a constant mean. However, if g is greater than one, both M and p in this example will grow without limit. This motivates introducing the following change of variables:

$$\hat{m}_t = m_t/M_t \quad \text{and} \quad \hat{p}_t = p_t/M_{t+1}. \tag{35}$$

With this change in variables, assuming that the cash-in-advance constraint is binding, the dynamic programming problem solved by households is

$$v(z, K, k, \hat{m}) = \max\{U(c_1, c_2, 1-h) + \beta E v(z', K', k', \hat{m}')$$

$$\text{s.t.} \quad z' = A(z) + \epsilon'$$

$$K' = (1-\delta)K + I$$

$$k' = (1-\delta)k + i$$

$$c_1 + c_2 + i + \frac{\hat{m}'}{\hat{p}} = w(z, K, H)h + r(z, K, H)k + \frac{(\hat{m} + g - 1)}{(g \cdot \hat{p})}$$

$$c_1 = \frac{(\hat{m} + g - 1)}{(g \cdot \hat{p})}$$

$$I = \mathbf{I}(z, K), \quad H = \mathbf{H}(z, K), \quad \hat{p} = \mathbf{P}(z, K). \tag{36}$$

An important feature of this problem, which is absent in (30), is the function $\mathbf{P}$, which expresses the relationship between the price level and the state of the economy. Because of this feature, this problem cannot be mapped into the notation of problem (21). In that problem, there is no analog to relative money holdings, $\hat{m}$, or the price level, $\hat{p}$. However, the following modified version of problem (21) incorporates these features. This more general formulation would also apply to other applications involving money in the utility function or cash-in-advance in addition to the particular example described above.

$$v(z, S, s, m) = \max\{r(z, S, s, m, D, p, d, m') + \beta E v(z', S', s', m')\} \tag{37}$$

$$\text{s.t.} \quad z' = A(z) + \epsilon' \tag{38}$$

$$s' = B(z, S, s, m, D, d, p, m') \tag{39}$$

$$S' = B(z, S, S, 1, D, D, p, 1) \tag{40}$$

$$D = \mathbf{D}(z, S), \; p = \mathbf{P}(z, S).$$

In this problem, m and p are the household's nominal money holdings and the price level, both expressed relative to the per capita money supply. Thus, they correspond to $\hat{m}$ and $\hat{p}$ in the above example.

A *recursive competitive equilibrium* consists of a set of decision rules for the household, $d = d(z, S, s, m)$; a decision rule determining the amount of money the household carries into the next period, $m' = m(z, S, s, m)$; a set of aggregate decision rules, $D = \mathbf{D}(z, S)$; a function determining the aggregate price level, $p = \mathbf{P}(z, S)$; and a value function, $v(z, S, s, m)$, such that

1) given the functions $\mathbf{D}$ and $\mathbf{P}$, the value function, v, satisfies equation (37), and d and m' are the associated decision rules; and

2) given the pricing function, $\mathbf{P}$, individual decisions are consistent with aggregate outcomes:

$$\mathbf{D}(z, S) = d(z, S, S, 1) \quad \text{and} \quad 1 = m(z, S, S, 1)$$

Note that in equilibrium, m' must equal one since m' is defined to be money holdings relative to the per capita money supply.

It is straightforward to express our cash-in-advance economy in terms of this notation. As in the first example, $d = (i, h)$, and the function B is simply $(1-\delta)k + i$. The return function, r, is given by

$$r(z, K, k, \hat{m}, I, H, \hat{p}, i, h, \hat{m}') = U(c_1, c_2, 1-h)$$

where

$$c_1 = \frac{(\hat{m} + g - 1)}{(g \cdot \hat{p})},$$

and

$$c_2 = w(z, K, H)h + r(z, K, H)k - i - \frac{\hat{m}'}{\hat{p}}.$$

5. Solving for a Recursive Competitive Equilibrium

In this section, we describe a method for finding a function **D** (for an economy without money) or a pair of functions **D** and **P** (for a monetary economy) that satisfies the definition of a RCE. As in Section 3, we consider economies for which the return function is quadratic. Since our examples do not generally deliver quadratic objectives, we again make use of the quadratic approximation procedure described in Section 3. We first explain how to compute a RCE for a nonmonetary economy by using methods similar to the successive approximations described in Section 3. Next, we show how essentially the same methods can be applied to economies with money.

