Linear Quadratic Approximations: An Introduction*

Javier Díaz-Giménez

Published as Chapter 2 of *Computational Methods for the Study of Dynamic Economies.* Ramon Marimon and Andrew Scott (eds). Oxford University Press.

Created: September 5, 1996
Last changes: April 24, 1998
Version: 5.12

*The comments of Andrés M. Alonso, Florin Bidian, Jorge Durán, Luis Puch and Andrew Scott are gratefully acknowledged. Jorge Durán also provided the MATLAB code. Address correspondence about this chapter to: Javier Díaz-Giménez, Departamento de Economía, Universidad Carlos III de Madrid, Calle Madrid 126, 28903 Getafe (Spain). E-mail address: kueli@eco.uc3m.es

1 Introduction

The main purpose of applied economic analysis is to evaluate the welfare consequences of economic policy (see Lucas (1987), Chapter 2). To achieve this ambitious goal, the first step is to construct computable model economies that can be used as laboratories to simulate the effects of economic policies. To capture the main features of the economic decision problems solved by people in the real world, the decision problems solved by the households that live in the model economies must be dynamic and stochastic.

Unfortunately, there are no closed form solutions for the dynamic decision problems that arise once the functional forms for preferences and for the aggregate technology of the model economies are calibrated to mimic key features of the aggregate behavior of real economies. One way to get around this difficulty, is to use numerical methods to compute approximated solutions to those calibrated decision problems.

The linear quadratic approximation (hereinafter referred to as the LQ approximation) is one of those methods. It is an approximation because it computes the solution to a quadratic expansion of the utility function about the steady state or the stable growth path of the model economies. The main purposes of this chapter are to review the theoretical basis for the LQ approximation and to illustrate its use with a detailed example.

The LQ approximation can be used to compute the solutions to many extensions of the neoclassical growth model. Some examples of this literature are the following: Brock and Mirman (1972), Kydland and Prescott (1982, 1988), Kydland (1984), Hansen (1985), Prescott (1986), Cho and Rogerson (1988), Christiano (1988), Greenwood, Hercowitz and Huffman (1988), Hansen and Sargent (1988), and King, Plosser and Rebelo (1988).

Very little of the material included in this chapter is original. Most of what is here can be found in Sargent (1987), Stokey and Lucas with Prescott (1989) and Hansen and Prescott (1995). Essentially this chapter contains a summarized review of dynamic programming as it applies to the calibrated neoclassical growth model, and a description of the LQ approximation to social planner problems. A detailed algorithm that can be used to solve this class of problems is also provided. The algorithm is illustrated with a simple example. The example has been programmed using MATLAB but it can also be easily implemented in any computer language that can handle matrix algebra such as GAUSS, FORTRAN 90 or C.

The rest of this chapter is organized as follows: Section 2 describes the standard neoclassical growth model. Section 3 describes a social planner problem that can be used to solve the model. Section 4 contains a recursive formulation of the planner's problem. Section 5 describes the LQ approximation to this problem. Section 6 offers some concluding comments and, finally, the Appendix contains a MATLAB program that illustrates the computational methods described in this chapter.

2 The basic neoclassical growth model

In this section we describe the basic neoclassical growth model in terms of population, endowments, preferences and technology.

2.1 Population

We assume that the model economy is inhabited by a large number of identical households who live forever.

2.2 Endowments

We assume that the model economy households are endowed with one unit of productive time that they can allocate to work in the market or to other uses. We also assume that the model economy households are endowed with $k_0 > 0$ units of productive capital which depreciates geometrically at a constant rate $0 < \delta < 1$, and which they may augment through investment, i.

2.3 Preferences

We assume that households in this economy order their stochastic processes on consumption and leisure according to

$$E \sum_{t=0}^{\infty} \beta^t U(c_t) \tag{1}$$

where E denotes the expectation operator, $0 < \beta < 1$ is the time discount factor, function U is generally assumed to be twice continuously differentiable, strictly increasing and strictly concave, and c_t denotes period t consumption. This description of household preferences implies that households do not value leisure. Hence, we are abstracting from the choice between labor and leisure. This assumption is customarily made in many exercises in growth theory (see Cooley and Prescott (1995)). We adopt it here because it reduces the household decision problem to choosing between consumption and savings, and therefore it simplifies the notation. The methods described in this chapter can be easily extended to solve model economies that include more than one decision variable, such as those that model the choice between labor and leisure, which is an essential feature in the study of business cycles.

2.4 Technology

We assume that in the model economy there is a freely accessible technology that makes it feasible to transform capital, k, into output, y, and that output can be costlessly transformed into either consumption or investment.[1] We make this assumption for simplicity. If we are to take it literally, we can think of an agricultural economy where the harvest of barley can be costlessly split between barley used for seeding (investment) and barley used for the costless production of beer (consumption).

We assume that the production technology can be described by the following neoclassical production function:

$$c_t + i_t = y_t = F(k_t, z_t) \tag{2}$$

where the production function, F, is generally assumed to be twice continuously differentiable, strictly increasing and strictly concave in k.[2]

Given our assumptions on capital depreciation, the law of motion of the capital stock is the following:

$$k_{t+1} = (1-\delta)k_t + i_t \tag{3}$$

Finally, variable z represents a technology shock. This shock is observed at the beginning of the period and it is assumed to follow a first order linear Markov process that can be described by

$$z_{t+1} = L(z_t) + \varepsilon_{t+1} \tag{4}$$

where function L is assumed to be linear and where the random variables ε_j are assumed to be independent and identically distributed draws from a distribution with zero mean and finite variance.

3 A social planner problem

One way to formalize the model economy described above is to use an appropriate specialization of the language of Arrow and Debreu (see Debreu (1959)) and to define a competitive

[1] Note that this assumption implies that in equilibrium the prices of consumption and investment must be identical.

