# Numerical Dynamic Programming in Economics

Hans Amman

*University of Amsterdam*

John Rust

*University of Wisconsin*

**Contents** [1]



---

[1] March, 1994 draft for *Handbook of Computational Economics*, H. Amman, D. Kendrick and J. Rust, (eds.). We are grateful for helpful discussions with Sudhakar Acath, Masanao Aoki, Tamer Basar, Willem Buiter, Ray Fair, Steven Hall, David Kendrick, James Kuelbs, Finn Kydland, Ken Judd, Berç Rustem, Mark Salmon, Bo Thio, Joseph Traub, and John Tsitsiklis. Note: this is a preliminary and incomplete: do not quote without permission of authors.

## 1. Introduction

Dynamic Programming (DP) is a central tool in economics because it allows us to formulate and solve a wide class of sequential decision-making problems under uncertainty. Many economic problems can be formulated as Markov decision processes (MDP's) in which a decision maker who is in state $s_t$ at time $t = 1, \ldots, T$ takes a decision $d_t$ that determines current utility $u(s_t, d_t)$ and affects the distribution of next period's state $s_{t+1}$ via a Markov transition probability $p(s_{t+1}|s_t, d_t)$. The problem is to determine an *optimal decision rule* $\delta$ that solves $V(s) \equiv max_\delta E_\delta \left\{ \sum_{t=0}^{T} \beta^t u(s_t, d_t) | s_0 = s \right\}$ where $E_\delta$ denotes expectation with respect to the controlled stochastic process $\{s_t, d_t\}$ induced by the decision rule $\delta \equiv \{\delta_1, \ldots, \delta_T\}$, and $\beta \in (0,1)$ denotes the discount factor. What makes the problem difficult is that instead of computing a sequence of optimal decisions $\{d_0, \ldots, d_T\}$ that are fixed *ex ante,* the optimal decision rule consists of a sequence of *functions* $\{\delta_1, \ldots, \delta_T\}$ that allow the *ex post* decision $d_t$ to vary as a best response to the realized history of the process $H_t$, i.e. $d_t = \delta_t(H_t)$. The method of *dynamic programming* (a term coined by Richard Bellman in his 1957 text) provides a constructive, recursive procedure for computing $\delta$ using the *value function* $V$ as a "shadow price" to decentralize a complicated stochastic/multiperiod optimization problem into a sequence of simpler deterministic/static optimization problems. [1] We provide a brief review of the main theoretical results about MDP's in section 2. We show that stationary, infinite horizon MDP's can be viewed as multidimensional generalizations of "geometric series" whose solutions are mathematically equivalent to computing the fixed point to a particular contraction mapping known as *Bellman's equation.*

Unfortunately, it is quite rare that one can explicitly solve Bellman's equation and derive analytical or "closed-form" solutions for the optimal decision rule $\delta$ or value function $V$. As a result, most interesting DP problems must be solved numerically on digital computers. From the standpoint of computation, there is an important distinction between *discrete MDP's* where the state and control variables can assume only a finite number of possible values versus *continuous MDP's* where the state or control variables can assume a continuum of possible values. Discrete MDP problems can be solved exactly (modulo rounding error in arithmetic operations), whereas the solutions to continuous MDP problems can only be approximated to within some solution tolerance $\epsilon$. Approximate solution methods may also be attractive for solving discrete MDP's that have a very large number $S$ of possible values for the state variables, or a large number $D$ of possible values of the control variables.