We begin by considering an economy where the problem solved by households is (21). As in section 3, z in equation (22) is an $\eta(z) \times 1$ vector of exogenous state variables, the first component of which is assumed to be constant over time (equal to one, without loss of generality). We continue to use the function $\eta(x)$ to denote the length of a column vector, x.

The steady state for the certainty version of this economy, $(\bar{z}, \bar{S}, \bar{s}, \bar{D}, \bar{d})$, which is required in computing the quadratic approximation of the return function, r, is the solution to the following set of equations:

$$\begin{aligned}
&\bar{z} = A(\bar{z});\\
&r_d(\bar{z}, \bar{S}, \bar{s}, \bar{D}, \bar{d})\\
&\quad + \beta r_s(\bar{z}, \bar{S}, \bar{s}, \bar{D}, \bar{d})[I - \beta B_s(\bar{z}, \bar{S}, \bar{s}, \bar{D}, \bar{d})]^{-1} B_d(\bar{z}, \bar{S}, \bar{s}, \bar{D}, \bar{d}) = 0\\
&\bar{S} = B(\bar{z}, \bar{S}, \bar{S}, \bar{D}, \bar{D});\\
&\bar{d} = \bar{D}, \bar{s} = \bar{S}.
\end{aligned} \tag{41}$$

Note that in (41) r_d, r_s, etc., have the same definition as (12).

Define y as the stacked vector, $(z, S, s, D_1, d_1, \ldots, D_{\eta(d)}, d_{\eta(d)})$, where the subscript denotes a particular component of D or d. Given the steady state of y, the quadratic approximation can be formed in precisely the manner described in Section 3. From this, we obtain the following linear-quadratic formulation of the household's problem, where the functions $\mathbf{D}_i$ are the unknown aggregate decision rules:

$$\begin{aligned}
&v(z, S, s) = \max\{y^T Qy + \beta E[v(z', S', s')|z]\}\\
&\text{s.t.}\quad (22)\text{–}(24)\\
&\qquad D_i = \mathbf{D}_i(z, S, D_1, \ldots, D_{i-1}),
\end{aligned} \tag{42}$$

where the $\mathbf{D}_i$, $i = 1, \ldots, \eta(d)$, are linear functions.

Finding a Recursive Competitive Equilibrium by Successive Approximations

Our computational procedure for finding the functions $\mathbf{D}_i$ that satisfy the requirements of a RCE for a linear-quadratic economy makes heavy use of the methods described in Section 3. As in the earlier section, we focus only on the certainty version of the household's problem (42), since the decision rules will be independent of the variance of ϵ. Successive approximations of the optimal value function, v, are obtained by iterating on the following mapping:

$$\begin{aligned}
&v^{n+1}(z, S, s) = \max\{y^T Qy + \beta v^n(z', S', s')\}\\
&\text{s.t.}\quad (22)\text{–}(24)\\
&\qquad D_i = \mathbf{D}_i^n(z, S, D_1, \ldots, D_{i-1}), \quad \text{for } i = 1, \ldots, \eta(d).
\end{aligned} \tag{43}$$

The functions $\mathbf{D}_i^n$ are the linear aggregate decision rules associated with the nth approximation of v. The precise way in which these functions are computed is described in Step 4 below.

Step 1. Choose a negative semidefinite matrix, v^0, of size $\eta(z, S, s) \times \eta(z, S, s)$.