[2] The standard neoclassical production function is $F(k_t, n_t, z_t)$, where n_t denotes the labor input. Function F is generally assumed to be twice continuously differentiable and strictly concave in both arguments and to display constant returns to scale. In our simplified version of the problem households do not value leisure. Hence, part of their optimal plan is to allocate their entire endowment of productive time to the market. Therefore $n_t^* = 1$ and $F(k_t, n_t, z_t)$ becomes $F(k_t, 1, z_t) = F(k_t, z_t)$.

equilibrium for this economy. Essentially this entails, *i.)* defining the competitive decision problems faced by the households and firms in the economy and, *ii.)* finding the stochastic processes for $(y_t,\ c_t,\ i_t,\ k_t)$, and the corresponding vector of prices, such that, when taking the price vector as given, the processes on quantities solve both the households and the firms decision problems, and that the prices are such that the processes on $(y_t,\ c_t,\ i_t)$ clear the goods market every period.

In general, finding the competitive equilibrium of this class of model economies is a difficult computational problem. Fortunately, for many of these model economies —essentially those that both assume complete markets and exclude externalities and distortionary taxes— we can prove that some version of the Second Welfare Theorem applies. This implies that the competitive equilibrium allocations that solve the model economies are identical to the Pareto optimal allocations that would be chosen by a benevolent social planner whose objective was to maximize the welfare of the representative household. This Welfare Theorem is useful because social planner problems are generally easier to solve than competitive equilibrium problems, since they do not include prices, and therefore their dimension is smaller.[3]

A version of the problem solved by a benevolent social planner of the model economy described in the previous section is the following:

$$\max_{\{c_t, i_t\}_{t=0}^{\infty}} E \sum_{t=0}^{\infty} \beta^t u(c_t) \tag{5}$$

subject to constraints (2), (3) and (4) and with $k_0 > 0$ and $z_0 > 0$ given.

Now that we have a low-dimension characterization of the problem that solves the model economy described in Section 2, our next task is to solve it. But first let us simplify the notation. Let $f(k,z) = F(k,z) + (1-\delta)k$. Then we can substitute constraints (2) and (3) into expression (5) to obtain

$$\max_{\{k_{t+1}\}_{t=0}^{\infty}} E \sum_{t=0}^{\infty} \beta^t U(f(k_t, z_t) - k_{t+1}) \tag{6}$$

subject to expression (4) and with $k_0 > 0$ and $z_0 > 0$ given.

The first order conditions of this problem can be summarized by the following Euler equations that must hold for every $t = 0, 1, 2, \ldots$:

$$U'\left[f(k_t, z_t) - k_{t+1}\right] = \beta E\left\{U'\left[f(k_{t+1}, z_{t+1}) - k_{t+2}\right] f'(k_{t+1}, z_{t+1})\right\} \tag{7}$$

[3]The methods described in this notes have been specifically designed to be applied to social planner problems. For a description of the computational methods that can be used to solve non-optimal economies see Danthine and Donaldson (1995).

This is a second order stochastic difference equation and in most cases it is hard to solve because closed form solutions cannot be found. One way to get around this problem is to use numerical methods to obtain approximate numerical solutions. Since numerical methods are typically intensive in computations, an efficient way to implement these methods is to use a computer. Since, as it stands, the problem described in expression (5) is not suitable to be solved using a computer, our next step is to find an alternative formulation of the social planner problem that can solved thus.

4 A recursive formulation of the social planner problem

Suppose that the problem described in expression (5) had been solved for every possible value of k_0 and z_0. Then we could define a function V such that $V(k_0, z_0)$ would give us the value of the maximized objective function when the initial conditions are (k_0, z_0). Thus defined, function V gives us the value for the household, measured in utils, of following its optimal plan. For obvious reasons we call this function a value function. If function V is such a value function, and if the initial conditions of the planners problem are given by the pair (k_0, z_0), then the definition of V is the following:

$$V(k_0, z_0) \equiv \max_{\{k_{t+1}\}_{t=0}^{\infty}} E \sum_{t=0}^{\infty} \beta^t U\left[f(k_t, z_t) - k_{t+1}\right] \tag{8}$$

subject to expression (4).

Our next task is to transform expression (8) in order to unveil its recursive structure. Moving the first term and β out of the summation sign we obtain:

$$V(k_0, z_0) \equiv \max_{\{k_{t+1}\}_{t=0}^{\infty}} \left\{ U\left[f(k_0, z_0) - k_1\right] + E \sum_{t=1}^{\infty} \beta^t U\left[f(k_t, z_t) - k_{t+1}\right] \right\} =$$

$$= \max_{\{k_{t+1}\}_{t=0}^{\infty}} \left\{ U\left[f(k_0, z_0) - k_1\right] + \beta E \sum_{t=0}^{\infty} \beta^t U\left[f(k_{t+1}, z_{t+1}) - k_{t+2}\right] \right\} \tag{9}$$

Note that the second term of expression (9) is β times the value, as defined in expression (8), of the optimal plan when the initial conditions are given by the pair (k_1, z_1). Formally then we can rewrite expression (9) as follows:

$$V(k_0, z_0) = \max_{k_1} \left\{ U\left[f(k_0, z_0) - k_1\right] + \beta E[V(k_1, z_1)|z_0] \right\}. \tag{10}$$

subject to expression (4).

