

Discussion Paper 49

Institute for Empirical Macroeconomics  
Federal Reserve Bank of Minneapolis  
250 Marquette Avenue  
Minneapolis, Minnesota 55480

August 1991

MINIMUM WEIGHTED RESIDUAL METHODS  
FOR SOLVING AGGREGATE GROWTH MODELS

Kenneth L. Judd\*

Hoover Institution

and National Bureau of Economic Research

\*The author gratefully acknowledges the comments of seminar participants at Northwestern University and Stanford University, and the research assistance of Pamela Chang.

This material is based on work supported by the National Science Foundation under Grant. No. SES-8722451. The Government has certain rights to this material.

Any opinions, findings, conclusions, or recommendations expressed herein are those of the author and not necessarily those of the National Science Foundation, the University of Minnesota, the Federal Reserve Bank of Minneapolis, or the Federal Reserve System.

# Minimum Weighted Residual Methods for Solving Dynamic Economic Models

## 1. Introduction

Economists are increasingly turning to numerical techniques for analyzing dynamic economic models. While numerical techniques have been used in dynamic economic analysis since Gustafson (1958), the set of problems analyzed with numerical methods has expanded, and there has recently been greater attention paid to developing and evaluating algorithms (see, for example, Taylor and Uhlig (1990)). While progress has been substantial, the numerical techniques have tended to be, or at least have appeared to be, problem specific. Also, economists have generally ignored standard numerical methods in favor of idiosyncratic and *ad hoc* methods.

This paper represents a general approach to the numerical solution of dynamic economic problems. More specifically, we will discuss a class of techniques, called *minimum weighted residual* (MWR) methods, which have been developed in the mathematical literature on numerical solutions to partial differential equations, but appear to be generally applicable to economic problems. For many continuous-time economic problems, partial differential equation formulations of equilibria are immediate and natural, and we could directly use these techniques. For some economic applications, a discrete-time formulation of the problem may be preferred. We will show that the basic ideas of MWR techniques can be adapted for discrete-time applications. In fact, we will point out that many of the techniques which are in current use have substantial similarities once these common features are highlighted. Moreover, once we take this more general approach and use standard tools from numerical analysis, it becomes clear why some methods outperform others, and how to devise efficient techniques.

The basic idea of MWR techniques is to first express equilibrium as a solution to some operator equation. That operator equation can be an ordinary differential equation, as in optimal control problems, a partial differential equation, as in continuous-time dynamic programming, or a functional equation, as in Euler equations expressing necessary conditions for recursive equilibria (Prescott and Mehra, 1980, Stokey and Lucas, 1989). Along with the

operator equation, a problem specifies a topological space, most likely an infinite-dimensional functional space, which contains the solution. Of course, space limitations make it impossible for computers to store such spaces. To make the problem tractable, MWR methods focus on a finite-dimensional subspace of approximations which is likely to be close to the true solution and representable on a computer. Within this space of candidate solutions, we then find an element which "best fits" the operator equation defining equilibrium.

While the basic idea is natural, there are many details. The key details are specifying the subspace within which we look for an approximate solution, defining what we mean by "best fit," and determining how to find the approximate solution. It is in studying these details that we see that numerical solution of dynamic nonlinear economic problems can be tractable, and that different implementations can vary greatly in performance.

This paper has two purposes. First, it lays out the general MWR approach to solving dynamic economic problems. We will see that many existing techniques are related, and discuss the relative merits of various implementations of the common underlying ideas. Second, we will also demonstrate the power, practicality, and efficiency of particular implementations of the MWR method for solving stylized discrete-time growth models. This is a useful focus since optimal growth problems are of substantial interest, and several other authors have discussed the speed of their proposed solution methods for this problem in Taylor and Uhlig (1990) and the accompanying papers, allowing us to make comparisons. Space limitations require a narrow focus, but from these examples it is clear that this method can be applied to a diverse collection of dynamic economic problems. Judd (1990) shows how to use these methods to solve dynamic programming problems, general dynamic equilibrium with distortions, and game-theoretic dynamic models in both discrete and continuous time.

No reader will be surprised that there are techniques which will solve these problems. The challenge, of course, is to find techniques which can run quickly on computers currently available. There are two basic reasons why we want fast algorithms. First, these algorithms can be used for econometric estimation of the underlying nonlinear models only if they are fast since empirical procedures would require repeated use of these algorithms. For example, using the programs described below one could easily compute maximum likelihood estimates

of simple macroeconomic models. Estimation of more elaborate models is also obviously possible, but with more elaborate models the value of speed increases. Second, the only way to convincingly demonstrate comparative dynamic <sup>1</sup> or comparative static propositions would be to calculate hundreds or thousands of examples.

Many dynamic economic problems, including some of those discussed below, have been solved on supercomputers using methods which are so inefficient that only supercomputers can be used. This mistakenly gives the impression that such computing power is necessary. The techniques discussed below need far less computer power. To emphasize this, I have limited myself to using relatively slow equipment. In examples below, I will give running times on a Compaq 386/20, running under DOS 3.1 and using a Weitek math coprocessor. Occasionally I needed to use a VAX 8650 for extra space. The key fact is that these problems can be solved on computers sitting on the desks of many economists. Furthermore, I use standard software, most of which is available at any serious computing facility or commercially at nominal cost. All the programs used below were written in FORTRAN, the dominant language in numerical analysis.

This paper is not intended to be an exhaustive review of numerical and approximation techniques. There are many other approaches which are also of potential value, including regular and singular perturbation techniques, and finite difference schemes, each of which has its own particular strengths (see Judd (1991)). One reason for examining MWR techniques is that they have the advantage of being strongly related at an intuitive level to standard econometric techniques, hence will appear familiar to economists. For discrete-time problems, it is the most natural approach. These methods are also gaining in popularity over finite-difference methods for continuous-time problems. Ultimately, one expects that the full range of nonlinear approximation techniques will be valuable to economists with the best algorithms combining techniques. First, however, this paper focusses on MWR techniques.

## 2. A Simple Example

Before describing MWR techniques in full generality, it is useful to examine how they

---

<sup>1</sup> Danthine and Donaldson (1981) is a good example of the value of numerical techniques for studying comparative dynamics.

would be used in a simple example. Seeing these techniques used in a familiar context will help us understand the basic ideas behind these techniques. Suppose we want to solve the differential equation

$$y' = y \tag{2.1}$$

for  $t \in [0, T]$  together with the initial condition

$$y(0) = 1 \tag{2.2}$$

The solution to this problem is  $y(x) = e^x$ . There are many techniques which one can use to solve such an equation. The most popular for this problem are finite difference methods, such as the Euler and Runge-Kutta schemes, which convert differential equations into difference equations.<sup>2</sup> In this paper, however, we are interested in MWR techniques and will illustrate them by applying the basic idea to this simple ordinary differential equation.

The first step is to change the way we view (2.1). We begin by defining the operator

$$Ly \equiv y' - y \tag{2.3}$$

on the space of all continuously differentiable functions on  $[0, T]$ .  $L$  maps  $C^1$  functions into the space of  $C^0$  functions. The differential equation (2.1,2) can be formulated as the problem

$$Ly = 0 \tag{2.4}$$

together with (2.2).

The next step is to find a way to represent potential solutions on the computer. By the Weierstrass Theorem, any  $C^1$  function on  $[0, T]$  can be expressed as a countably infinite weighted sum of polynomial terms. However, since our computer can only handle finite objects, MWR methods focus on finding some finitely representable approximation,

$$\hat{y}(t; \vec{a}) \equiv 1 + \sum_{i=1}^n a_i t^i$$

---

<sup>2</sup> The Euler scheme fixes a step size  $\Delta x$ , and solves the difference equation

$$\hat{y}(x + \Delta x) = \hat{y}(x) + e^x \Delta x.$$

$y(0) = 1$  initiates the difference scheme, which then continues for  $n$  steps where  $n$  is the least integer such that  $n\Delta x > T$ .

which “fits” (2.4) with “small” error. Note that the initial condition has now been taken care of since  $\hat{y}(0; \vec{a}) = 1$  for any choice of  $\vec{a}$ . We have now reduced the infinite-dimensional problem to a finite-dimensional one since the problem now is to find a good  $\vec{a} \in R^n$ <sup>3</sup>.

We next operationalize the idea of a good “fit”. We define the residual function,

$$R(t; \vec{a}) \equiv L\hat{y}(t; \vec{a})$$

If  $\hat{y}(t; \vec{a})$  were a solution, then the residual function would be the zero function. MWR methods choose  $\vec{a}$  so as to make  $R(t; \vec{a})$ , when viewed as a function of  $t$ , as close to the zero function as possible.

The first approach is quite natural for economists. The *least-squares* approach defines the fit to be the  $L^2$  norm of the residual function, and chooses  $\vec{a}$  so as to minimize the “sum of squared residuals”:

$$\min_{\vec{a}} \int_0^T R(t; \vec{a})^2 dt$$

The least-squares method is a direct implementation of the idea to make small the error of the approximation. In general, one could develop alternative implementations by using different measures of the “loss” due to a nonzero  $R$ .

However, most MWR techniques find a good-fitting approximation in a less direct fashion. For these techniques, the basic idea is that the true solution would have a zero residual error function; in particular, its product with any other function is zero. Therefore, one way to find the  $n$  arguments of  $\vec{a}$  is to fix  $n$  functions,  $p_i(t)$ ,  $i = 1, \dots, n$ , and choose  $\vec{a}$  so that

$$\int_0^T p_i(t)R(t; \vec{a})dt = 0, \quad i = 1, \dots, n.$$

There are obviously many ways to implement this idea with different specifications of the  $p_i(t)$ . For example, in the *method of moments* implementation the  $p_i(t)$  are the first  $n$  polynomials.

A somewhat different method is the *collocation* method. It finds an approximation which satisfies the differential equation exactly at some chosen points. Specifically, for a fixed set

---

<sup>3</sup> Whereas finite-difference methods focus on computing values of a solution at specified values of  $t$ , MWR methods focus on computing coefficients of a decomposition of a solution. If I may borrow jargon from the time series literature, one can roughly think of finite-difference methods as state-space methods, and MWR methods as spectral methods.

of  $n$  points  $\{t_i\}_{i=1}^n$  in  $[0, T]$ , we choose  $\vec{a}$  so that

$$R(t_i; \vec{a}) = 0, \quad i = 1, \dots, n$$

In the interest of space, I will not report in detail on the quality of the solutions here. The results for this problem are that least-squares and method of moments produced good results, and simple collocation schemes, such as the case of uniformly spaced nodes, produced bad results. See Judd (1991) for a more complete discussion.

This simple example illustrates the basic approach of MWR procedures applied to a familiar problem. First we express the problem as the zero of some operator. We then express the candidate approximations as finite sums of simple functions. The coefficients of these finite sums are then fixed by imposing conditions which would be satisfied by the exact solution. For this simple problem, we have no other numerical problems since these procedures reduce it to a set of linear equations for  $\vec{a}$ . With this example in mind, we will now discuss the general Minimum Weighted Residuals method of solution and how to proceed when the problem does not reduce to linear equations in  $\vec{a}$ .