Steps 2 through 7 describe how to obtain successive approximations of the value function, v. Given a matrix v^n, these steps explain how to compute v^{n+1}.

Step 2. Define x to be the stacked vector $(y, z', S', s') = (z, S, s, D_1, d_1, \ldots, D_{\eta(d)}, d_{\eta(d)}, z', S', s')$. Construct a matrix $R^{[\eta(x)]}$, which is of dimension $\eta(x) \times \eta(x)$, and which contains the matrix Q in the top left corner and the matrix βv^n in the lower right corner. All other elements are set equal to zero. The quadratic expression on the right side of (43) can now be written $x^T R^{[\eta(x)]} x$.

Step 3. Eliminate s', S', and z' by using the linear laws of motion, (22)–(24). This is done by using equation (17) from Section 3. After these substitutions, the quadratic expression becomes $x^T R^{[\eta(y)]} x$.

The next three steps are used to eliminate the aggregate and individual decision variables, D_j and d_j. Beginning with $j = \eta(d)$, these steps must be repeated $\eta(d)$ times to eliminate each of the D_j and d_j in turn. The description here assumes that the jth decision variable (D_j and d_j), which corresponds to the $J-1$st and Jth elements of x, where $J \equiv \eta(z, S, s) + 2j$, is being eliminated. Decision variables with index greater than j are assumed to have already been eliminated.

Step 4. To obtain the function $\mathbf{D}_j^n$, consider the first order condition with respect to d_j:

$$\sum_{i \leq J} R_{Ji}^{(J)} x_i = 0, \tag{44}$$

where $J = \eta(z, S, s) + 2j$. At this point it is important to examine if the second-order conditions are satisfied by checking whether $R_{JJ}^{(J)}$ is less than zero.

Substitute the aggregate consistency conditions into (44) by setting $s = S$ and $d_i = D_i$ for $i = 1, \ldots, j$, thereby eliminating s and the remaining components of d. Solving for D_j, we obtain the aggregate decision rule $\mathbf{D}_j^n$:

$$x_{J-1} = D_j = \sum_{i=1}^{J-2} \delta_i x_i, \tag{45}$$

where

$$\delta_i = \begin{cases} R_{Ji}^{(J)}/\bar{R} & \text{for } i = 1, \ldots, \eta(z) \\ (R_{Ji}^{(J)} + R_{J,i+\eta(S)}^{(J)})/\bar{R} & \text{for } i = \eta(z)+1, \ldots, \eta(z, S) \\ 0 & \text{for } i = \eta(z, S)+1, \ldots, \eta(z, S, s) \\ (R_{Ji}^{(J)} + R_{J,i+1}^{(J)})/\bar{R} & \text{for } i = \eta(z, S, s)+1, \ldots, J-3 \\ & \quad \text{(increments of 2)} \\ 0 & \text{for } i = \eta(z, S, s)+2, \ldots, J-2 \\ & \quad \text{(increments of 2)} \end{cases}$$

and

$$\bar{R} = -(R_{J,J-1}^{(J)} + R_{JJ}^{(J)}).$$

The first set of δ's are the coefficients on the components of z in (45). The remaining four sets of δ's are coefficients on the components of S, s, D, and d, respectively. Since (45) is an aggregate decision rule, the coefficients on s and d are equal to zero.

Step 5. Solve equation (44) for x_J and use the resulting linear expression to eliminate $x_J = d_j$ from the right side of (43), using the substitution procedure described in Section 3.

Step 6. Use equation (45) to eliminate $x_{J-1} = D_j$.

Steps 4 through 6 are repeated until all decision variables have been eliminated. After these substitutions, the right side of (43) becomes $x^T R^{\eta(z,S,s)} x$.

Step 7. Define v^{n+1} to be the matrix formed by the first $\eta(z, S, s)$ rows and columns of $R^{\eta(z,S,s)}$. Compare the elements of v^{n+1} with the elements of v^n, ignoring the (1,1) element. If they are sufficiently close, stop the iterations. If not, repeat the procedure beginning with Step 2, using v^{n+1} in place of v^n.