This formulation of the planner's problem highlights its recursive structure. In every period t, the planner faces the same decision: choosing the following period capital stock, k_{t+1}, that maximizes the current return plus the discounted value of the optimal plan from period $t+1$ onwards. Since this problem repeats itself every period the time subscripts become irrelevant, and we can transform expression (10) into

$$V(k,z) = \max_{k'} \left\{ U\left[f(k,z) - k' \right] + \beta E[V(k',z')|z] \right\}. \tag{11}$$

subject to expression (4).[4]

Expression (11) is a version of Bellman's equation, named after Richard Bellman (1957). Since function V is unknown, expression (11) is a functional equation. A consequence of the recursive structure of problem (11) is that its optimal solutions are self-enforcing. This means that as time advances the planner has no incentive to deviate from the original optimal plans. This property is known as Bellman's principle of optimality and the decision rules that satisfy this property are said to be time consistent.[5]

In general not every maximization problem can be mapped into a recursive problem. A characterization, perhaps not minimal, of problems that can be thus mapped is the following: *i.)* The return function, U, has to be time separable in the contemporaneous state and control variables,[6] and *ii.)* the objective function and the constraints have to be such that current decisions affect current and future returns but not past returns.[7]

If we knew function V, we could use expression (11) to define a function g such that for every value of the current capital stock, k, and the current technology shock, z, function $k' = g(k,z)$ would give us the value of the capital stock one period ahead that attains the maximum of expression (11), and, in the cases when they are equivalent, of expression (5). Since function g gives us the optimal capital accumulation decision we call it the optimal decision rule. Unfortunately we do not know function V —in fact, we are not even sure of its existence— and therefore we cannot compute function g. Richard Bellman (1957) developed the methods of dynamic programming that prove the existence of function V, characterize its properties, and allow us to compute it. A formal description of these methods and their

[4] Henceforth we adopt the dynamic programming convention by which primes denote the forwarded values of variables.

[5] Some examples of policies that are not time consistent have been described by Kydland and Prescott (1977) and by Calvo (1978). They mostly arise when agents are assumed to play differential games (*e.g.* the responses of some policymakers to the taking of hostages. For instance, a policy that calls for negotiations today once the hostages have been taken and that promises never to negotiate again in the future is not time consistent).

[6] In dynamic programming jargon the state variables are the variables that summarize the position of the system before the current period decisions are made and the control variables are those decisions. The possible choices for state and control variables need not be unique, an they can incorporate some redundancies. In our example, the state variables are is k_t and z_t, and the control variable is k_{t+1}. In Section 5 below we change the control variable to i_t.

[7] A formal analysis of the equivalence between the maximization and the recursive problems can be found in Sections 4.1 and 9.1 of Stokey and Lucas with Prescott (1989).

application to economics is beyond the scope of this chapter and it can be found in Sargent (1987) and Stokey and Lucas with Prescott (1989), amongst others.

Heuristically, one of these methods is the following: since we do not know function V, we start out by choosing any differentiable and concave function as our initial candidate. Then we use Bellman's equation to compute the associated decision rule and the iterated value function and we proceed repeatedly until the sequence of functions thus defined converges. In algorithmic language this method amounts to the following:

- Step 1: Guess a differentiable and concave candidate value function, $V_n(k, z)$.
- Step 2: Use Bellman's equation to compute $k' = g_n(k, z)$ and $V_{n+1}(k, z) = T[V_n(k, z)]$, where $T[V_n(k, z)] = \max_{k'} \{U[f(k, z) - k'] + \beta E[V_n(k', z')|z]\}$, subject to expression (4).
- Step 3: If $V_{n+1} = V_n$, we are done. Else, update V_n and goto Step 1.

The previous algorithm returns a sequence of functions defined recursively as follows: $V_1 = T(V_0)$, $V_2 = T(V_1)$, ..., $V_{n+1} = T(V_n)$ and attempts to compute the limit of the sequence, $V^* = \lim_{n\to\infty} V_n$.[8]

Before we discuss the existence of this limit function, V^*, it is important to understand that if did exist, function V^* would be the optimal value function that we are looking for. Suppose by way of contradiction that $V_{n+1} \neq V_n$, that is, that we have not found a fixed point of mapping T, could V_n be the optimal value function? The answer to this question should be "no", and the intuitive reason for this answer can be found in Bellman's principle of optimality. If function V changes, then the optimal decision rule will also change and the optimal decisions will not be self-enforcing. In fact, a necessary and sufficient condition for decisions to be self-enforcing is that the value function does not change. Then and only then, the optimization problem will be always the same, the optimal decisions will also be the same, the households have no incentives to deviate from their original plans, and the solutions to the optimization problems will be self-enforcing.

The formal proof of the existence of an optimal value function, V^*, escapes the scope of is chapter and it can be found in the literature referenced above. Informally the proof shows that Bellman's operator, T, is a contraction in the appropriate metric space, and then it uses the Contraction Mapping Theorem to establish that a unique fixed point, $V^* = T(V^*)$, exists, and that it can be attained recursively using Bellman's Operator starting from any differentiable and concave function, V_0. In the next section we exploit this result to compute the LQ approximation to dynamic programs.

Exercise 1: Consider the deterministic version of the standard neoclassical growth model where $U(c_t) = \log c_t$ and $f(k_t) = k_t^\alpha$. Define Bellman's operator, T, associated to the social planner's

[8]Note that finding the limit of the sequence is equivalent to finding a fixed point of operator T, $V^* = T(V^*)$.

problem for this economy. Assume that $V_0(k) = 0$, *and compute* $V_1(k) = T[V_0(k)]$, $V_2(k) = T[V_1(k)]$, *and the associated decision rules,* $g_0(k)$, $g_1(k)$ *and* $g_2(k)$.

Solution: Bellman's operator, T, *for this problem can be defined as*

$$V_{n+1} = T(V_n) = \max_{k'} \{\log(k_t^\alpha - k_{t+1}) + \beta V_n(k')\}$$

The first three terms in the sequence of functions are:

$$V_0(k) = 0,\ V_1(k) = \alpha \log k \text{ and } V_2(k) = (\alpha + \alpha^2\beta) \log k + (1 + \alpha\beta) \log[\alpha\beta/(1 + \alpha\beta)]$$

The associated decision rules are:

$$g_0(k) = 0,\ g_1(k) = [\alpha\beta/(1 + \alpha\beta)]k^\alpha \text{ and } g_2(k) = [(\alpha + \alpha^2\beta)/(1 + \alpha + \alpha^2\beta)]k^\alpha$$

5 A linear quadratic approximation

As Exercise 1 illustrates, a problem with using Bellman's operator recursively to compute the limit in the sequence of value functions is that this method is very demanding algebraically. An important exception to this rule is the case of quadratic return functions. In the following pages we show that if the return function is quadratic in the state and control variables, and if the constraints are linear in the states, forwarded states and control variables, then the optimal value function is quadratic in the state variables. Moreover, in this case Bellman's operator maps quadratic functions into quadratic functions. Therefore, in the case of linear quadratic problems, if our initial guess is a quadratic function, then Bellman's operator returns another quadratic function, and therefore every function in the recursive sequence is quadratic.