### 3. General Algorithm

We will next proceed to describe the approach in a more general context. The ability to do this is a strength of MWR methods. We begin by representing a problem as a solution to an operator equation:

$$\mathcal{N}(f) = 0$$

where  $\mathcal{N} : B \rightarrow B$ ,  $B$  is a Banach space of functions  $f : D \subset \mathbb{R}^N \rightarrow \mathbb{R}^M$ . In our simple example above,  $D = [0, T]$ ,  $f : D \rightarrow \mathbb{R}^1$ , and  $\mathcal{N} = \frac{d}{dt} - I$  where  $I$  is the identity operator. A wide variety of economic problems can be so represented. For example, in the optimal growth problems below, the domain  $D$  will represent the  $N$  state variables, the unknown function  $f$  will represent the decision rules, and the operator  $\mathcal{N}$  will represent  $M$  Euler conditions on the policy rules. Typically  $\mathcal{N}$  is a composition of algebraic operations, differential and integral operators, and functional compositions, and is frequently nonlinear.

We shall show how to implement the canonical MWR technique in a step-by-step fashion. We will first give an overview of the approach and then highlight the critical issues for each step and how the pieces interact.

The first step is to decide how to represent approximate solutions. One general way is to assume that our approximation,  $\hat{f}$ , is built up as a linear combination of simple functions.<sup>4</sup> We will also need a concept of when two linear combinations are close or far apart. Therefore, the first step is to choose a basis and an appropriate concept of distance:

Step 1) Choose a basis,  $\Phi = \{\varphi_i\}_{i=1}^{\infty}$  and an inner product,  $\langle \cdot, \cdot \rangle$ , over  $B$ .

The basis should be flexible, capable of yielding a good approximation for the solution, and the inner product should induce a useful norm on the space spanned by  $\Phi$ .

Next, we decide how many of these basis elements we will use:

Step 2) Choose a degree of approximation :  $n$

The approximation<sup>5</sup> is  $\hat{f} \equiv \sum_{i=1}^n a_i \varphi_i(x)$ . The best choice of  $n$ , cannot be determined *a priori*. Generally, the only "correct" choice is  $n = \infty$ . If the choice of the basis is good then larger  $n$  will yield better approximations, but one is most interested in the smallest  $n$  which yields an acceptable approximation. One initially begins with small  $n$  and increases  $n$  until some diagnostic indicates little is gained by continuing.

Step 1 lays down the topological structure of our approximation and Step 2 fixes the degrees of freedom of the approximation. Once we have made these basic decisions, we begin our search for an approximate solution to the problem. Since the true solution  $f$  satisfies  $\mathcal{N}(f) = 0$ , we will choose as our approximation some  $\hat{f}$  which makes  $\mathcal{N}(\hat{f})$  nearly equal to the zero function, where by near we refer to properties defined by the inner product,  $\langle \cdot, \cdot \rangle$ . Since  $\hat{f}$  is parameterized by  $\vec{a}$ , the problem reduces to finding a  $\vec{a}$  which makes  $\mathcal{N}(\hat{f})$  nearly zero. This search for  $\vec{a}$  is the focus of Steps 3-5.

Step 3) For a guess  $\vec{a}$ , compute the approximation,  $\hat{f} \equiv \sum_{i=1}^n a_i \varphi_i(x)$ , and the residual function,

$$R(x; \vec{a}) \equiv (\mathcal{N}(\hat{f}))(x)$$

The first guess of  $\vec{a}$  should reflect some initial knowledge about the solution. Since this

---

<sup>4</sup> Nonlinear combinations are also possible, but we stay with linear combinations here since linear approximation theory is a much more developed theory than nonlinear approximation theory.

<sup>5</sup> The convention is that the  $\phi_i$  increase in "complexity" and "nonlinearity" as  $i$  increases, and that the first  $n$  elements are used.



step tends to be problem specific, we shall discuss it in the context of individual problems. After the initial guess, further guesses are generated in Steps 4 and 5.

Step 4) Choose a collection of  $n$  functions,  $p_i : D \rightarrow R^M$ ,  $i = 1, \dots, n$  and, for each guess of  $\vec{a}$ , compute the  $n$  projections,  $P_i(\cdot) \equiv \langle R(\cdot; \vec{a}), p_i(\cdot) \rangle$ .

Step 5) By iterating over steps 3 and 4, find  $\vec{a}$  which sets the  $n$  projections equal to zero.

This general algorithm breaks the numerical problem into several distinct steps. It points out the many distinct techniques of numerical analysis which are important. First, in Steps 1 and 2 we choose the finite-dimensional space wherein we look for approximate solutions, hoping that within this set there is something "close" to the real solution. These steps require us to think seriously about approximation theory methods. Second, Step 4 will involve numerical integration if we cannot explicitly compute the integrals which define the projections. Third, Step 5 is a distinct numerical problem, involving the solution of a nonlinear set of simultaneous equations or the solution of a minimization problem. We shall now consider each of these numerical problems in isolation.

### *Choice of Basis*

There are many criteria which the basis and inner product should satisfy. The full basis  $\Phi$  for the space of candidate solutions should be "rich"; in particular, it should be complete in  $B$ . This will be necessary for proving any convergence results.

Computational considerations should also play a role in choosing a basis. The  $\varphi_i$  should be simple to compute. They also should be similar in size to avoid scaling problems. The basis elements should look something like the solution so that only a few elements can give a good approximation. While asymptotic results such as the Stone-Weierstrass Theorem may lull one into accepting polynomial approximations, practical success requires a basis where only a few elements will do the job. The individual terms should also be "different"; ideally, they should be orthogonal with respect to the inner product  $\langle \cdot, \cdot \rangle$ . The reasons are essentially the same as why one wants uncorrelated explanatory variables in a regression. Nonorthogonal bases will reduce numerical accuracy just as multicollinear regressors will enlarge confidence intervals. In our discussion below, we will see that algorithms which solve

for  $\vec{a}$  frequently must invert matrices. Accuracy of these inversions in the face of round-off error requires that the rows and columns should not be highly correlated. Orthogonal bases will help avoid nearly singular matrices in these intermediate steps.

Examples of possible bases are numerous. First, one may consider the ordinary polynomials,  $\{1, x, x^2, x^3, \dots\}$ . If  $B$  is the set of bounded measurable functions on a compact set then the Stone-Weierstrass Theorem assures us of their completeness in the  $L^1$  norm. However, they may not be a good choice since they are very similar. For example, they are all monotonically increasing and positive on  $R^+$  and they will not be orthogonal in any natural inner product on  $R^+$ . They will also vary a great deal in size. This does not mean that we cannot use ordinary polynomials, just that it is preferable to use orthonormal collections with the appropriate inner products. A generally useful choice are systems of Chebyshev polynomials, which I shall now discuss in detail. <sup>6</sup>

### *Chebyshev Polynomials*

Because of their usefulness, we will review the key properties of Chebyshev polynomials. They are defined over  $[-1, 1]$  by the formula

$$T_n(x) \equiv \cos(n \arccos x)$$

and are generated by the recursion scheme

$$T_0(x) = 1$$

$$T_1(x) = x$$

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x).$$

The restriction to  $[-1, 1]$  is inessential since one can define Chebyshev polynomials over any bounded interval by a linear transformation. One hint of their special properties is given by the identity  $T_n(\cos \theta) = \cos n\theta$  which in itself is an orthogonal sequence on  $[0, 2\pi]$ . Since this

---

<sup>6</sup> Nonpolynomial alternatives include various sequences of trigonometric and exponential functions. The choice depends on the range,  $D$ , computational demands, and the expected shape of a solution. In physics, trigonometric bases such as  $\{1, \sin x, \sin 2x, \sin 3x, \dots\}$  are often used since solutions are often periodic, allowing for Fourier series techniques. In economic problems, however, solutions are generally not periodic, and periodic approximations to nonperiodic functions require many terms.

identity ties Chebyshev polynomials to Fourier analysis, it is not surprising that Chebyshev polynomials are useful for approximation.

The Chebyshev polynomials obey the continuous orthogonality relationship

$$\int_{-1}^1 T_i(x)T_j(x)(1-x^2)^{-\frac{1}{2}}dx = 0, \quad i \neq j.$$

This shows that the Chebyshev polynomials are orthogonal with respect to the inner product defined by the weighting function  $(1-x^2)^{-\frac{1}{2}}$ . Also important for our purposes is the discrete orthogonality relationship

$$\sum_{\ell=1}^n T_i(z_\ell^n)T_j(z_\ell^n) = 0, \quad i \neq j$$

where, for the purposes of this paper, we define the  $z_\ell^n$ ,  $\ell = 1, \dots, n$ , to be the zeroes of  $T_n$ .

They are given by the formula

$$z_\ell^n \equiv \cos\left(\frac{(2\ell-1)\pi}{2n}\right), \quad \ell = 1, \dots, n.$$

Our interest in Chebyshev polynomials lies in their ability to approximate functions well.

Suppose  $f(x)$  is a continuous function and

$$c_j \equiv \frac{2}{n} \sum_{\ell=1}^n f(z_\ell^n)T_j(z_\ell^n).$$

Then the function

$$I_n^f(x) = -\frac{1}{2}c_0 + \sum_{k=1}^n c_k T_k(x)$$

agrees with  $f(x)$  on the  $z_\ell^n$ , thereby being a polynomial of degree  $n-1$  which interpolates  $f(x)$  at the  $z_\ell^n$ . In this way, we can approximate a function by interpolation at the points  $z_\ell^n$ . We call  $I_n^f$  the  $n$ 'th degree Chebyshev interpolant of  $f$ .

One can interpolate at any collection of  $n$  points. There are, however, strong reasons to choose  $I_n^f$ . Suppose that we want to find a  $n$ 'th degree approximation,  $p_n(x)$ , which keeps the maximum error,  $\|f - p_n\|_\infty$ , small. The following theorem indicates some advantages of  $I_n^f$ .

**Chebyshev Interpolation Theorem:** Suppose  $f \in C^k[a, b]$ . Then there is some  $d_k$  such that for all  $n$

$$\|f - I_n^f\|_\infty \leq \left(\frac{2}{\pi} \log(n+1) + 2\right) \frac{d_k}{n^k} \|f^{(k)}\|_\infty$$

This theorem (see Atkinson for proofs) says that the Chebyshev interpolant converges to  $f$  as we use more Chebyshev zeroes. Convergence may seem to be an unremarkable property, but counterexamples show that interpolation at uniformly spaced points does not necessarily converge as we use more points. This theorem also shows that the rate of convergence is more rapid for smooth functions. Other theorems show that Chebyshev interpolation is best in general and that few terms are needed to get a good approximation. Given these properties, it is clear that Chebyshev methods will be valuable for economic problems.