Once the iterations have converged, the equilibrium aggregate decision rules can be computed from the set of $\eta(d)$ equations (45) obtained in the last iteration. The procedure for obtaining these is analogous to the procedure described in Step 6 at the end of Section 3. Finally, one should check whether the steady states obtained from solving (41) are the same as the steady states implied by the linear equilibrium decision rules.

Solving for a Recursive Competitive Equilibrium in a Monetary Economy

We now explain how this method can be modified to solve for a RCE in a monetary model, where money is introduced either through a cash-in-advance constraint, as in the model described in Section 4, or by introducing money directly into the utility function.

The problem solved by households in these models is stated in equation (37). As usual, the first component of z is equal to one. The additional variables, m and p, are defined as in the basic model with money in Section 4. Both of these are one-dimensional variables.

The quadratic approximation of the return function, r, is formed in the same way as above, where the vector y is defined to be the stacked vector $(z, S, s, m, D_1, d_1, \ldots, D_{\eta(d)}, d_{\eta(d)}, p, m')$. The steady state is computed by solving a set of equations analogous to (12) in Section 3 or (41), noting that steady-state money holdings, $\bar{m}$, are equal to one.

The steps involved in generating successive approximations are very similar to those described above for solving a social planning problem. Successive approximations are computed by iterating on the following mapping, which is similar to

(43):

$$v^{n+1}(z, S, s, m) = \max\{y^T Qy + \beta v^n(z', S', s', m')\}$$
$$\text{s.t.} \quad (38)\text{–}(40)$$
$$D_i = \mathbf{D}_i^n(z, S, D_1, \ldots, D_{i-1}), i = 1, \ldots, \eta(d),$$
$$p = \mathbf{P}^n(z, S, D_1, \ldots, D_{\eta(d)}). \tag{46}$$

The functions $\mathbf{D}_i^n$ and $\mathbf{P}^n$ are the aggregate decision rules and pricing function associated with the nth approximation of v.

Steps 1–3 can be followed almost exactly as described, except that v^0 must be of dimension $[\eta(z, S, s) + 1] \times [\eta(z, S, s) + 1]$ and x is defined to be the stacked vector, $(z, S, s, m, D_1, d_1, \ldots, D_{\eta(d)}, d_{\eta(d)}, p, m', z', S', s', m')$. Notice that m' appears in this vector twice since it enters the value function for the next period as well as the current return function. The second m' is eliminated in Step 3 by using the equation $x_{\eta(x)} = x_{\eta(y)}$, that is, by setting the second m' equal to the first m'.

Step 4 for this problem differs from the nonmonetary case since the pricing function, $\mathbf{P}^n(z, S, D_1, \ldots, D_{\eta(d)})$, must be obtained in addition to the aggregate decision rules, denoted by $\mathbf{D}_i^n, \ldots, \mathbf{D}_{\eta(d)}^n$. The function $\mathbf{P}^n$ is computed from the first-order condition associated with m' by imposing the aggregate consistency conditions—$s = S, m = m' = 1$, and $d_i = D_i, i = 1, \ldots, \eta(d)$—and then solving for p. The expression $p = \mathbf{P}^n[z, S, D_1, \ldots, D_{\eta(d)}]$ is used to eliminate $x_{\eta(y)-1}$ as described in Step 6.

The rest of the procedure is unchanged from the nonmonetary one except that the accounting is slightly different because of the additional components of x. The decision variables, d and D, are eliminated as explained in Steps 4–6, and v^{n+1} is computed. Successive approximations of v are computed until they converge. Finally, the equilibrium decision rules and pricing function are computed from the functions $\mathbf{P}^n$ and $\mathbf{D}_i^n, i = 1, \ldots, \eta(d)$, associated with the last approximation of v.