Unfortunately quadratic return functions are not very useful in macroeconomics since they are not supported by the stylized facts that describe the aggregate time series of most of real economies. Instead, evidence from real economies suggests that we should consider CES utility functions. One way to get around this problem is to use Taylor's expansion to approximate the CES functions that we are interested in with a quadratic function. This procedure and the proposition described in the previous paragraph are the gist of the LQ approximation.[9]

The LQ approximation is a reasonable approximation when the following conditions are met: *i.)* the deterministic versions of the model economies converge to a stable steady states, or to stable growth paths,[10] and *ii.)* the local dynamics about the steady state of the real economy in which we are interested are well approximated by a linear law of motion. The reason for this last condition is that, as we have already mentioned, the optimal decision rules obtained using the LQ approximation are linear and, therefore, this method should not be used when there is evidence of non-linearities in the real economy data. The real business

[9]Note that when we use Taylor's expansion the original function and the approximated function take the same value typically in only one point (the point about which the original function is being "expanded"). In most cases, therefore, Taylor's expansion is a good approximation only in a small neighborhood of that point.

[10]Note that these steady states or stable growth paths are the points about which the original return functions are expanded.

cycle literature is full of examples where the LQ approximation has been successfully used. Some specific examples of economies that can be solved using this method are those described in the papers listed in the introduction to this chapter.

5.1 A general solution algorithm

We now generalize the problem described in expression (11) to the case of multiple state and control variables. A general notation for this extended class of problems is the following:

$$V(z,s) = \max_{d} \left\{ r(z,s,d) + \beta E \left[V(z',s') \mid z \right] \right\} \quad s.t. \tag{12}$$

$$s' = A(z,s,d) \tag{13}$$

$$z' = L(z) + \varepsilon' \tag{14}$$

where z is now a vector of n_z exogenous state variables, s is a vector of n_s endogenous state variables, d is a vector of n_d control variables, ε is a vector of n_ε random variables with zero mean and finite variance, function r is the return function, and functions A and L are linear.[11]

It is important to note that the LQ approximation can only be used in those problems that can be mapped into a structure similar to the one described in expressions (12)–(14). Therefore every non-linear constraint in the original problem must be first substituted into the original return function. This restriction limits the class of problems that can be solved using the LQ approximation to those that at most have as many non-linear constraints as there are arguments in the utility function.

Exercise 2: Consider a stochastic version of the standard neoclassical growth model where $U(c_t) = \log c_t$, $F(k_t, z_t) = e^{z_t} k_t^{\alpha}$, $z_{t+1} = \rho z_t + \varepsilon_{t+1}$, $\varepsilon_t \sim N(0, \sigma_\varepsilon^2)$, $0 < \alpha < 1$, $0 < \rho < 1$ *and, a.) write the Bellman equation associated to the planner problem, and b.) substitute the non-linear constraints into the utility function and identify the vector of exogenous states,* z, *the vector of endogenous states,* s, *the vector of controls,* d, *the return function,* r, *and linear functions* A *and* L *that map the example that we have discussed in the previous section into the generalized notation described above.*

Solution: The Bellman equation associated to the planner's problem is

$$V(k,z) = \max_{c,k'} \{\log c + \beta E[V(k',z')|z]\}$$

$s.t.$

[11] Note that henceforth, for reasons that will become clear later, the precise ordering of variables z, s and d is crucially important. The exogenous state variables, z, must come first, the endogenous state variables, s, must come next, and the decision variables, d, must come last.

$$c + i = e^z k^\alpha$$
$$k' = (1-\delta)k + i$$
$$z' = \rho z + \varepsilon'$$

After substituting the non-linear constraint into the utility function the Bellman equation becomes

$$V(k,z) = \max_i \{\log(e^z k^\alpha - i) + \beta E[V(k',z')|z]\}$$
s.t.
$$k' = (1-\delta)k + i$$
$$z' = \rho z + \varepsilon'$$
where $z = z$, $s = k$, $d = i$, $r(z,k,i) = \log(e^z k^\alpha - i$, $A(z,k,i) = (1-\delta)k + i$ *and* $L(z) = \rho z$.

A solution algorithm to solve dynamic programs of the class described in expressions (12)–(14) is the following:

- **Step 1:** Choose a point about which to expand the return function. In most cases this point is the steady state of the deterministic version of the model economy, $(\bar{z}, \bar{s}, \bar{d})$, that we obtain when we substitute the random variables with their unconditional means.[12]
- **Step 2:** Construct a quadratic approximation of $r(z,s,d)$ about $(\bar{z}, \bar{s}, \bar{d})$.
- **Step 3:** Compute the optimal value function $V^*(z,s)$ by successive iterations on Bellman's operator $V_{n+1}(z,s) = T[V_n(z,s)] = \max_d \{r(z,s,d) + \beta E[V(z',s')|z]\}$ subject to expressions (13) and (14).

In the subsections that follow we discuss these steps in greater detail.

5.2 Step 1: Computing the steady state

To compute the steady states of the model economies first we substitute the shocks with their unconditional means and we obtain the first order conditions of the resulting deterministic versions of social planner problems. Then we impose the appropriate steady state conditions. Finally we calculate the steady state values for both the endogenous and the exogenous state variables and for the control variables, $(\bar{z}, \bar{s}, \bar{d})$. In the cases of model economies that converge to stable growth paths we first transform the problem so that its solution is stationary over time.[13] Exercise 3 illustrates this procedure.