### *Other Bases*

Depending on the nature of the bases, MWR methods are separated into *Finite Element* and *Spectral* methods; both types have been used in economic applications. Spectral techniques use basis functions which are generally nonzero on  $D$ . These basis functions are also  $C^\infty$ , imposing (possibly undesirable) smoothness conditions on the approximate solutions. On the other hand, finite-element techniques use basis functions which have small support, that is, they are zero except over a small portion of  $D$ , and at each point of  $D$  all but a few of the basis elements are zero. The simplest common finite element basis on  $R^1$  uses *tent* functions, that is, functions which are zero, except on an interval where two linear pieces form a tent. Such bases span spaces of piecewise linear functions. More generally, finite element bases use low-order polynomials to find good local approximations, which are then pieced together to form the global approximation. The speed advantage of low-order polynomials implies that finite-element algorithms will have superior properties in many problems, particularly in multidimensional problems. Furthermore, the large size of the problems are cut down to manageable levels by exploiting the sparseness of the resulting Jacobians and Hessians. The disadvantage of finite-element methods is the difficulty of piecing together the local approximations if one wants to impose prior information concerning the smoothness of the solution. See Burnett (1987) for a basic discussion of Finite Element methods. Splines (see Prenter, 1989) are used sometimes in finite element methods. In this paper, most of the examples are known to have very smooth solutions and we will use a spectral approach with a Chebyshev polynomial basis.

Most interesting problems in economics involve more than one state variable – physical

versus human capital, capital stocks of duopoly competitors, wealth distribution across investor groups, etc. *Tensor product* methods build multidimensional basis functions up from simple one-dimensional basis functions. If  $\{\varphi_i(x)\}_{i=1}^{\infty}$  is a basis for functions of one real variable, then the set of pairwise products,  $\{\varphi_i(x)\varphi_j(y)\}_{i,j=1}^{\infty}$  is the tensor product basis for functions of two variables. To handle  $n$  dimensional problems in general, one can take the  $n$ -fold tensor product of a one-dimensional basis. One advantage of the tensor product approach is that if the one-dimensional basis is orthogonal, the tensor product is also orthogonal. The disadvantage is that the number of elements increases exponentially in the dimension. <sup>7</sup>

### *Choice of Projection Conditions*

Minimum Weighted Residual techniques include a variety of special methods. As discussed in the previous section, the least-squares method implements the basic idea directly by minimizing a  $L^2$  norm of the residual. More generally we specify some inner product,  $\langle \cdot, \cdot \rangle$ , of  $B$ , and use  $\langle \cdot, \cdot \rangle$  to measure the "size" of the residual function,  $R$ . The general strategy is to find an  $\vec{a}$  which makes  $R$  small. We will generally use inner products which are of the form

$$\langle f(x), g(x) \rangle \equiv \int_D f(x)g(x)w(x) dx$$

for some weighting function  $w(x)$ . There are several ways to proceed.

The first approach is quite natural for economists. The *least-squares* approach computes the  $L^2$  norm of the residual function, i.e.,  $\langle R(x; \vec{a}), R(x; \vec{a}) \rangle$  and chooses  $\vec{a}$  so as to minimize the "sum of squared residuals":

$$\min_{\vec{a}} \quad \langle R(x; \vec{a}), R(x; \vec{a}) \rangle.$$

We have thereby reduced the problem of solving a differential equation to solving a non-linear minimization problem in  $R^n$ , a tractable problem. Of course, the standard difficulties will arise. For example, there may be local minima which are not proper solutions. However,

---

<sup>7</sup> There are several alternative ways to forming bases. For example, there are ways of constructing multidimensional bases which avoid the "curse of dimensionality". See Judd(1991) for a discussion of these alternatives and their potential applications to economic problems.

there is no reason for these problems to arise more often here than in any other context, such as maximum likelihood estimation, where extremal problems are solved numerically.

The least-squares method is a direct implementation of the idea to make small the error of the approximation. In general, one could develop alternative implementations by using different norms. However, most MWR techniques find a good-fitting approximation in a less direct fashion. For these techniques, the basic idea is that the true solution would have a zero residual error function; in particular, its projection in all directions is zero. Therefore, one way to find the  $n$  arguments of  $\vec{a}$  is to fix  $n$  projections and choose  $\vec{a}$  so that the projection of the resulting residual function in each of those  $n$  directions is zero. Formally, these methods find  $\vec{a}$  such that  $\langle R, p_i \rangle = 0$  for some specified collection of functions,  $p_i$ . Different choices of the  $p_i$  defines different implementations of the MWR method.

It is clear that the least-squares and alternative implementations of MWR ideas are similar since one way to solve a least-squares problem is to solve the nonlinear set of equations generated by its first-order conditions:

$$\left\langle R, \frac{\partial R}{\partial a_i} \right\rangle = 0.$$

Of course, computer implementations of these methods would be quite dissimilar since least-squares problems are typically not solved by solving their first-order conditions. Seeing the least-squares method expressed as a system of projection equations gives us some indication why other methods may be better. The projection directions in the least-squares case, the gradients of the residual function, could be highly correlated. Furthermore, the projection directions depend on the guess for  $\vec{a}$ . In other methods, we have tight control over the projection directions, an important consideration in nonlinear problems. Also, in economic problems we may have a preference for approximations which have zero projections in certain directions, such as the average error in an Euler equation. Many of the alternative techniques will naturally include that condition.

One such alternative technique is the *Galerkin* method. In the Galerkin method we use the first  $n$  elements of the basis for the projection directions. Therefore,  $\vec{a}$  is chosen to solve the following set of equations:

$$P_i(\vec{a}) \equiv \langle R(x; \vec{a}), \varphi_i(x) \rangle = 0, \quad i = 1, \dots, n$$

Notice that here we have reduced the problem of solving a differential equation to one of solving a set of nonlinear equations. In some cases the Galerkin projection equations are the first-order conditions to some minimization problem, as is often the case in linear problems from physics. When we have such an equivalence, the Galerkin method is also called the *Rayleigh-Ritz* method. This is not as likely to happen in economics problems because of their inherent nonlinearities.

There are obviously many ways to implement the projection idea. Some of the other common ones are *method of moments*, *subdomain*, and *collocation*. If  $D \subset R^1$ , then the method of moments chooses the first  $n$  polynomials for the projection directions; i.e., we find  $\vec{a}$  which solves the system

$$P_i(\vec{a}) \equiv \langle R(x; \vec{a}), x^{i-1} \rangle = 0, \quad i = 1, \dots, n$$

If  $D$  is of higher dimension, then we project  $R$  against a sufficient number of low-order multivariate monomials. In the subdomain method, the idea is to find an approximation which is good on average on a collection of subsets which cover the whole domain. More specifically, we choose  $\vec{a}$  so that

$$P_i(\vec{a}) \equiv \langle R(x; \vec{a}), I_{D_i} \rangle = 0, \quad i = 1, \dots, n$$

where  $\{D_i\}_{i=1}^n$  is a sequence of intervals covering  $D$ , and  $I_{D_i}$  is the indicator function for  $D_i$ .

The collocation method is another example of a projection method. Recall that a collocation method chooses  $\vec{a}$  so that the functional equation holds exactly at  $n$  fixed points. That is, we choose  $\vec{a}$  to solve

$$R(x_i; \vec{a}) = 0, \quad i = 1, \dots, n$$

where  $\{x_i\}_{i=1}^n$  are  $n$  fixed points from  $D$ . This is a special case of the projection approach since  $R(x_i; \vec{a})$  equals  $\langle R(x; \vec{a}), \delta(x - x_i) \rangle$ , the projection of  $R(x; \vec{a})$  against the Dirac delta function at  $x_i$ .

*Orthogonal collocation* is the method where the  $x_i$  are the zeroes of the  $n$ 'th basis element and the basis elements are orthogonal with respect to the inner product. It turns out to be a particularly powerful application of MWR ideas when used in conjunction with a Chebyshev

polynomial basis. This is not a surprise in light of the Chebyshev Interpolation Theorem. Suppose that  $R(z_i^n; \vec{a}) = 0$ ,  $i = 1, \dots, n$ , where, recall, we have defined the  $z_i^n$  to be the  $n$  zeroes of  $T_n$ . As long as  $R(x; \vec{a})$  is smooth in  $x$ , the Chebyshev Interpolation Theorem says that these zero conditions force  $R(x; \vec{a})$  to be close to zero. The optimality of Chebyshev interpolation also says that if one is going to use collocation, these are the best possible points to use. Even after absorbing these considerations, it is not certain that even orthogonal collocation is a reliable method. Its performance turns out to be surprisingly good.

### *Evaluation of Projections*

Step 3 is the meat of the problem since the major decision in the MWR approach is choosing which projection conditions to use, and the major computational task is the computation of those projections. The collocation method is fastest in this regard since it only uses the value of  $R$  at  $n$  points. More generally, the projections will involve integration. In some cases one may be able to explicitly perform the integration. This is generally possible for linear problems, and possible for special nonlinear problems in physics and chemistry. However, our experience with the economic applications below is that this will often be impossible for nonlinear economic problems. We instead need to use quadrature techniques to compute the integrals associated with the evaluation of  $\langle \cdot, \cdot \rangle$ . A typical quadrature formula approximates  $\int_a^b f(x)g(x) dx$  with a finite sum  $\sum_{i=1}^n w_i f(x_i)$  where the  $x_i$  are the quadrature nodes and the  $w_i$  are the weights. Since these formulas also evaluate  $R$  at just a finite number of points, quadrature-based MWR techniques are essentially weighted collocation methods. The advantage of quadrature formulas is that information at more points is used to compute a more accurate approximation of the projections.

### *Initial Guesses*

As always with nonlinear computations, having a good initial guess is important. Fortunately, this is generally not a big problem. Typically, there are degenerate cases for which we can find the solution, which in turn will be a good guess for the problem we want to solve. These considerations are often problem-specific ways of generating initial guesses.

### *Finding the Solution*

Step 5, which determines  $\vec{a}$  by solving the projection conditions specified in Step 3, uses



either a minimization algorithm (in the least-squares approach) or a nonlinear algebraic equation solver to solve the system  $P(\vec{a}) = 0$ . Since I do not do anything unusual in this step and economists have some experience with solving optimization problems and nonlinear equations, I will not discuss how these problems are solved numerically. I use standard FORTRAN software for both kinds of problems. I use the Stanford Optimization Laboratories' MINOS and NPSOL packages for minimization problems. These are quite powerful and flexible packages, capable of handling arbitrary nonlinear constrained optimization problems. Simpler unconstrained routines include IMSL's ZXMIN. For nonlinear equation solving, I use MINPACK programs or IMSL routines ZSPOW or ZSCNT.