6. Extensions to Heterogeneous-Agent Economies

An advantage of the methods we have described in this chapter is that they can be extended in a straightforward manner to an important class of economies in which agents are not ex ante identical. In this final section, we describe how to compute an equilibrium for an extension of the basic model in which agents differ according to preferences and initial capital holdings. More complicated heterogeneous-agent environments, including economies with n-period–lived overlapping generations, can be studied with methods similar to the one used for this example.[19]

Suppose that the economy consists of N types of households, with λ_i being the fraction of type $i = 1, \ldots, N$. It follows that the total measure of households is one. A household of type i solves the following problem:

$$\max E \sum_{t=0}^{\infty} \beta^t U_i(c_{it}, \ell_{it}), \quad 0 < \beta < 1 \tag{47}$$
$$\text{s.t.} \quad h_{it} + \ell_{it} = 1$$
$$z_{t+1} = A(z_t) + \epsilon_{t+1} \tag{48}$$
$$c_{it} + x_{it} = w_t h_{it} + r_t k_{it} \tag{49}$$
$$k_{it+1} = (1 - \delta)k_{it} + x_{it} \tag{50}$$
$$w_t = z_t F_H(K_t, H_t) \tag{51}$$
$$r_t = z_t F_K(K_t, H_t). \tag{52}$$

The initial k_{i0} is given as well as the stochastic process generating sequences $\{K_t, H_t\}_{t=0}^{\infty}$.

The variables $c_{it}, x_{it}, h_{it}, \ell_{it}$, and k_{it} denote consumption, investment, hours worked, leisure, and capital stock of household i in time t.[20] Equations (51) and (52) are the equilibrium wage and rental rates as derived for the basic model with taxes in Section 4 under the assumption that there is a single constant-returns-to-scale technology. As usual, $K_t = \sum \lambda_i K_{it}$ and $H_t = \sum \lambda_i H_{it}$ are the per capita capital stock and hours worked, respectively, where K_{it} and H_{it} are the per capita capital stock and hours for households of type i. Equation (48) is the law of motion for the technology shock, where A is a linear function and ϵ is an i.i.d. random variable with finite variance.

As can be seen from the budget constraint (49), agents are permitted to use income from capital and labor to purchase units of current-period output (consumption and investment goods) only. They are not permitted to purchase (or sell) state-contingent claims to next-period units of output. This is not necessarily an innocuous assumption, as it is in economies where agents are identical. In addition, allowing these trades does complicate the solution procedure somewhat. Ríos-Rull (1992a) describes how to compute a recursive competitive equilibrium of an overlapping-generations economy in which these sorts of trades are permitted.

In our recursive formulation of this economy, we utilize the following notational conventions: k_i is the capital stock of a particular household of type i; K_j, for $j = 1, \ldots, N$, are the per capita capital stocks of households of types j; and $\mathbf{K} = (K_1, \ldots, K_N)^T$ is a vector describing the entire distribution of capital stocks. These same conventions apply to the decision variables, hours, h, and investment, x.

If we substitute (51) and (52) into (49), solve (49) for c_{it} and substitute the resulting function into the utility function, household i's optimization problem can be expressed as the following dynamic program:

$$v_i(z, \mathbf{K}, k_i) = \max\{r_i(z, \mathbf{K}, k_i, \mathbf{X}, \mathbf{H}, x_i, h_i) + \beta E[v_i(z', \mathbf{K}', k_i')|z]\} \tag{53}$$

$$\text{s.t.} \quad z' = A(z) + \epsilon';$$

$$k_i' = (1-\delta)k_i + x_i; \tag{54}$$

$$K_j' = (1-\delta)K_j + X_j, \text{ for } j = 1, \ldots, N; \tag{55}$$

$$H_j = \mathrm{H}_j(z, \mathbf{K}), X_j = \mathrm{X}_j(z, \mathbf{K}) \text{ for } j = 1, \ldots, N. \tag{56}$$

Notice that although factor prices depend only on the per capita capital stock, the state of the economy includes the vector of per capita stocks held by each type. Equation (55) is the law of motion for the K_j. Equation (56) states that the per capita hours and per capita investment (for each j) are given functions of the state of the economy.