There are two basic types of approximate solution methods: 1) discrete approximation, and 2) parametric approximation. Discrete methods solve a finite state MDP problem that approximates the original continuous MDP problem. Since the methods for solving discrete MDP's have been well developed and exposited in the operations research literature (e.g. Bertsekas, 1987, Porteus, 1980 and Puterman, 1990), this chapter will only briefly review

[1] In finite horizon problems $V$ actually denotes an entire sequence of value functions, $V \equiv \{V_0^T, \ldots, V_T^T\}$, just as $\delta$ denotes a sequence of decision rules. In the stationary infinite-horizon case, the solution $(V, \delta)$ reduces to a pair of functions of the current state $s$.

the latest developments in this area and focus instead on evaluating the ability of these methods to solve continuous MDP problems. Parametric approximation methods treat either $V$ or $\delta$ or both as flexible parametric functions of a finite-dimensional parameter vector $\alpha$, and generally require a nonlinear optimization subroutine to find a value $\alpha^*$ such that the resulting parametric approximation $(V_{\alpha^*}, \delta_{\alpha^*})$ "best fits" the true solution $(V, \delta)$ in an appropriate sense (e.g. minimum mean squared error). However another possibility is to use a nonlinear equation solver to choose $\alpha^*$ to satisfy certain "orthogonality conditions". Thus the parametric approximation methods considered in this chapter fall in the general class of *projection methods* described in chapter 6 by Judd. We also present hybrid methods that avoid the need for optimization or solution of nonlinear equations that is required by most parametric approximation methods. These latter methods use kernel-based "smoothers" to compute a parametric approximation $(V_\alpha, \delta_\alpha)$ using elementary algebraic operations and a potentially noisy set of observations of the true solution $(V, \delta)$ at selected points in the domain.

Approximate solution methods present us with a tradeoff between the desired precision $\epsilon$ of the numerical solution and the amount of computer time (and storage space) needed to compute it. Solution time will also be an increasing function of any relevant measure of the size or *dimension* $k$ of an MDP problem. In general, economists are interested in using the most efficient possible algorithm for any specified values of $(\epsilon, k)$, especially since in many economic problems the MDP solution algorithm is embedded or "nested" as a subroutine inside a large optimization or equilibrium problem. Examples include computing competitive equilibria of stochastic general equilibrium models with incomplete markets (Hansen and Sargent, 1993, Imrohoroglu and Imrohoroglu, 1993, McGuire and Pakes, 1993), maximum likelihood estimation of unknown parameters of $u$ and $p$ using data on observed states and decisions of actual decision-makers (Eckstein and Wolpin, 1987, Rust, 1994 and Sargent, 1979), and computation and econometric estimation of Bayesian Nash equilibria of dynamic games (McKelvey and Palfrey, 1992). All of these problems are solved by "polyalgorithms" that contain MDP solution algorithms as subroutines. The MDP subroutines are themselves polyalgorithms consisting of individual subroutines for numerical integration, optimization, approximation, and solution of systems of linear and nonlinear equations. Since the MDP problem must be repeatedly re-solved for various trial sequences of $\{\beta, u, p\}$, speed, accuracy, and numerical stability are critical.

A great many alternative approximate and exact solution methods have been proposed in the last 30 years. Bellman contributed to the development of both discrete and parametric approximation methods for solving MDP's (e.g. policy iteration, Bellman 1957, and polynomial approximation, Bellman *et. al.* 1965). Recently there has been considerable controversy in economics about the relative merits of discrete versus parametric approximation methods for solving continuous MDP problems. Part of the controversy arose from a "horse race" in the 1990 *Journal of Business and Economic Statistics* (Taylor and Uhlig, 1990) in which a number of alternative solution methods competed in their ability to solve the classical Brock-Mirman stochastic growth model described in section 2.5. More recently Judd 1993 has claimed that

> Approximating continuous-state problems with finite state Markov chains limits the range of problems which can be analyzed. Fortunately, state-space discretization is unnecessary. For the past thirty years, the standard procedure in Operations Research literature (see Bellman, 1963, Dantzig 1974, Daniel, 1976) has been to approximate value functions and policy rules over continuous state spaces with orthogonal polynomials, splines, or other suitable families of functions. This results in far faster algorithms and avoids the errors associated with making the problem unrealistically "lumpy". (p. 3)

This chapter attempts to offer some new perspectives on this debate by comparing discrete and parametric approximation approaches both theoretically and practically. Although there is no simple answer to the question of which approach is best we do provide a conceptual framework and a set of numerical comparisons that we hope will help individual researchers decide which of the myriad of available methods is best for solving their particular problems.