Exercise 3: Consider the stochastic version of the standard neoclassical growth model described in Exercise 2 and compute the steady state of the deterministic version of this economy as a function of parameters $(\alpha, \beta, \delta, \rho, \sigma_\varepsilon^2)$.

[12] Henceforth the symbol "$^-$" above a variable denotes the steady state value of that variable.
[13] For an example where this transformation is carried out explicitly see Hansen and Prescott (1995).

Solution: To compute the steady state, it is easier to consider the maximization problem, which in this case is the following:

$\max_{\{c_t,k_{t+1}\}} E\sum_{t=0}^{\infty}\beta^t \log c_t$
s.t.
$c_t + i_t = e_t^z k_t^\alpha$
$k_{t+1} = (1-\delta)k_t + i_t$
$z_{t+1} = \rho z_t + \varepsilon_{t+1}$

Substituting the shocks by their unconditional means and dropping the expectations operator this problem becomes,

$\max_{c_t,k_{t+1}} \sum_{t=0}^{\infty}\beta^t \log c_t$
s.t.
$c_t + i_t = k_t^\alpha$
$k_{t+1} = (1-\delta)k_t + i_t$

Computing the first order conditions for this problem and substituting out the Lagrange multipliers we obtain the following Euler equation:

$c_{t+1} = \beta c_t[\alpha k_{t+1}^{\alpha-1} + (1-\delta)]$

Imposing the steady state condition, $c_t = c_{t+1} = \bar{c}$, *this expression becomes,*

$1 = \beta[\alpha\bar{k}^{\alpha-1} + (1-\delta)]$

Finally, after solving for $\bar{k}$, *we obtain the steady state,* $(\bar{z},\bar{k},\bar{i})$*:*

$\bar{z} = 0$, $\bar{k} = [\alpha\beta/(1-\beta+\beta\delta)]^{[1/(1-\alpha)]}$, $\bar{i} = \delta\bar{k}$.

5.3 Step 2: Constructing the quadratic approximation

The next step is to approximate the return function, $r(z,s,d)$, about the model economy's steady state, $(\bar{z},\bar{s},\bar{d})$. In the remainder of the chapter we switch to matrix notation for its conciseness and because it is particularly suitable for implementation using a computer. In matrix notation, the second order Taylor approximation of function r is:[14]

$$r(z,s,d) \simeq \bar{R} + (W-\bar{W})^T\bar{J} + \frac{1}{2}(W-\bar{W})^T\bar{H}(W-\bar{W}) \tag{15}$$

where number $\bar{R}$ is function r evaluated at the model economy's steady state, $\bar{R} = r(\bar{z},\bar{s},\bar{d})$, column vector W is the vector of ordered state and control variables, $W = [z\ \ s\ \ d]^T$, column vector $\bar{W}$ is vector W evaluated at the steady state, $\bar{W} = [\bar{z}\ \ \bar{s}\ \ \bar{d}]^T$, column vector $\bar{J}$ is the Jacobian vector evaluated at the steady state, $\bar{J} = [\bar{J}_z\ \ \bar{J}_s\ \ \bar{J}_d]^T$, and matrix $\bar{H}$ is the Hessian

[14]Henceforth the superscript, ^{T}, indicates the transpose of a matrix.

matrix evaluated at the steady state,

$$\bar{H} = \begin{bmatrix} \bar{H}_{zz} & \bar{H}_{zs} & \bar{H}_{zd} \\ \bar{H}_{sz} & \bar{H}_{ss} & \bar{H}_{sd} \\ \bar{H}_{dz} & \bar{H}_{ds} & \bar{H}_{dd} \end{bmatrix}$$

Multiplying out expression (15), and grouping the independent terms, the linear terms and the quadratic terms, we obtain

$$r(z,s,d) \simeq (\bar{R} - \bar{W}^T\bar{J} + \frac{1}{2}\bar{W}^T\bar{H}\bar{W}) + W^T(\bar{J} - \bar{H}\bar{W}) + \frac{1}{2}W^T\bar{H}W \tag{16}$$

Since expression (16) is a quadratic form, we can rewrite it as:

$$r(z,s,d) \simeq \begin{bmatrix} 1 & W^T \end{bmatrix} \begin{bmatrix} Q_{11} & Q_{12}^T \\ Q_{12} & Q_{22} \end{bmatrix} \begin{bmatrix} 1 \\ W \end{bmatrix} \tag{17}$$

where $Q_{11} = \bar{R} - \bar{W}^T\bar{J} + \frac{1}{2}\bar{W}^T\bar{H}\bar{W}$, $Q_{12} = \frac{1}{2}(\bar{J} - \bar{H}\bar{W})$, and $Q_{22} = \frac{1}{2}\bar{H}$ and where the role played by the 1's is to select the constant term of the quadratic expression. As we illustrate in the Appendix, this partitioning of matrix Q simplifies the inputing of the data.

In a more compact notation, expression (17) becomes

$$r(z,s,d) \simeq \begin{bmatrix} 1 & W^T \end{bmatrix} Q \begin{bmatrix} 1 \\ W \end{bmatrix} \tag{18}$$

where matrix Q is a square symmetric matrix of dimension $(1 + n_z + n_s + n_d)$.

Exercise 4: Consider the model economy described in Exercise 2. Obtain a quadratic expansion of the return function about the model economy's steady state. Write out the expansion in matrix notation.

Solution: See the Appendix

5.4 Step 3: Computing the optimal value function

The next step is to compute the optimal value function V^* by repeated iterations on Bellman's operator, $V_{n+1}(z,s) = T[V_n(z,s)]$. Using matrix notation the definition of Bellman's operator is the following:

$$V_{n+1}(z,s) = T[V_n(z,s)] = \max_d \left\{ \begin{bmatrix} 1 & W^T \end{bmatrix} Q \begin{bmatrix} 1 \\ W \end{bmatrix} + \beta E\left[V_n(z',s') \mid z\right] \right\} \tag{19}$$

subject to constraints (13) and (14).