### *Coordination among Steps 1-5*

We now see what is needed for efficiency. We need basis functions which are easy to evaluate since they will be frequently evaluated. The integration in Step 4 must be accurate but fast. This can be helped by using quadrature formulas which work well the basis. Finally, the nonlinear equation solver in Step 5 needs to be efficient and should be able to use all the information arising from Step 4 calculations. Step 5 will typically use gradient information about the integrals of step 4. It is therefore important to do those gradient calculations quickly, doing them analytically when possible. Problem formulations which make gradient calculation easy will have a substantial advantage.

A particularly important interaction is that between the choice of a basis and the solution of the nonlinear problem in  $R^n$ . Most methods for solving the system  $P(\vec{a}) = 0$  will invert its Jacobian,  $P_{\vec{a}}(\vec{a}) = 0$ . If this matrix is nearly singular near the solution, the accuracy of this inversion will be poor due to round-off error and convergence will be slow. The *spectral condition number*, defined to be the ratio of the largest (in magnitude) and smallest eigenvalues of the Jacobian, is a commonly used index of being nearly singular. A condition number of  $10^6$  tells you that you lose roughly six significant digits when you solve for  $\vec{a}$ . Bases with ordinary polynomials can easily generate Jacobians with condition numbers on the order of  $10^{10}$ , in which case one has probably lost all significant digits on, say, a 13-digit machine since other operations will surely lose a few digits.

Some authors, such as den Haan and Marcet (1990), and Wright and Williams (1990),

have experienced no problems with ordinary polynomials. This is not surprising since they generally use low degree polynomials (den Haan and Marcet often use only linear polynomials), in which case the Jacobian has a low condition number. Discussions with individuals who have used ordinary polynomials to calculate more flexible nonlinear approximations and my own experience with ordinary polynomials attest to the fact that these theoretical problems do arise in practice. While the nonlinearities in  $P$  may still cause trouble, one reduces the likelihood of conditioning problems by starting with an orthogonal basis.

Most methods used in numerical analysis of economic models fall within the general description above. We will see these connections below when we compare how various methods attack a common problem. The key fact is that the methods differ in their choices of basis, fitting criterion, and quadrature techniques. With the general method laid out, we will now report on a particularly important application, discrete-time optimal growth, to show its usefulness.

#### 4. Discrete – Time Optimal Growth

In this section we examine optimal growth problems in discrete time and show how MWR techniques can be adapted to calculate solutions. This will be valuable for four reasons. First, discrete time is commonly used in dynamic economic analyses. Second, in the deterministic case we will be able to do a thorough job in checking our solutions since other reliable, but slow, algorithms are available to deliver tight bounds on the true solution. Third, the stochastic case is one which has been studied by many others with various numerical techniques. In fact, one point we make below is that most of these procedures are really MWR methods. By recognizing the common MWR approach underlying these procedures, we can better understand their differences, particularly in accuracy and speed. We hope that the comparative performances of these various implementations of MWR ideas in the discrete-time stochastic optimal growth problem is indicative of their relative value in other future problems.

Fourth, the application of MWR methods to this class of mathematical problems is unusual. MWR methods are used extensively for solving ordinary and partial differential equations in the physical sciences; therefore, their value in solving continuous-time economic

models is assured. However, the functional equations describing discrete-time optimal growth problems do not fit either class. Therefore, it is of substantial technical interest to show that MWR methods do a good job in solving the kind of discrete-time functional equations which commonly arise in economic analyses.

### *Deterministic Optimal Growth*

We first examine a deterministic growth problem. We assume that utility is

$$\sum_{t=0}^{\infty} \beta^t u(c_t) . \quad (4.1)$$

We first examine the deterministic case; hence, capital obeys

$$k_{t+1} = f(k_t) - c_t \quad (4.2)$$

As shown in Bizer and Judd (1989), to calculate the optimal consumption policy,  $h(k)$ , it is enough to focus on the Euler equation,

$$\begin{aligned} 0 &= u'(h(k)) - \beta u'(h(f(k) - h(k))) f'(f(k) - h(k)) \\ &\equiv (\mathcal{N}(h))(k) \end{aligned} \quad (4.3)$$

There are procedures for calculating extremely accurate solutions to this problem. If the capital stock could be only a finite number of values (i.e., "lumpy") then dynamic programming can solve out for the equilibrium exactly. Monotonicity properties of the problem, specifically the fact that  $h(k)$  is monotonically increasing, allow us to compute explicit and tight bounds on the true continuous state space solution by examining discretized problems. To this end, I computed the solution to our problem assuming that  $k$  must be some multiple of .000001, where 1.0 is the steady state. We solved for the policy function over the range [.5, 1.3], a total of 800,000 capital stocks. The main difficulty in this calculation is the need for space to store the policy function. Because this is a one-dimensional deterministic problem, time is no problem for this calculation, taking only a few minutes<sup>8</sup> on a VAX 8650, which is roughly three times as fast as a Compaq 386/20.

<sup>8</sup> I did *not* do value function iteration, a method which would have taken much longer. Instead I used the fact that we know the policy function at the steady state and that both the consumption and savings functions are monotonic to directly compute the policy function in one pass beginning with the steady state. I suspect that something similar may also be possible to create a good benchmark for the stochastic problem.

We shall now describe the details of a MWR approach to the problem. First of all, our approximation to  $h$  is parametrically given by

$$\hat{h}(k; \vec{a}) = \sum_{i=1}^n a_i \psi_i(k)$$

where  $\psi_i(k) \equiv T_{i-1}(2\frac{k-k_m}{k_M-k_m} - 1)$  and  $n$  is the number of terms used. The domain  $D$  of our approximation will be  $[k_m, k_M]$ . Since the special properties of Chebyshev polynomials apply to their restriction to  $[-1, 1]$ , we need to apply the linear transformation  $2\frac{k-k_m}{k_M-k_m} - 1$  to  $k$  to permit us to form Chebyshev polynomials on  $D$ .  $k_m$  and  $k_M$  are chosen so that the solution will have  $k$  confined to  $[k_m, k_M]$ . In particular,  $[k_m, k_M]$  must contain the steady state, a point which we can determine before calculations begin.

Given the Euler equation, (4.3), the residual function becomes

$$R(k; \vec{a}) = u'(h(k; \vec{a})) - \beta u'(h(f(k) - h(k; \vec{a}); \vec{a})) f'(f(k) - h(k; \vec{a}))$$

To compute  $\vec{a}$ , we can do one of several things. First, we consider collocation. We choose  $n$  values of  $k$ , denoted by  $k_i$ ,  $i = 1, \dots, n$ . We then choose  $\vec{a}$  so that  $R(k_i; \vec{a}) = 0$  for each  $i$ . Orthogonal collocation chooses the  $k_i$  to be the  $n$  zeroes of  $\psi_{n+1}$ , which are themselves linear transforms of the Chebyshev zeroes,  $z_i^n$ , defined above.

The Galerkin method is another alternative. We need to define a norm. Since we use Chebyshev polynomials as a basis, we will use the inner product

$$\langle f(k), g(k) \rangle \equiv \int_{k_m}^{k_M} f(k)g(k)w(k)dk$$

where

$$w(k) \equiv \left(1 - \left(2\frac{k-k_m}{k_M-k_m} - 1\right)^2\right)^{-\frac{1}{2}}.$$

With this choice of inner product, the basis is orthogonal and close to being orthonormal.

The Galerkin method computes the  $n$  projections

$$P_i(\vec{a}) \equiv \int_{k_m}^{k_M} R(k; \vec{a}) \psi_i(k) w(k) dk, \quad i = 1, \dots, n$$

and chooses  $\vec{a}$  so that  $P(\vec{a}) = 0$ . Here the difficulty is that each  $P_i(\vec{a})$  needs to be computed numerically. We used Gauss-Chebyshev quadrature since the basis elements are Chebyshev

polynomials. That is, we approximate  $P_i(\bar{a})$  with

$$\sum_{j=1}^m R(k_j; \bar{a}) \psi_i(k_j).$$

for some  $m > n$ , with the  $k_j$  being the  $m$  zeroes of  $\psi_{m+1}$ .

We get some idea of the accuracy of these MWR techniques (which yielded practically indistinguishable results as we moved from collocation to Galerkin with varying choices of  $m$ ) by comparing it with the presumably very accurate solution from the discretized problem. The results are shown in Table 1. We assumed a CRRA utility function,  $u(c) = c^{\gamma+1}/(\gamma+1)$  and a Cobb-Douglas production function  $f(k) = Ak^\alpha$  where  $A$  is chosen so that the deterministic steady state is  $k = 1$ . The first column gives the consumption choice as a function of capital indicated by the discrete state space method. The second through sixth columns give the difference between the discrete solution and the MWR solution, *multiplied by 10,000*, for  $n = 20, 10, 7, 5, 3$ , where  $n$  is the number of Chebyshev polynomials included in the approximation. Note that the approximation of the aggregate consumption function for  $n = 10$  disagrees with the discrete state space result by no more than one part in 10,000, an acceptable error for most purposes. Note that the  $n = 10$  approximation appears to be better than  $n = 20$ . This may be due to error in the discrete state space algorithm since the differences are slight. These numerical experiments indicate that the MWR method works well for the discrete-time optimal growth model, *demonstrating its usefulness even for the nonstandard functional equations which arise often in discrete-time economic models.*

While I did not implement it, I should mention the least-squares approach. In the least-squares procedure, one forms the sum of squared errors

$$\min_{\bar{a}} \int_{k_m}^{k_M} R(k; \bar{a})^2 dk$$

Again we have an integral which needs to be repeatedly evaluated. Standard quadrature considerations indicate that the Gauss-Legendre quadrature formula is the most appropriate (see Judd (1991)).

### *Stochastic Optimal Growth*

We next turn to a stochastic optimal growth model. This example will show us how to handle multidimensional problems and the conditional expectations which arise in stochastic

dynamic problems. More specifically, we examine the problem

$$\max E \left\{ \sum_{t=0}^{\infty} \beta^t u(\bar{c}_t) \right\} \quad (4.4)$$

$$k_{t+1} = \theta_t f(k_t) - c_t \quad (4.5)$$

where  $\theta_t$  is a stationary AR(1) multiplicative productivity parameter. We will assume that the productivity shock obeys  $\ln \theta_{t+1} = \rho \ln \theta_t + \epsilon_{t+1}$  and that the  $\epsilon_t \sim N(0, \sigma^2)$  are independent. The multiplicative specification of the shock is clearly not essential.

In this problem, both the beginning-of-period capital stock and the current value of  $\theta$  are needed for a sufficient description of the state. Hence, the Euler equation<sup>9</sup> is

$$u'(h(k, \theta)) = \beta E \left\{ u'(h(\theta f(k) - h(k, \theta), \bar{\theta})) \bar{\theta} f'(\theta f(k) - h(k, \theta)) \mid \theta \right\} \quad (4.6)$$

At this point, we will engage in a little trickery. The main point of this paper is that good numerical algorithms can be constructed in a systematic fashion using the MWR approach. While a systematic approach is useful, one can often increase speed by making an insightful transformation of the problem (in short, a lucky guess). We shall now make one. We know that MWR algorithms work well for linear problems. Perhaps our algorithm will do better if we make it more like a linear problem. To that end, rewrite (4.6) as

$$0 = h(k, \theta) - (u')^{-1} \left( \beta E \left\{ u'(h(\theta f(k) - h(k, \theta), \bar{\theta})) \bar{\theta} f'(\theta f(k) - h(k, \theta)) \mid \theta \right\} \right) \quad (4.7)$$

Note that the RHS of (4.7) has two terms, one linear in  $h(k, \theta)$ , and the other is similar to a CRTS function of next period's potential consumption values.