Our distinction between discrete and parametric approximation methods is somewhat oversimplified: discretization can be viewed as a particular kind of parametric approximation and virtually any parametric approximation method requires discrete approximation in order to compute various integrals such as those appearing in the conditional expectation operator $E_\delta$. Although discrete approximation methods ordinarily generate solutions $\hat{V}$ lying in a finite dimensional Euclidean space $R^N$ and parametric methods produce an approximate solution $V_{\alpha^*}$ directly in the infinite dimensional space $B$ where the true solution $V$ lives, this too is an artificial distinction since a discretized solution $\hat{V} \in R^N$ can always be mapped into an element of $B$ by interpolating the $N$ coordinates of $\hat{V}$. However these are subtleties: we will see that discretization methods involve a fundamentally different solution philosophy than the parametric methods we will be presenting in section 5.[2] The difference is easiest to see in the case of continuous infinite horizon MDP problems. Parametric methods generate approximate solutions living in an $N$-dimensional submanifold $B_N$ that is typically defined as the image of a smooth mapping $\phi : A_N \rightarrow B_N$ where the set $A_N$ is generally a compact subset of $R^N$: $V_\alpha \equiv \phi(\alpha) \in B_N = \phi(A_N)$. Depending on how we specify the set $A_N$ and the mapping $\phi$ we can obtain a wide variety of different parametric approximation methods. Parametric methods also require us to specify a procedure for determining a particular point $V_{\alpha^*} \in B_N$ that "best approximates" the true solution $V$. Here we see the crucial distinction between the two approaches: *discrete methods such as successive approximations and policy iteration preserve and fully exploit the fact that $V$ is the fixed point to a contraction mapping $V = \Gamma(V)$ whereas parametric methods typically transform the fixed point problem into a problem of optimizing some smooth function $g(\alpha)$ or of solving a system of nonlinear equations $g'(\alpha) = 0$ for some functions $g$ and $g'$ that generally do not retain the nice mathematical properties of the Bellman operator* $\Gamma$. Section 5.4 presents an example where $B_N$ is a linear manifold generated by linear combinations of Chebyshev polynomials (i.e. $\phi$ is a linear transformation), and an example where $B_N$ is a nonlinear manifold generated by the output of a single layer neural network with coefficient vector $\alpha$ (i.e. $\phi$ is a smooth nonlinear mapping). In both of these cases $V_\alpha = \phi(\alpha)$ is easy to compute for any given $\alpha \in A_N$: the difficult problem is to find the vector $\alpha^*$ such that the associated value function $V_{\alpha^*}$ approximately

[2] We ignore the issue of approximating the optimal decision rule $\delta$ due to the fact that it can be computed from the approximate solution for $V$ as shown in section 2.2.

solves Bellman's equation. Under the least squares formulation of the approximation problem $\alpha^*$ is defined by the solution to the following minimization problem:

$$\alpha^* = \underset{\alpha \in A}{argmin}\, g(\alpha) \equiv |V_\alpha - \Gamma(V_\alpha)|_2 \equiv \sqrt{\int |V_\alpha(s) - \Gamma(V_\alpha)|^2 \mu(ds)}$$

where and $\mu$ is a probability measure on the continuous state space $S$. Note also that evaluation of $g$ requires computation of multivariate integrals, both in the evaluation of $\Gamma(V_\alpha)$ and in the evaluation of $|V_\alpha - \Gamma(V_\alpha)|_2$. These integrals cannot be computed analytically in most cases, so some sort of numerical integration routine must be employed to repeatedly recalculate them for different trial values of $\alpha$. Also, we will generally require a *global* minimum $\alpha^*$ of $g$ to guarantee that the approximate solution $V_{\alpha^*}$ will be sufficiently close to the true solution $V$, but this may difficult if $g$ is not smooth or has lots of local optima.[3] Discrete methods do not suffer from these problems since global convergence to the true solution is guaranteed by the fact that $\Gamma$ is a contraction mapping. However the hope is that a judicious choice of $\phi$ will allow the parametric approach to provide a very accurate approximation of $V$ using a relatively small number of parameters $N$.[4]