The initial guess for the value function, V_n, can be any quadratic and concave function $V_n(s,z) = F^T P_n F$, where ordered column vector F is $F = [1 \;\; z \;\; s]^T$, and where matrix P_n is any symmetric and negative semi-definite matrix of dimension $(1 + n_z + n_s)$.[15]

Substituting this expression for V_n into expression (19), we obtain:

$$V_{n+1}(z,s) = \max_{d} \left\{ \begin{bmatrix} 1 & W^T \end{bmatrix} Q \begin{bmatrix} 1 \\ W \end{bmatrix} + \beta E\left[(F')^T P_n F' \mid z \right] \right\} \tag{20}$$

subject to constraints (13) and (14).

Next we must compute the vector of decision rules, d_n, associated to V_n, and the next element in the sequence of value functions, V_{n+1}. To do this we start out by taking care of the expectation operator, E.

In general operating with expectations is hard, but in the case of linear quadratic problems the Certainty Equivalence Principle simplifies matters considerably. Linear quadratic problems have the following two features: *i.)* the covariance matrix of the of the vector of random variables, Σ, only shows up in the independent terms of the optimal value functions, V^*, and *ii.)* the optimal decision rules, d^*, are independent of this matrix.[16] These two features of linear quadratic problems are known as the Certainty Equivalence Principle.[17]

Returning to our problem, the Certainty Equivalence Principle allows us to assume, without loss of generality, that $\Sigma = 0$. Therefore we can drop the expectations operator by simply replacing the vector of random variables, ε, by its unconditional mean, $\mu(\varepsilon) = 0$. Once we carry out this substitution, expression (20) and constraints (13) and (14) become

$$V_{n+1}(z,s) = \max_{d} \left\{ \begin{bmatrix} 1 & W^T \end{bmatrix} Q \begin{bmatrix} 1 \\ W \end{bmatrix} + \beta (F')^T P_n F' \right\} \tag{21}$$

subject to $s' = A(z,s,d)$ and $z' = L(z)$.

The next step is to transform expression (21) into a quadratic form in $[1 \;\; W^T]$. To do this we use constraints $s' = A(z,s,d)$ and $z' = L(z)$ to substitute the forwarded values of the states out of expression (21). Specifically, we have to find a rectangular matrix B of dimensions $[(1 + n_z + n_s) \times (1 + n_z + n_s + n_d)]$ that satisfies $F' = B[1 \;\; W^T]^T$. In some cases

[15] Any square matrix of the appropriate dimension made up of small negative numbers in the diagonal and zeros elsewhere satisfies these two conditions.

[16] More specifically it can be shown that $V^*(z,s) = F^T P^* F + a$ where P^* is a symmetric and negative semi-definite matrix and $a = [\beta/(1-\beta)] tr P^* \Sigma$ ("tr" here denotes the trace of a matrix), and that $d^* = (J^*)^T F$ where J^* is a vector of constants which is independent of Σ.

[17] Note that the Certainty Equivalence Principle only holds when the objective function is quadratic and the constraints are linear. It does not characterize stochastic control problems in general.

this transformation might require some ingenuity.[18]

Exercise 5: Find matrix B for the model economy described in Exercise 2.

Solution:

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \rho & 0 & 0 \\ 0 & 0 & 1-\delta & 1 \end{bmatrix}$$

Once we have found matrix B, we substitute $F' = B[1 \;\; W^T]^T$ into expression (21) and we obtain

$$V_{n+1}(z,s) = \max_d \left\{ \begin{bmatrix} 1 & W^T \end{bmatrix} Q \begin{bmatrix} 1 \\ W \end{bmatrix} + \beta \begin{bmatrix} 1 & W^T \end{bmatrix} B^T P_n B \begin{bmatrix} 1 \\ W \end{bmatrix} \right\} \tag{22}$$

Note that henceforth we no longer need to consider any constraints because we have substituted them into the utility function.

Using the distributive law of matrix sums and products expression (22) becomes

$$V_{n+1}(z,s) = \max_d \left\{ \begin{bmatrix} 1 & W^T \end{bmatrix} [Q + \beta B^T P_n B] \begin{bmatrix} 1 \\ W \end{bmatrix} \right\} \tag{23}$$

where matrix $[Q + \beta B^T P_n B]$ is a square, symmetric and negative semi-definite matrix and, therefore, the expression inside the curly brackets is indeed quadratic in z, s and d.

The next step is to differentiate expression (23) to obtain the vector of decision rules, $d_n(z,s)$.[19] To simplify the algebra involved in differentiating expression (23), let $M_n = B^T P_n B$ and partition matrices Q and M_n so as to separate the state variables from the control variables.[20] Specifically, let

$$Q = \begin{bmatrix} Q_{FF} & Q_{Fd}^T \\ Q_{Fd} & Q_{dd} \end{bmatrix} \text{ and } M = \begin{bmatrix} M_{FF} & M_{Fd}^T \\ M_{Fd} & M_{dd} \end{bmatrix},$$

where matrices Q_{FF} and M_{FF} are symmetric matrices of dimension $(1+n_z+n_s)$, matrices Q_{dd} and M_{dd} are symmetric matrices of dimension (n_d), and matrices Q_{Fd} and M_{Fd} are rectangular matrices of dimension $[n_d \times (1+n_z+n_s)]$.[21]

[18] Note that to perform this transformation constraints $s' = A(z,s,d)$ and $z' = L(z)$ must be linear in (z,s,d).

[19] Note that since expression (23) is quadratic in (z,s,d) the first order conditions are linear in (z,s).

[20] Note that matrix M_n changes with every iteration, n. To avoid cluttering the notation when we partition matrix M_n we drop subscript n. On the other hand matrix Q remains invariant throughout the iterations.

[21] Note that this partition of matrix Q is completely unrelated to the partition that we have used in expression (17). Once again, the reason that justifies these partitions is to simplify the calculations.