A *recursive competitive equilibrium* for this economy consists of a set of decision rules for households of type i, $\mathrm{h}_i(z, \mathbf{K}, k_i)$ and $\mathrm{x}_i(z, \mathbf{K}, k_i)$ for $i = 1, \ldots, N$; a set of per capita decision rules, $\mathrm{H}_i(z, \mathbf{K})$ and $\mathrm{X}_i(z, \mathbf{K})$, for each i; and a set of value functions, $v_i(z, \mathbf{K}, k_i)$ for each i, such that

1) given the per capita decision rules, the value function for type i, v_i, satisfies equation (53), and h_i and x_i are the associated decision rules; and

2) $\mathrm{H}_i(z, \mathbf{K}) = \mathrm{h}_i(z, \mathbf{K}, K_i)$ and $\mathrm{X}_i(z, \mathbf{K}) = \mathrm{x}_i(z, \mathbf{K}, K_i)$ for each i.

To compute an equilibrium, we approximate the return function of each type by a quadratic function, using the quadratic approximation procedure described in Section 3. To obtain successive approximations of the value function for type i we iterate on the following mapping, where y_i, is the stacked vector $(z, \mathbf{K}, k_i, \mathbf{X}, \mathbf{H}, x_i, h_i)$:

$$v_i^{n+1}(z, \mathbf{K}, k_i) = \max\{y_i^T Q_i y_i + \beta v_i^n(z', \mathbf{K}', k_i')\}$$

$$\text{s.t.} \quad z' = A(z), \ (54), \ (55),$$

$$X_j = \mathrm{X}_j^n(z, \mathbf{K}) \text{ and } H_j = H_j^n(z, \mathbf{K}) \text{ for } j = 1, \ldots, N. \tag{57}$$

To compute v_i^{n+1}, X_i^n, and H_i^n for a household of type i given a quadratic function v_i^n, we first employ the substitution procedure described in Section 3 to eliminate z', $\mathbf{K}'$, and k_i', using the linear equations $z' = A(z)$, (55) and (54), respectively. This must be done for each i.

Next, for each type i, we take the first-order conditions with respect to x_i and h_i. Substituting the aggregate consistency conditions—$x_i = X_i$, $h_i = H_i$, and $k_i = K_i$ for each i—into these first order conditions, we obtain $2N$ equations that are used to solve for the $2N$ unknowns $\mathbf{X}$ and $\mathbf{H}$ as functions of z and $\mathbf{K}$. These are the functions X_j^n and H_j^n, for $j = 1, \ldots, N$, that appear in the mapping (57).

The third step is to use the first-order conditions to solve for x_i and h_i as a function of $(z, \mathbf{K}, k_i, \mathbf{X}, \mathbf{H})$ for each i. The resulting functions are used to eliminate x_i and h_i from the right side of the mapping (57). The final step is to eliminate $\mathbf{X}$ and

$\mathbf{H}$ by using the linear functions, X_j^n and H_j^n, for each j. From this, we obtain a quadratic function of z, $\mathbf{K}$, and k_i, which is used as the next approximation of v_i, v_i^{n+1}. The procedure is repeated until the iterations have converged.

Notes

We are grateful to Mariano Cortes and Rajeev Dhawan for research assistance. This research was supported in part by NSF Grants SES-8921346 and SES-6205417.

1. This repeats some of the material covered in Chapter 1, but this seemed to us to be useful for carrying out the purposes of the current chapter.