The procedure is similar to the deterministic case, but there are some extra twists due to the stochastic shocks. First of all our approximation of the policy function is now given by the double sum

$$\hat{h}(k, \theta; \mathbf{a}) = \sum_{i=1}^{n_k} \sum_{j=1}^{n_\theta} a_{ij} \psi_{ij}(k, \theta)$$

<sup>9</sup> While we have concentrated on optimal growth problems, nowhere is the optimality of equilibria used. Bizer and Judd (1989) show that the equilibrium of a taxed economy can also be described by an Euler equation similar to (4.6). See Judd (1991) for a more general discussion of taxed economies. Klenow (1991) demonstrates the efficiency and speed of MWR methods for models with elastic labor supply and externalities. It is clear that any equilibrium which can be expressed as a system of Euler equations can be efficiently computed by MWR methods.

where  $\psi_{ij}(k, \theta) \equiv T_{i-1}(2\frac{k-k_m}{k_M-k_m}-1)T_{j-1}(2\frac{\theta-\theta_m}{\theta_M-\theta_m}-1)$ . This tensor-product approach to two-dimensional approximation is reasonable here because the dimension is low and the tensor-product approach allows for a straightforward extension of the one-dimensional techniques to higher dimensions. Also note that we have added four parameters to our problem:  $k_M$ ,  $k_m$ ,  $\theta_M$ , and  $\theta_m$ .  $k_m$  and  $k_M$  are chosen so that  $k$  is usually confined to  $[k_m, k_M]$ , and similarly for  $\theta_M$ , and  $\theta_m$ . It is hoped that our solution is a good approximation on the rectangle  $[k_m, k_M] \times [\theta_m, \theta_M]$ .

Since tomorrow's log productivity level,  $\ln \theta'$ , conditional on today's log productivity level,  $\ln \theta$ , is distributed as  $\rho \ln \theta + \sigma Z$  for  $Z \sim N(0, \sigma^2)$ , the residual equation, (4.7), becomes

$$0 = \hat{h}(k, \theta; \mathbf{a}) - (u')^{-1} \left( \beta \int_{-\infty}^{\infty} u'(\hat{h}(\theta f(k)) - \hat{h}(k, \theta; \mathbf{a}), e^{\sigma z \theta^\rho}) e^{\sigma z \theta^\rho} f'(\theta f(k) - \hat{h}(k, \theta; \mathbf{a})) \frac{e^{-z^2/2}}{\sqrt{2\pi}} dz \right)$$

The integral in this equation needs to be evaluated numerically. If we define  $I(k, \theta, \mathbf{a}, z)$  to be

$$I(k, \theta, \mathbf{a}, z) \equiv u'(\hat{h}(f(k)\theta - \hat{h}(k, \theta; \mathbf{a}), e^{\sigma z \theta^\rho})) e^{\sigma z \theta^\rho} f'(\theta f(k) - \hat{h}(k, \theta; \mathbf{a})) \pi^{-\frac{1}{2}}$$

we can then approximate the above integral in the following way:

$$\int_{-\infty}^{\infty} I(k, \theta, \mathbf{a}, z) \frac{e^{-z^2/2}}{\sqrt{2}} dz \doteq \sum_{j=1}^{m_z} I(k, \theta, \mathbf{a}, \sqrt{2}z_j) w_j$$

where  $w_j, z_j$  are Gauss-Hermite quadrature weights and points.

So we now define the residual function to be

$$R(k, \theta; \mathbf{a}) = \hat{h}(k, \theta; \mathbf{a}) - (u')^{-1} \left( \beta \sum_{j=1}^m I(k, \theta, \mathbf{a}, z_j) w_j \right)$$

With this residual function, we can proceed as we did in the deterministic problem. The collocation method starts with choosing  $n_k$  capital stocks,  $\{k_i\}_{i=1}^{n_k}$ , and  $n_\theta$  productivity levels,  $\{\theta_i\}_{i=1}^{n_\theta}$ , and then finding  $\mathbf{a}$  so that  $R(k_i, \theta_j; \mathbf{a}) = 0$  for all  $i = 1, \dots, n_k$  and  $j = 1, \dots, n_\theta$ .

The Galerkin approach wants to form the  $n_k n_\theta$  projections

$$P_{ij}(\mathbf{a}) \equiv \int_{k_m}^{k_M} \int_{\theta_m}^{\theta_M} R(k, \theta; \mathbf{a}) \psi_{ij}(k, \theta) d\theta dk$$

and chooses  $\mathbf{a}$  so that  $P_{ij} = 0$  for all  $i$  and  $j$ . Again,  $P_{ij}(\mathbf{a})$  needs to be computed numerically. Since the  $\psi_{ij}$  are Chebyshev polynomials, we use Gauss-Chebyshev quadrature points, which are the zeroes of Chebyshev polynomials. If we use  $m_k$  values of  $k$  and  $m_\theta$  values of  $\theta$  to compute the projections, we solve the system

$$\hat{P}_{ij}(\mathbf{a}) \equiv \sum_{i=1}^{m_k} \sum_{j=1}^{m_\theta} R(k_i, \theta_j; \mathbf{a}) \psi_{ij}(k_i, \theta_j) = 0 \quad (4.7)$$

where

$$\begin{aligned} k_i &= k_m + \frac{1}{2}(k_M - k_m)(z_i^{m_k} + 1) & i = 1, \dots, m_k \\ \theta_j &= \theta_m + \frac{1}{2}(\theta_M - \theta_m)(z_j^{m_\theta} + 1) & j = 1, \dots, m_\theta \\ z_i^n &\equiv 1 + \cos\left(\frac{(2i-1)\pi}{2n}\right), & i = 1, \dots, n. \end{aligned}$$

We now have the nonlinear equation in  $\mathbf{a}$  which will generate our approximation,  $\hat{h}(k, \theta)$ . Below I will discuss the variety of specifications I ran.

For the initial guess, I used the consumption function which is linear in capital, goes through the deterministic steady state and consumes zero at zero capital stock. Convergence for this problem was slower than for the deterministic case, but still under a few minutes in all cases. The key factor in the slower speed was the increased number of unknown coefficients. Numerical analysis theory indicates that solutions of nonlinear equations take time roughly proportional to the cube of the number of unknowns. The increase in time as I added unknown coefficients wasn't quite that bad, but clearly at least quadratic.

Again I should note that least-squares fitting would proceed by solving

$$\min_{\mathbf{a}} \int_D R(k, \theta; \mathbf{a})^2 dk d\theta$$

where the Gauss-Legendre rule is the sensible way to approximate the indicated integral.

### *Accuracy Checks*

Once we have a candidate solution, we want to run some checks to test its quality. Accuracy for the stochastic case is more difficult to check than in the deterministic case. Such checks should use aspects of the solution which are not used in the solution method.

Some might wonder just how accurate we need an approximation to be. In fact, Danthine, et al. (1987) have argued that the linear approximation computed in Magill (1977) is



adequate for macroeconomic purposes. However, their tests concerned only a few economic variables such as consumption and output. In light of the results in Magill (1977) (and, more generally, in Bensoussan (1988)), this is not surprising. The adequacy of the linear approximation is much less likely once one turns to other economic variables, such as risk premia, term structure of interest rates, and their correlations since these variables involve higher-order properties of tastes and technology, as documented in Judd (1990). Therefore, here we will attempt to find approximations which are as accurate as possible given limitations on computer time and space.

A direct procedure is to check how much, if at all,  $\mathcal{N}(\hat{h})$  differs from the zero function. A standard check is to see just how close  $\mathcal{N}(\hat{h})$  is to the zero function. First we should understand what a deviation from zero means in economic terms. Consider (4.7). It is a difference between consumption at a capital stock  $k$  and productivity level  $\theta$ , and what that consumption would be if an optimizing agent would choose if he knew that tomorrow he would use the consumption rule  $\hat{h}(k, \theta)$ . Therefore, our residual equation applied to the approximate solution is an expression of the optimization error in consumption terms. Dividing (4.7) by  $\hat{h}(k, \theta)$  yields a dimension-free quantity expressing that optimization error as a fraction of current consumption. One advantage of this direct approach to accuracy checking is that we can express the resulting errors in economic terms, essentially in terms of how irrational agents would be in using the approximate rule.

### *Results*

With our approximation method specified and accuracy checks determined, we can now see just how fast we can compute our approximations, and how accurate they are. To do so we make taste and technology specifications which bracket a wide range of empirically plausible values. Table 2 summarizes typical cases. We again assumed  $u(c) = c^{\gamma+1}/(\gamma + 1)$ , and  $f(k) = Ak^\alpha$  where  $A$  is chosen so that the deterministic steady state is  $k = 1$ . Throughout Table 2 I set  $k_m = .3$  and  $k_M = 3.0$ . I also chose  $m_z = 8$ , the eight-point (which is fifteenth-order accurate) Gauss-Hermite quadrature rule, to compute the conditional expectation. Table 2 lets  $\sigma$  and  $\rho$  vary; for each choice,  $\theta_M$  is set equal to the long-run value of  $\theta$  which would occur if  $\epsilon_t = 3\sigma$  for all  $t$ , and  $\theta_m = 1/\theta_M$ . It is extremely unlikely for  $\theta$  to spend much

if any time outside of  $[\theta_m, \theta_M]$ .

The different blocks in Table 2 correspond to various choices of  $(n_k, n_\theta, m_k, m_\theta)$ . For example  $(2, 2, 2, 2)$  is the case where the approximated policy function is of the form  $a_1 + a_2k + a_3\theta + a_4k\theta$ , that is, an element of the tensor product of the linear basis in  $k$  and the linear basis in  $\theta$ .  $(2, 2, 2, 2)$  also indicates that we choose  $\bar{a}$  so that the Euler equation fits exactly at the four zeroes of  $\psi_{3,3}$ , an orthogonal collocation procedure.  $(10, 6, m_k, m_\theta)$  corresponds to allowing  $k$  terms up to  $k^9$ ,  $\theta$  terms up to  $\theta^5$ , and all possible pairwise products of those  $k$  and  $\theta$  terms. Orthogonal collocation was used whenever  $n_k = m_k$  and  $n_\theta = m_\theta$ . Otherwise, we implement a Gaussian quadrature version of the Galerkin procedure, as described in (4.8).