Substituting the partitioned matrices into expression (23) we obtain[22]

$$V_{n+1}(z,s)=\max_{d}\left\{\left[\begin{array}{cc} F^T & d^T \end{array}\right]\left[\begin{array}{cc} Q_{FF}+\beta M_{FF} & Q_{Fd}^T+\beta M_{Fd}^T \\ Q_{Fd}+\beta M_{Fd} & Q_{dd}+\beta M_{dd} \end{array}\right]\left[\begin{array}{c} F \\ d \end{array}\right]\right\} \tag{24}$$

Expression (24) is a quadratic form in $[F^T \;\; d^T]$, and hence it is equivalent to

$$V_{n+1}(z,s)=\max_{d}\left\{F^T[Q_{FF}+\beta M_{FF}]F+2d^T[Q_{Fd}+\beta M_{Fd}]F+d^T[Q_{dd}+\beta M_{dd}]d\right\} \tag{25}$$

Differentiating expression (25) with respect to d^T and equating to zero we obtain the following first order conditions:

$$2[Q_{Fd}+\beta M_{Fd}]F+2[Q_{dd}+\beta M_{dd}]d=0 \tag{26}$$

from which we can readily compute the vector of linear decision rules, $d_n(z,s)$, associated to value function $V_n(z,s)$ to be[23]

$$d_n(z,s)=-(Q_{dd}+\beta M_{dd})^{-1}(Q_{Fd}+\beta M_{Fd})F=J_n^T F. \tag{27}$$

where vector J_n is the vector of the coefficients of the decision rules associated with value function, $V_n(z,s)$. Specifically

$$J_n^T=-(Q_{dd}+\beta M_{dd})^{-1}(Q_{Fd}+\beta M_{Fd}) \tag{28}$$

Substituting expression (27) into (25), we can drop the maximization operator and we obtain

$$V_{n+1}(z,s)=$$

$$F^T\left[Q_{FF}+\beta M_{FF}-(Q_{Fd}+\beta M_{Fd})^T(Q_{dd}+\beta M_{dd})^{-1}(Q_{Fd}+\beta M_{Fd})\right]F \tag{29}$$

which is, indeed, the quadratic expression that we are looking for, $V_{n+1}=F^T P_{n+1}F$. In this expression matrix P_{n+1} is the symmetric matrix that we use to define the iterated value function $V_{n+1}(z,s)$. Specifically

$$P_{n+1}=Q_{FF}+\beta M_{FF}-(Q_{Fd}+\beta M_{Fd})^T(Q_{dd}+\beta M_{dd})^{-1}(Q_{Fd}+\beta M_{Fd}) \tag{30}$$

[22]Note that $[1 \;\; W^T]=[1 \;\; z \;\; s \;\; d]=[F^T \;\; d]$.

[23]Note that the concavity of V_n guarantees that the first order necessary conditions described in expression (26) are also sufficient.

The last step in this iterative procedure is to compare P_n and P_{n+1}. If it turns out that $P_n = P_{n+1} = P^*$, then it immediately follows that $V_n = V_{n+1} = V^*$ and we have found the optimal value function. If, on the other hand, $P_n \neq P_{n+1}$, then we update P_n and we iterate once again until convergence.[24]

Exercise 6: Let $\alpha = 0.33$, $\beta = 0.96$, $\delta = 0.1$ and $\rho = 0.95$ and compute the optimal value function, V^*, and the associated optimal decision rule, d^*, for the LQ approximation to the social planner's problem of the model economy described in Exercise 2

Solution: see the Appendix.

6 Conclusion

In this chapter we have described a LQ approximation to social planner problems. Our method is based on the following two basic results of dynamic programming: *i.)* that in LQ problems Bellman's operator maps quadratic functions into quadratic functions, and *ii.)* that the Contraction Mapping Theorem guarantees the convergence of the sequence of value functions obtained using Bellman's operator recursively. We have used algorithmic language and matrix notation to simplify the implementation of this method using a computer. As we have already mentioned, the LQ approximation has been successfully used in many exercises in real business cycle theory. Since the proof of a pie is to eat it, to prove the understanding of the subject, we provide the reader with one last exercise taken from that literature.

Exercise 7: Consider the basic Kydland and Prescott divisible labor calibrated model economy described in Hansen (1985). a.) Use the LQ approximation to compute the optimal value function and the optimal decision rules associated to the social planner problem of that economy, and b.) use the decision rules to reproduce the results reported in the third and fourth columns of Table 1 of that paper.

Solution: Sorry, none is provided. This time you are on your own.

[24] In the case of linear quadratic programs, the Contraction Mapping Theorem guarantees that $\lim_{n\to\infty} P_n = P^*$, which of course, is equivalent to $\lim_{n\to\infty} V_n = V^*$.

7 References

Bellman, R. 1957. *Dynamic Programming.* Princeton, N.J.: Princeton Universtiy Press.

Brock, W. A. and L. J. Mirman. 1972. "Optimal Economic Growth and Uncertainty: The Discounted Case." *Journal of Economic Theory* 4:497–513.

Calvo, G. A. 1978. "On the time consistency of optimal policy in a monetary economy." *Econometrica* 46:1411–28.

Cho, J. O. and R. Rogerson. 1988. "Family labor supply and aggregate fluctuations." *Journal of Monetary Economics* 21:233–46.

Christiano, L. J. 1988. "Why does inventory investment fluctuate so much?." *Journal of Monetary Economics* 21:247–80.

Cooley and Prescott. 1995. "Economic growth and business cycles." In Cooley, T. F., ed. *Frontiers of Business Cycle Research*, Princeton University Press.

Danthine and Donaldson. 1995. "Computing equilibria of non-optimal economies." In Cooley, T. F., ed. *Frontiers of Business Cycle Research*, Princeton University Press.

Debreu, G. 1959. *The Theory of Value.* New Haven, Conn.: Yale University Press.

Greenwood J., Z. Hercowitz and G. W. Huffman. 1988. "Investment, capacity utilization and the real business cycle." *American Economic Review* 78:402–17.