2. See Stokey and Lucas with Prescott (1989) for details on the dynamic general equilibrium theory underlying the approach employed in this section. It is worth noting that in some special cases a similar approach can be used when equilibrium allocations are not Pareto optimal. In these cases, equilibrium allocations are the solution to *some* dynamic optimization problem that is different from the problem that gives the optimal allocation. See Becker (1985) for an example where this approach is employed.

3. Distinguishing between z and s is not important in cases where a social-planning problem is to be solved. However, it is important in cases where the Second Welfare Theorem does not hold. We make this distinction here so that the notation will be consistent throughout the chapter.

4. Although it is clear that (2) allows for the possibility that components of z evolve as a continuous-state Markov process, it is not difficult to modify the solution method described in this section to allow for components of z to follow a finite-state Markov chain. In this case, instead of solving for a single value function, $v(z, s)$, there is a separate value function, $v_z(s)$, for each z in the state space.

5. Examples from this literature include Cho and Rogerson (1988), Christiano (1988), Greenwood, Hercowitz, and Huffman (1988), Kydland (1984a), Kydland and Prescott (1982, 1988), Hansen (1985), Hansen and Sargent (1988), and King, Plosser, and Rebelo (1988a). In each of these papers, numerical methods are used to solve a planning problem, such as (2.1). Long and Plosser (1983) consider an example where the planning problem can be solved analytically so that numerical methods are not required.

6. In this example we show how to add deterministic growth to the model. It is also possible to introduce stochastic growth by assuming that the technology shock evolves as a random walk with drift. The details are given in Hansen (1989).

7. We have imposed the result that with utility separable in consumption and leisure, optimal consumption is the same for those who work and for those who do not work.

8. Notice that although n, and not h, is a decision variable for an individual household, the social planner does choose h.

9. In practice, we recommend choosing $\bar{h}$ so that the steady state computed from the linear decision rules are the same (up to, say, six decimal places) as the steady state for the nonlinear economy. In addition, if the steady states for the components of y differ significantly in absolute value, it may be desirable to set h_i^i proportional to the steady state, $h_i^i = \bar{h}\bar{y}_i$, as long as $\bar{y}_i$ is different from zero.

10. In the rest of this section, unless we say otherwise, references to v are to the optimal value function for this linear-quadratic problem, as opposed to the optimal value function for problem (1).

11. The return function, r, given that the utility function is strictly concave, is bounded from above. See Stokey, Lucas, and Prescott (1989) for a discussion of discounted dynamic programming with returns bounded from above.

12. In practice, one could just as well eliminate this last row and column. However, we have chosen to fill this last row and column with zeros to simplify notation.

13. In practice, one should ignore the constant term of v^{n+1} and v^n (the (1,1) element) when doing this comparison. The reason is that this term takes relatively longer to converge and has no effect on the policy functions.

14. Here, and in the rest of the paper, we use lower-case letters (e.g., h and k) to denote quantities associated with a particular household. Capital letters (H and K) denote economy-wide (per capita) quantities that are determined in equilibrium but are not influenced by the actions of any individual household.

15. The notion of a recursive competitive equilibrium is developed in Prescott and Mehra (1980).

16. A natural extension of this example would be to model the tax rates as exogenous stochastic processes, or as depending on the endogenous state variables. See Braun (1990), Chang (1990), Greenwood and Huffman (1991), and McGrattan (1989) for applications of this sort to equilibrium business cycle theory.

17. See Sargent (1987) for a detailed discussion of this issue.

18. Both of these types of models are discussed in Sargent's (1987) textbook, and standard references are provided. Papers that contain applications of these monetary models to equilibrium business cycle theory include Cooley and Hansen (1989), Huh (1993), and Kydland (1989).

19. Ríos-Rull (1992a) extends and applies these methods to the study of models with n–period lived overlapping generations.

20. Notice that we have switched notation from previous sections. Previously, investment was denoted by i_t, but the letter i is now used to index type of household. Therefore, we now use x_t to denote investment.