To test for the quality of the candidate solution, we evaluate (4.7) at a large number of  $(k, \theta)$  combinations which themselves were not used in finding a solution. We define the relative error at  $(k, \theta)$  to be  $E(k, \theta) \equiv R(\hat{h})(k, \theta)/\hat{h}(k, \theta)$ . The entries are the base 10 logarithm of various norms of  $E(\hat{h})$ . Columns 1, 2, and 3 report  $\log_{10} \| E \|_\infty$ ,  $\log_{10} \| E \|_2$ , and  $\log_{10} \| E \|_1$ . All of these norms were calculated by using 1500 grid points in  $[k_m, k_M] \times [\theta_m, \theta_M]$ , 50 in  $[k_m, k_M]$  and 30 in  $[\theta_m, \theta_M]$ . Base 10 logs of these norms are natural measures for our exercise.  $\log_{10} \| E \|_\infty$  is the (lothe maximum error we found,  $\log_{10} \| R \|_1$  represents the average error, and  $\log_{10} \| E \|_2$  represents the  $L^2$  norm of the errors. For example, an entry of -3 under the  $\log_{10} \| E \|_\infty$  column says that a person with \$1,000 of consumption makes *at most* a one dollar error in current consumption. Since solution paths concentrate near the center of  $D$ , we are particularly concerned about accuracy there. We define  $E_I$  to be  $E$  restricted to the inner rectangle  $[k'_m, k'_M] \times [\theta'_m, \theta'_M]$  where  $y' = \frac{1}{2}(y + 1)$ ,  $y = n_k, n_\theta, m_k, m_\theta$ . Columns 4 and 5 report  $\log_{10} \| E_I \|_\infty$  and  $\log_{10} \| E_I \|_2$ .

There are several points to note. First, note that the errors are rather small. Even for the  $(2, 2, 2, 2)$  case, the errors are small, roughly one dollar per hundred, as long as the utility function is as concave as  $\log c$ . Second, as we allow the approximation to use more terms the errors fall until in the  $(10, 6, 10, 6)$  case, we often find optimization errors of less than one dollar per million. Third, the various norms of the residual function have very similar values, indicating that the errors are uniformly small. In particular, the similarity in values for the norms of  $E$  and  $E_I$  indicates that the solution is almost as good at the edges of the

state space as in the middle.

Fourth, these methods are fast. The solutions in the (2, 2, 2, 2) case were solved in .2 to .4 seconds and in the (4, 3, 4, 3) case in 1.1 to 2 seconds. The slow parameterization throughout the table was  $\gamma = -15.0$ ,  $\rho = .8$ , and  $\sigma = .04$ , which took 3 seconds for the (4, 3, 4, 3) case. The speed advantage of orthogonal collocation is demonstrated by the fact that the (7, 5, 7, 5) cases generally took 8 to 18 seconds, whereas the (7, 5, 20, 12) Galerkin cases took three times as long, which is expected since the projections were integrals using 240 points instead of 35. An intriguing exception was the slow parameterization which took nearly two minutes for collocation but only a minute and a half for Galerkin. Apparently the extra information used by the Galerkin procedure helped the nonlinear equation solver to avoid bad directions. (10, 6, 10, 6) cases generally took 27 to 72 seconds with the bad parameterization taking 100 seconds. The corresponding (10, 6, 20, 15) cases took roughly four times as long.

The dependence of speed and accuracy on parameters was clear. The algorithm was faster for less concave utility functions, less persistent productivity shocks, and smaller productivity shocks. The slowest parameterization,  $\gamma = -15.0$ ,  $\rho = .8$ , and  $\sigma = .04$ , presents difficulties for the algorithm since the policy function involves little saving on average and the range of  $\theta$  and  $k$  is large. Since there is little central tendency in the true solution, the algorithm will have a tendency to examine candidates which are qualitatively very bad, such as policy rules which violate transversality, and give little help towards finding a good approximation.

Accuracy was greatest for highly concave utility, and for small productivity shocks lacking in persistence. The dependence on concavity is intuitive since highly concave utility implies that consumption is nearly equal to permanent income, whereas nearly linear concavity would imply a highly nonlinear consumption rule with very little consumption except near the deterministic steady state capital stock. With larger and more persistent shocks, the range of  $\theta$ ,  $[\theta_m, \theta_M]$ , grows, forcing the approximation to cover a larger area and making accuracy more difficult for a fixed number of degrees of freedom.

While the number of cases reported in Table 2 are small note that they cover a wider range of parameter values. I have also performed these same calculations for  $\gamma = -10.0$ ,  $-7.0$ ,

-5.0, -4.0, -2.0, and -.5,  $\sigma = .02$  and  $.03$ , and  $\rho = .6$ ,  $.5$ , and  $.4$ , and several other choices of  $(n_k, n_\theta, m_k, m_\theta)$ . Since the patterns described above held over the whole collection of calculations, it suffices to report the cases in Table 2.

Fourth, note that the orthogonal collocation method does remarkably well given the small amount of computation. This is indicated by the small optimization errors reported and the fact that by going to Galerkin procedures which use many more grid points, very little if any accuracy is gained.

Another way to check for accuracy is to see how the computed solution changes when we use higher-order and different quadrature schemes. Again, I found trivial sensitivity to these changes. For example, using  $m_z = 4$  instead of 8 resulted in very few differences (Table 2 was unchanged) and cut the running time by almost half.

## 5. Comparisons with Alternative Methods

Numerical solutions of the stochastic growth model have been examined extensively in the literature, utilizing various methods. This allows us to compare MWR methods with those alternatives.

Most approximation techniques used in economics fit into the MWR framework. Gustafson (1958) used piecewise linear functions and fit them at points, basically a collocation method with a finite-element basis. This is also the method used in Bizer and Judd (1989), which solves for a dynamic general equilibrium with taxes and risk. Bizer and Judd (1989) prove existence of equilibrium by monotone operator methods arising from the Euler equation. The existence was constructive and strongly suggests convergence of the numerical algorithm. Wright and Williams (1990, 1982) have used ordinary polynomials. These authors typically used a "time iterative" technique based on the Euler equation, that is, they computed the time  $t$  solution from the time  $t + 1$  solution. The disadvantage with that approach is the slow convergence common to algorithms which are (essentially) motivated by the iterated value function representation of dynamic programming. The differences in running time were substantial. Typical cases in Bizer-Judd took a few minutes for the deterministic case, and about 15 minutes for the stochastic case. Furthermore, Bizer and Judd allowed only two values for the productivity shock, whereas the calculations above approximated lognormal

productivity shocks. The advantage of the time-iterative method lies in its use of any available monotonicity properties of the operator, in which case the method is very reliable. On the other hand, the approach we used above ignores monotonicity and the time structure, gaining speed but risking convergence problems.

Taylor and Uhlig (1990) and their collaborators discussed several methods to solve the stochastic optimal growth problem. Christiano (1990) solves it by looking at a related model where the capital stock is forced to remain on a finite grid of possible values. This is the same as approximating the optimal investment policy with a step function. Generally, methods which discretize the state space are really examples of MWR techniques where the basis consists of step functions. However, step functions are highly inefficient ways to approximate the true investment policy function which we know to be smooth. Because of this, the discretized state space method is unlikely to be of much value in economic analysis.

Another method discussed in Taylor and Uhlig is the parameterized expectations method of Marcet (1989). As we do in Step 1, Marcet specifies a basis, restricting the set of functions to exponentials of low-order polynomials. However, this basis is probably not orthogonal given the other aspects of the algorithm. As we do, he evaluates the fit of a candidate approximation, parameterized by a choice of coefficients for the exponential polynomial. However, instead of using a standard quadrature approach, he simulates the dynamic path which results from a candidate approximation, calculating the Euler equation errors along the simulated path, and use them to construct a norm for the aggregate error. This simulation approach to the calculation of the implicit norm has many undesirable properties. First, since the Euler equation errors are evaluated at random points, his is essentially a Monte Carlo quadrature approach to computing the total error. To make up for the errors which are introduced by a Monte Carlo approach, long simulations must be run, substantially reducing the computational efficiency of the algorithm.

Second, the points at which the errors are calculated depend on the guess for  $\vec{a}$ . Such an endogeneity serves no clear purpose and could generate extraneous solutions. This feature of Marcet's algorithm is particularly problematic if it were applied to dynamic games. In many dynamic games, there are states which are never reached in equilibrium, and would not be

checked out by a simulation-based fitting method. However, the equilibrium solution at those states may be very important in the overall determination of the equilibrium. Therefore, there could be many non-subgame perfect equilibria which would pass muster in Marcet's algorithm. This is a strong example demonstrating the dangers of an algorithm which allows the approximation to influence the evaluation of its accuracy. In contrast, the MWR methods discussed above can be applied to dynamic games (see Judd (1990)).

As in Step 5, Marcet tries many candidates for  $\bar{\alpha}$  before settling on a good one. However, his method for moving from one guess to another, based on learning ideas, has unknown convergence properties. He indicates that the reliable parameterizations of his iteration scheme are slow. Judd (1991) shows that Marcet's scheme will often have explosive oscillations, particularly as one attempts to use more flexible approximations. This is because Marcet's iterative scheme is essentially a hog cycle in a Banach space. While the learning ideas behind his iterative scheme may have some intuitive appeal, there is no numerical advantage for using them. Since there are many algorithms for solving nonlinear problems which are quadratic in convergence, it would seem superior to take a nonlinear equations approach in Step 5.

den Haan and Marcet used Marcet's method to solve the stochastic growth model we solved above. He reports taking about three minutes on a Compaq 386/25 to solve for a solution with three free parameters. By comparison, the MWR program takes .3 seconds to solve for a four-parameter policy function on a Compaq 386/20, a roughly 500-fold improvement.

den Haan and Marcet (1990) have suggested a statistical test for accuracy. Their idea is to simulate the model and subject the resulting synthetic data to GMM empirical tests of the Euler equation. After doing this for several such synthetic data sets, they then ask if the resulting distribution of various test statistics differ significantly from what they would be if the data were generated by the true policy function. While the den Haan and Marcet procedure is intuitive, it is unclear how powerful and reliable it is and is much less direct than the method we used above. First, Tauchen (1986) has shown that GMM procedures suffer from finite sample bias. Therefore, even if the approximation is good, it may fail the den Haan and Marcet test. Second, Klenow (1991) has found that the den Haan-Marcet

procedure failed to reject candidate solutions which had 1-2% errors in consumption. He found that the den Haan-Marcet test was fooled by consumption errors which were large but had no discernible pattern. Using the direct procedure described above would immediately uncover these optimization errors.