Hansen, G. 1985. "Indivisible labor and the business cycle." *Journal of Monetary Economics* 16:309–27.

Hansen, G. and E. C. Prescott. 1995. "Recursive methods for computing equilibria of business cycle models." In Cooley, T. F., ed. *Frontiers of Business Cycle Research*, Princeton University Press.

Hansen, G. and T. J. Sargent. 1988. "Straight time and overtime in equilibrium." *Journal of Monetary Economics* 21:281–308.

King, R. G., C. I. Plosser and S. T. Rebelo. 1988. "Production, growth and business cycles. I. The basic neoclassical model." *Journal of Monetary Economics* 21:195–232.

Kydland, F. E. 1984. "Labor force heterogeneity and the business cycle." *Carnegie-Rochester Conference Series on Public Policy* 21:173–208.

Kydland, F. E. and E. C. Prescott. 1977. "Rules rather than discretion: the inconsistency of optimal plans". *Journal of Political Economy* 85:473–91.

Kydland, F. E. and E. C. Prescott. 1982. "Time to build and aggregate fluctuations." *Econometrica* 50:1345-70.

Kydland, F. E. and E. C. Prescott. 1988. "The workweek of capital and its cyclical implications." *Journal of Monetary Economics* 21:343–60.

Lucas, R. E. 1987. *Models of Business Cycles.* New York: Basil Blackwell.

Prescott, E. C. 1986. "Theory ahead of business cycle measurement". *Federal Reserve Bank* *of Minneapolis Quarterly Review* (Fall): 9–22.

Sargent, T.J, 1987. *Dynamic Macroeconomic Theory*. Cambridge, MA: Harvard University Press. Chapter 1 and Appendix.

Stokey, N. L. and R.E. Lucas with E. C. Prescott. 1989. *Recursive Methods in Economic Dynamics.* Cambridge, MA: Harvard University Press. Specially Chapter 2.

Appendix: A MATLAB program that illustrates the LQ method

```
% This program computes the value function and the optimal
% decision rules of the linear-quadratic approximation to the
% problem described in Exercise 6.

% Author:  Jorge Duran
% e-mail:  xurxo@eco.uc3m.es

% This message is for the loop below.

disp(' The program is iterating on Bellman's equation.  Please wait...')

% Step 0:  Input the parameter values

a = 0.33; % alpha
b = 0.96; % beta
p = 0.95; % rho
d = 0.10; % delta
e = exp(1); % Number e (intrascendent but necessary)

% Step 1:  Compute the steady state

K = ((a*b)/(1-b*(1-d)))^(1/(1-a));
X = d*K;
Z = 0;

% Step 2:  Construct the quadratic expansion of the utility function

% Step 2.1:  Evaluate all the first and second order derivatives
% of the utility function at the steady state:

R = log(e^Z*K^a - X);

Jz = (e^Z*K^a) / (e^Z*K^a - X);
Jk = (e^Z*a*K^(a-1)) / (e^Z*K^a - X);
Jx = (-1) / (e^Z*K^a - X);

Hzz = (((e^Z*K^a) * (e^Z*K^a - X)) - (e^Z*K^a)^2) / ((e^Z*K^a - X)^2);
Hkk = (((e^Z*a*(a-1)*K^(a-2))*(e^Z*K^a - X)) - ...
(e^Z*a*K^(a-1))^2 )/((e^Z* K^a - X)^2);
Hxx = (-1)/((e^Z*K^a - X)^2);
Hzk = (((e^Z*a*K^(a-1))*(e^Z*K^a - X)) - (e^Z*K^a*e^Z*a*K^(a-1)))/ ...
((e^Z*K^a - X)^2);
Hzx = (e^Z*K^a) / ((e^Z*K^a - X)^2);
Hkx = (e^Z*a*K^(a-1))/((e^Z*K^a - X)^2);
```

```
% Step 2.2:  Define the Jacobian vector and the Hessian matrix evaluated
% at the steady state

DJ = [Jz Jk Jx ]';
D2H = [Hzz Hzk Hzx
Hzk Hkk Hkx
Hzx Hkx Hxx];

% Step 2.3:  Define matrix Q

W = [Z K X]';

Q11 = R - W'*DJ + 0.5*W'*D2H*W;
Q12 = 0.5*(DJ-D2H*W);
Q22 = 0.5*D2H;

Q=[ Q11 Q12'
Q12 Q22];

% Step 3:  Compute the optimal value function.

% Step 3.1:  Partition matrix Q to separate the state and control variables

Qff = Q(1:3,1:3);
Qfd = Q(4,1:3);
Qdd = Q(4,4);

% Step 3.2:  Input matrix B

B = [ 1 0 0 0
0 p 0 0
0 0 1-d 1];

% Step 3.3:  Input matrix P0

P = [-0.1 0 0
0 -0.1 0
0 0 -0.1 ];

% Step 3.4:  Initialize auxiliary matrix A

A=ones(3);

% Step 3.5:  Iterate on Bellman's equation until convergence

while (norm(A-P)/norm(A))>0.0000001
A = P;
M=B'*P*B;
```

```
Mff = M(1:3,1:3);
Mfd = M(4,1:3);
Mdd = M(4,4);
P=Qff + (b*Mff) - (Qfd'+(b*Mfd)')*inv(Qdd +(b*Mdd))*(Qfd+(b*Mfd));
end

% Step 4:  Output the results

disp(' ')
disp(' The linear policy for investment is d*=J'F, where vector J is:')
J=-(inv(Qdd+(b*Mdd))*(Qfd+(b*Mfd)))';
J
disp(' ')
disp(' The optimal value function is V*=F'PF, where matrix P is:')
P

% END OF MATLAB FILE
```

The output of this program should be the following:

```
J =

0.4983
0.8607
-0.0411

P =

-0.4025 8.0839 0.7369
8.0839 1.0029 -0.1915
0.7369 -0.1915 -0.0819
```