Coleman used a solution method very similar to Bizer and Judd, except that he used an inefficient basis. He divided  $(\log k, \log \theta)$  space into equi-sized rectangles and, to evaluate the policy function between these nodes, uses bilinear interpolation; that is, he assumed that the policy function is a linear combination of 1,  $\log k$ ,  $\log \theta$ , and  $\log k \log \theta$ . This method is a finite element method, but the more standard finite element scheme for two-dimensional problems is to use piecewise planar functions over triangles (see Burnett). The use of bilinear interpolation on rectangles implicitly assumes a basis of two-dimensional functions formed by taking the tensor product of tent functions in each dimension. While this fits into our scheme, it is a poor choice of basis functions for this problem since all such functions are "saddles", that is, they are convex (in terms of the logs) along one diagonal and concave in the other. The policy function is forced to have a golf ball-like texture where the global nature of the policy function's curvature is often the opposite of the local curvature. There is no economic reason for restricting the policy function to have this pockmarked shape. Of course, one can overcome this restriction by taking sufficiently small rectangles, but that will slow down the solution. Indeed, Coleman reports using a grid of 50 capital stocks and 20 productivity levels for a total of 1000 free parameters. Such a large number of free parameters will make for a very time-consuming computation; Coleman reports using nearly three minutes on a 38 MIPS machine. This would correspond to two to three hours on a 20 MHz personal computer, about five times slower than Marcet. On the other hand, Coleman is attempting to find a good approximation over a wider range of states than Marcet.

In his approach to solving the discrete-time stochastic growth model, Tauchen (1990) discretized  $(k, \theta)$  space, choosing the 90 capital stocks so that  $\log k$  is uniformly spaced and the  $\theta$  to fit a twenty-point Gauss-Hermite quadrature rule, for a total of 1800 points, and then treated the problem as a dynamic programming problem on a discrete space. While this may appear to be similar to our MWR procedure above, note the difference in how the  $\theta$  values

were chosen. In (4.7), there are two places where we chose values for  $\theta$ . We first chose points at which  $R(k, \theta; \bar{x})$  will be evaluated. These choices were governed by the norm which we were using, and in our case were given by using Gauss-Chebyshev quadrature points. Then, given such a  $\theta$ , we evaluated the conditional expectation in (4.6), which requires Gauss-Hermite choices for  $\theta'$ . The fact that the two integrals should use different quadrature rules yields a mixed set of  $\theta$  values which are eventually used, but a set which is motivated by standard quadrature considerations. The other crucial difference between our procedure and Tauchen (1990) is that we do not reduce the problem to a finite state problem.

It should be noted, however, that Tauchen's excellent (1986) numerical study of Lucas (1978) used standard linear integral equation methods, which happen to be equivalent to a MWR collocation technique. Therefore, the method used above is the generalization of Tauchen (1986) to optimal growth problems.

Because of the lack of reported details, I cannot compare my results to all the methods discussed in Taylor and Uhlig. For example, Baxter et al. did not report running times, but since their approach was the same as in Bizer and Judd, their times are presumably similar. The backsolving approach of Sims (1990) and Ingram (1990) does not appear to be comparable to the exercise above since their objective is to find the disturbance which will make the policy rule fit some functional form. The linear-quadratic approximation of McGratton (1990) was fast to compute, but did not appear to be a good approximation globally.

## 6. Convergence Properties of Galerkin Methods

When using numerical procedures, it is desirable to know something concerning the error of the solution. An important focus of theoretical numerical analysis is the derivation of bounds on errors. Two kinds of error results are desirable. First, it is desirable to derive an upper bound on the error for a given level of approximation. Second, if such upper bounds are not possible, it may still be valuable to know that the error goes to zero asymptotically, that is, as one lets the degree of approximation become arbitrarily large. For most methods, the first kind of error information is rarely available. More typical in numerical algorithms for differential equations are asymptotic results. There has been little work on proving that the



algorithms used by economists are asymptotically valid. Miranda (1977) has made progress in this direction with finite-element methods in some models.

Fortunately, there are general theorems concerning the consistency of the Galerkin method. Recall that the Galerkin takes projections of the residual function against the basis elements, and the integrals are theoretically exact. Zeidler (1989) proves consistency for the Galerkin method assuming that the nonlinear operator  $N$  is monotone, coercive, and satisfies a growth condition. Galerkin methods are quite natural for computational purposes since a common theoretical way to prove the existence of a solution to an operator is to prove the existence of a solution to an infinite collection of projection conditions. In fact, Zeidler shows that if these conditions are satisfied one simultaneously proves the existence of a (weak) solution and the consistency of the Galerkin method. Similarly, using degree theory, Krasnosel'skii and Zabreiko (1984) demonstrate consistency for a more general set of projection methods (possibly including Galerkin methods which use numerical quadrature).

I am not claiming that I have shown that the operators used above satisfy the sufficient conditions discussed in Zeidler, Krasnosel'skii and Zabreiko, and elsewhere. Even though it remains to be seen whether these theorems do cover our problems, they do indicate that MWR methods are potentially valid for our economic problems. They also point us in potential fruitful directions for proving both existence results for the underlying operators, and consistency results for alternative solution methods.

## 12. Comments on the Literature and Conclusions

Unfortunately, there are no numerical analysis books which are written for economists. In order to learn about standard numerical methods, one must wade through a literature which is written for physicists, chemists, and engineers. This is particularly if one wants to learn about nonlinear methods since they tend to be problem specific. To help the reader, I have listed a fair number of references on the methods discussed above plus some which have material which will prove useful in economics. The Fletcher books (1984, 1988) proved the most useful for me in studying MWR techniques. There are many other useful techniques used in MWR approaches; Judd (1990) discusses more of them.

We have shown that a general class of techniques from the numerical partial differential

equations literature can be usefully applied and adapted to solve nonlinear economic problems. These problems include simple ones related to growth and equilibrium with concave technologies, as well as more complicated ones related to increasing returns to scale and strategic interactions in oligopoly theory. Since dynamic economic problems can generally be formulated as solutions to differential equations, it is clear that the examples discussed above are just a tiny fraction of those for which MWR methods will be useful.

## References

- Atkinson, Kendall. *An Introduction to Numerical Analysis* (Second Edition), John Wiley and Sons (1989).
- Baxter, Marianne, Mario J. Crucini, and K. Geert Rouwenhorst, "Solving The Stochastic Growth Models by a Discrete-State-Space, Euler-Equation Approach", *Journal of Business and Economic Statistics*, 8 (No. 1, January 1990): 19-21.
- Alain Bensoussan, *Perturbation Methods in Optimal Control*, John Wiley and Sons: 1988.
- Bizer, David S., and Kenneth L. Judd. "Uncertainty and Taxation," *American Economic Review* (May, 1989).
- Brock, William A., and Leonard J. Mirman. "Optimal Economic Growth and Uncertainty: The Discounted Case," *Journal of Economic Theory* 4 (1972), 479-513.
- Burnett, David S. *Finite Element Analysis* Addison-Wesley: Reading, Massachusetts, 1987.
- Christiano, Lawrence J., "Solving The Stochastic Growth Model by Linear-Quadratic Approximation and by Value-Function Iteration", *Journal of Business and Economic Statistics*, 8 (No. 1, January 1990):23-26.
- Coleman, Wilbur John, II, "Solving The Stochastic Growth Model by Policy Function Iteration", *Journal of Business and Economic Statistics*, 8 (No. 1, January 1990):27-29.
- Danthine, Jean-Pierre, John B. Donaldson, and Rajnish Mehra, "On Some Computational Aspects of Equilibrium Business Cycle Theory", *Journal of Economic Dynamics and Control* 13 (No. 3, July, 1989): 449-470.
- Danthine, Jean-Pierre, and John B. Donaldson. "Stochastic Properties of Fast vs. Slow Growing Economies". *Econometrica* 49 (1981), 1007-33.
- Fletcher, C.A.J. *Computational Galerkin Techniques* Springer-Verlog: New York, 1984.
- Fletcher, C.A.J., *Computational Techniques for Fluid Dynamics: Vol. I, II*, Springer-Verlag: Berlin, 1988.
- Gill, Phillip, E., Walter Murray, Michael A. Saunders, and Margaret H. White (1983), "User's Guide for SOL/NPSOL: A Fortran Package for Nonlinear Programming," Technical Report Number SOL 83-12, Systems Optimization Laboratory, Department of Operations Research, Stanford University.

- Gustafson, Robert L., "Carryover Levels for Grains: A Method for Determining Amounts That Are Optimal under Specified Conditions." USDA Technical Bulletin 1178 (1958).
- den Haan, Wouter J, and Albert Marcet, "Solving The Stochastic Growth Model by Parameterizing Expectations", *Journal of Business and Economic Statistics*, 8 (No. 1, January 1990): 31-34.
- Judd, Kenneth L. "Numerical Methods in Economics", Hoover Institution, 1991.
- Judd, Kenneth L. "Asymptotic Methods in Dynamic Economic Models", mimeo, Hoover Institution (October, 1990).
- Judd, Kenneth L. "Minimum Weighted Residual Methods for Solving Dynamic Economic Models," Hoover Institution, (1990).
- Klenow, Peter J. "Externalities and Business Cycles," Stanford University (1991).
- Krasnosel'skii, Mark A., and Zabreiko, Petrovic. *Geometrical Methods of Nonlinear Analysis*. Springer-Verlag (1984).
- Marcet, Albert. "Solving Nonlinear Stochastic Models by Parametrizing Expectations", Carnegie Mellon University, mimeo (1989).
- McGratton, Ellen R., "Solving The Stochastic Growth Model by Linear-Quadratic Approximation", *Journal of Business and Economic Statistics*, 8 (No. 1, January 1990):41-43.
- Miranda, Mario J. "A Computational Rational Expectations Model for Agricultural Price Stabilization Programs". University of Connecticut, (1987).
- Miranda, Mario J., and Peter G. Helmburger. "The Effects of Commodity Price Stabilization Programs." *American Economic Review* (March, 1988): 46-58.
- Nürnbergger, Günther. *Approximation by Spline Functions*, Springer-Verlog: Berlin (1989).
- Prenter, P.M. *Splines and Variational Methods*, John Wiley & Sons: New York (1989).
- Prescott, Edward C., and Rajnish Mehra. "Recursive Competitive Equilibrium: The Case of Homogeneous Agents," *Econometrica* 48 (1980): 1365-1379.
- Sims, Christopher A., "Solving The Stochastic Growth Model by Backsolving with a Particular Nonlinear Form for the Decision Rule", *Journal of Business and Economic Statistics*, 8 (No. 1, January 1990):45-47.
- Tauchen, George, "Quadrature-Based Methods for Obtaining Approximate Solutions to the

- Integral Equations of Nonlinear Rational Expectations Models", Duke University mimeo, (1986).
- Tauchen, George, "Solving The Stochastic Growth Model by Using Quadrature Methods and Value-Function Iterations", *Journal of Business and Economic Statistics*, 8 (No. 1, January 1990):49-51.
- Taylor, John B., and Harald Uhlig, "Solving Nonlinear Stochastic Growth Models: A Comparison of Alternative Solution Methods", *Journal of Business and Economic Statistics*, 8 (No. 1, January 1990): 1-18.
- Williams, Jeffrey, and Brian Wright. *Storage and Commodity Markets*, Cambridge University Press, in press (January 1991).
- Williams, Jeffrey, and Brian Wright. "The Economic Role of Commodity Storage," *Economic Journal* 92 (September 1982): 596-614.
- Williams, Jeffrey, and Brian Wright. "The Roles of Public and Private Storage in Managing Oil Import Disruptions," *Bell Journal of Economics* 13 (Autumn 1982): 341-353.
- Zeidler, Eberhard. *Nonlinear Functional Analysis and Its Applications: Volumes I*. New York: Springer-Verlag, 1986.
- Zeidler, Eberhard. *Nonlinear Functional Analysis: Volume II*. New York: Springer-Verlag, 1989.

Table 1

	CAP	PROD	CONS	$n = 20$	$n = 10$	$n = 7$	$n = 4$	$n = 2$
$\gamma = -0.500$	0.50	0.1253211	0.0911211	0.01	0.01	0.01	-0.02	-2.14
	0.60	0.1331736	0.1050036	0.00	0.00	0.00	0.07	5.42
	0.70	0.1401954	0.1185654	-0.03	-0.03	-0.02	-0.14	9.77
	0.80	0.1465765	0.1318765	0.00	0.00	0.00	-0.27	11.61
	0.90	0.1524457	0.1449757	0.01	0.01	0.01	-0.27	11.33
	1.00	0.1578947	0.1578947	0.00	0.00	0.00	-0.14	9.25
	1.10	0.1629916	0.1706616	-0.01	-0.01	-0.01	0.08	5.65
	1.20	0.1677882	0.1832982	0.02	0.02	0.02	0.28	0.75
	1.30	0.1723252	0.1958152	0.02	0.02	0.02	0.24	-5.36
$\gamma = -0.900$	0.50	0.1253211	0.1010611	0.01	0.05	0.05	0.02	-4.78
	0.60	0.1331736	0.1132936	-0.22	0.01	0.01	0.17	7.71
	0.70	0.1401954	0.1250054	-0.25	0.03	0.03	-0.12	14.99
	0.80	0.1465765	0.1362965	-0.10	0.04	0.04	-0.39	18.06
	0.90	0.1524457	0.1472357	0.10	-0.03	-0.03	-0.49	17.62
	1.00	0.1578947	0.1578947	0.35	0.00	-0.01	-0.23	14.37
	1.10	0.1629916	0.1683016	0.40	-0.02	-0.02	0.12	8.60
	1.20	0.1677882	0.1784982	0.32	0.02	0.02	0.45	0.73
	1.30	0.1723252	0.1884952	0.07	-0.04	-0.04	0.31	-9.13
$\gamma = -3.000$	0.50	0.1253211	0.1147611	0.02	0.03	0.03	0.01	-11.82
	0.60	0.1331736	0.1245336	-0.16	0.04	0.04	0.39	5.11
	0.70	0.1401954	0.1335954	-0.34	-0.03	-0.03	-0.10	14.94
	0.80	0.1465765	0.1421165	-0.20	-0.02	-0.01	-0.48	19.36
	0.90	0.1524457	0.1501957	0.11	0.04	0.04	-0.49	19.37
	1.00	0.1578947	0.1578947	0.29	0.00	-0.01	-0.26	15.57
	1.10	0.1629916	0.1652816	0.34	-0.02	-0.02	0.18	8.65
	1.20	0.1677882	0.1723982	0.23	-0.03	-0.03	0.51	-0.98
	1.30	0.1723252	0.1792852	0.11	0.02	0.02	0.41	-12.90
$\gamma = -10.000$	0.50	0.1253211	0.1214511	0.14	-0.02	-0.02	-0.24	-19.27
	0.60	0.1331736	0.1299836	0.60	-0.03	-0.04	0.39	-0.26
	0.70	0.1401954	0.1377454	0.51	-0.05	-0.05	0.02	11.05
	0.80	0.1465765	0.1449165	0.30	0.00	0.01	-0.35	16.45
	0.90	0.1524457	0.1516057	0.01	0.03	0.02	-0.43	17.03
	1.00	0.1578947	0.1578947	-0.37	0.00	-0.01	-0.21	13.61
	1.10	0.1629916	0.1638516	-0.72	-0.02	-0.02	0.20	6.87
	1.20	0.1677882	0.1695282	-0.89	0.00	0.00	0.51	-2.68
	1.30	0.1723252	0.1749552	-0.59	-0.01	-0.01	0.22	-14.72

Table 2

$\gamma$	$\rho$	$\sigma$	$n_k = 2 \quad n_\theta = 2 \quad m_k = 2 \quad m_\theta = 2$					$n_k = 4 \quad n_\theta = 3 \quad m_k = 4 \quad m_\theta = 3$				
			log <sub>10</sub> of:					log <sub>10</sub> of:				
			$\ E\ _\infty$	$\ E\ _2$	$\ E\ _1$	$\ E_I\ _\infty$	$\ E_I\ _2$	$\ E\ _\infty$	$\ E\ _2$	$\ E\ _1$	$\ E_I\ _\infty$	$\ E_I\ _2$
-15.00	0.80	0.01	-2.13	-2.71	-2.80	-2.58	-2.63	-3.00	-3.66	-3.83	-3.70	-3.96
-15.00	0.80	0.04	-1.89	-2.47	-2.54	-2.28	-2.36	-2.44	-2.80	-2.87	-2.59	-2.78
-15.00	0.30	0.01	-2.20	-2.72	-2.82	-2.63	-2.67	-3.05	-3.68	-3.86	-3.82	-3.97
-15.00	0.30	0.04	-2.13	-2.71	-2.80	-2.58	-2.64	-2.97	-3.65	-3.83	-3.70	-3.95
-0.90	0.80	0.01	-1.63	-2.28	-2.42	-2.26	-2.31	-2.65	-3.32	-3.50	-3.44	-3.62
-0.90	0.80	0.04	-1.48	-2.17	-2.31	-2.09	-2.18	-2.46	-2.96	-3.04	-2.79	-3.05
-0.90	0.30	0.01	-1.68	-2.28	-2.43	-2.29	-2.33	-2.69	-3.33	-3.51	-3.49	-3.62
-0.90	0.30	0.04	-1.64	-2.28	-2.42	-2.26	-2.31	-2.65	-3.32	-3.50	-3.44	-3.62
-0.10	0.80	0.01	-0.01	-0.79	-1.22	-1.34	-1.40	-1.68	-2.39	-2.65	-2.70	-2.86
-0.10	0.80	0.04	0.01	-0.81	-1.19	-1.20	-1.27	-1.48	-2.12	-2.22	-1.89	-2.18
-0.10	0.30	0.01	0.04	-0.79	-1.22	-1.36	-1.41	-1.67	-2.39	-2.65	-2.74	-2.86
-0.10	0.30	0.04	0.18	-0.78	-1.22	-1.35	-1.41	-1.63	-2.40	-2.65	-2.74	-2.86
			$n_k = 7 \quad n_\theta = 5 \quad m_k = 7 \quad m_\theta = 5$					$n_k = 7 \quad n_\theta = 5 \quad m_k = 20 \quad m_\theta = 12$				
-15.00	0.80	0.01	-4.28	-4.99	-5.19	-5.00	-5.32	-4.43	-5.05	-5.18	-4.91	-5.21
-15.00	0.80	0.04	-3.36	-3.88	-4.00	-3.70	-3.90	-3.30	-3.87	-3.95	-3.67	-3.86
-15.00	0.30	0.01	-4.37	-5.03	-5.23	-5.10	-5.35	-4.55	-5.11	-5.22	-4.99	-5.23
-15.00	0.30	0.04	-4.24	-4.97	-5.19	-4.96	-5.31	-4.38	-5.03	-5.18	-4.87	-5.20
-0.90	0.80	0.01	-4.00	-4.67	-4.86	-4.73	-4.98	-4.14	-4.73	-4.86	-4.65	-4.89
-0.90	0.80	0.04	-3.44	-4.01	-4.12	-3.84	-4.10	-3.54	-4.03	-4.11	-3.83	-4.05
-0.90	0.30	0.01	-4.06	-4.68	-4.87	-4.77	-4.98	-4.20	-4.74	-4.87	-4.69	-4.90
-0.90	0.30	0.04	-4.00	-4.67	-4.86	-4.73	-4.98	-4.14	-4.73	-4.86	-4.65	-4.89
-0.10	0.80	0.01	-3.40	-4.11	-4.37	-4.35	-4.56	-3.47	-4.16	-4.39	-4.32	-4.54
-0.10	0.80	0.04	-2.50	-3.13	-3.22	-2.93	-3.21	-2.60	-3.10	-3.17	-2.91	-3.15
-0.10	0.30	0.01	-3.44	-4.11	-4.36	-4.36	-4.56	-3.51	-4.16	-4.39	-4.33	-4.54
-0.10	0.30	0.04	-3.43	-4.11	-4.37	-4.36	-4.56	-3.49	-4.16	-4.39	-4.33	-4.54
			$n_k = 10 \quad n_\theta = 6 \quad m_k = 10 \quad m_\theta = 6$					$n_k = 10 \quad n_\theta = 6 \quad m_k = 25 \quad m_\theta = 15$				
-15.00	0.80	0.01	-5.48	-6.22	-6.43	-6.19	-6.55	-5.61	-6.28	-6.42	-6.11	-6.44
-15.00	0.80	0.04	-3.81	-4.29	-4.38	-4.11	-4.36	-3.88	-4.30	-4.37	-4.11	-4.37
-15.00	0.30	0.01	-5.66	-6.30	-6.49	-6.31	-6.59	-5.80	-6.38	-6.49	-6.24	-6.48
-15.00	0.30	0.04	-5.45	-6.20	-6.43	-6.15	-6.52	-5.57	-6.26	-6.42	-6.08	-6.43
-0.90	0.80	0.01	-5.16	-5.92	-6.13	-5.97	-6.23	-5.22	-5.99	-6.13	-5.90	-6.14
-0.90	0.80	0.04	-3.91	-4.48	-4.59	-4.26	-4.51	-4.00	-4.49	-4.57	-4.26	-4.49
-0.90	0.30	0.01	-5.32	-5.94	-6.14	-6.01	-6.23	-5.47	-6.01	-6.14	-5.95	-6.15
-0.90	0.30	0.04	-5.27	-5.93	-6.13	-5.96	-6.21	-5.41	-6.00	-6.12	-5.89	-6.14
-0.10	0.80	0.01	-5.09	-5.87	-6.12	-5.94	-6.27	-5.17	-5.92	-6.15	-5.94	-6.26
-0.10	0.80	0.04	-2.99	-3.59	-3.68	-3.37	-3.64	-3.09	-3.56	-3.64	-3.38	-3.61
-0.10	0.30	0.01	-5.22	-5.88	-6.12	-6.04	-6.27	-5.28	-5.92	-6.14	-6.02	-6.26
-0.10	0.30	0.04	-5.17	-5.87	-6.12	-6.01	-6.27	-5.23	-5.91	-6.14	-5.99	-6.26