A minimal requirement of any sensible solution method is that it be *consistent:* i.e. by increasing the number $N$ of grid points in a discrete approximation of $\hat{V}$ or the number of $\alpha$ coefficients in a parametric approximation $V_{\alpha^*}$ we ought to be able to approximate the true solution arbitrarily accurately.[5] Consistency theorems for discrete approximation of MDP's date back to Fox 1973 and Bertsekas 1987, but to our knowlege there are no corresponding published results establishing the consistency of parametric methods. [6] Borrowing from standard results on the consistency of non-parametric regression (Härdle, 1987), approximation error bounds for specific classes of functions such as orthogonal polynomials (Rivlin, 1969), error bounds known as *Kolmorgorov N-widths* for more general linear subspaces of functions (Pinkus, 1985), and approximation error bounds for nonlinear manifolds such as neural networks (Hornik, Stinchombe and White, 1993), we provide a general framework for proving the consistency of a variety of parametric approximation methods in sections 5.2 and 5.4.

---

[3] In this vein section 5 discusses the desirability of solving a slightly perturbed version of Bellman's equation with an approximate Bellman operator $\Gamma_\sigma$ that is everywhere differentiable (unlike $\Gamma$ which has kinks at certain points $V \in B$).

[4] In some cases parametric methods allow us to exploit certain types of prior information we might have about the solution $V$, i.e. monotonicity, convexity, etc. For details, see Judd, 1994.

[5] Consistency of parametric methods also depends on being able to find arbitrarily accurate solutions to the associated nonlinear optimization and multivariate integration problems.

[6] There is a closely literature on discretization and parametric approximation of abstract operator equations (Cryer, 1982, Krasnosel'skii *et. al.* 1982, and Rall, 1982) which can be viewed as encompassing the methods outlined in this chapter as a special case. However this literature and the closely related literature on approximation of collectively compact operators (Anselone, 1971 and Anselone and Ansorge, 1981) is rather abstract and it is often easier to prove the convergence of a particular method from first principles than to verify the general sufficient conditions required for these more general consistency results.

Although it's comforting to know that a numerical method is consistent, there is a practical limitation to one's ability to solve continuous MDP's arbitrarily accurately: namely, Bellman's (1957) *curse of dimensionality*. This is the well-known exponential rise in the time and space required to compute the solution to an MDP problem as the number of components (dimensions) $k_s$ of the state variable and the number of components $k_d$ of the control variable increases. Although one typically thinks of the curse of dimensionality as arising from the discretization of continuous MDP's, it also occurs in discrete MDP's that have many state and control variables. Parametric approximation methods are also subject to their own version of the curse of dimensionality: in order to guarantee that a parametric approximation $V_\alpha$ is within an abitrary tolerance $\epsilon$ of the true solution $V$, we must increase the dimension $N_\alpha$ of the parameter vector $\alpha$ at a sufficiently rapid rate as $\epsilon \to 0$. However the amount of computer time required to solve the associated optimization problem or system of nonlinear equations increases exponentially fast as $N_\alpha \to \infty$, at least on a worst-case basis (Nemirovsky and Yudin, 1983).

An important unresolved question is whether we can somehow circumvent the curse of dimensionality through a clever choice of solution algorithm, perhaps for a restricted class of problems exhibiting special structure. For example, Nemirovsky and Yudin 1978 demonstrate that the amount of time required to solve an $N$-dimensional *convex* optimization problem only increases linearly in $N$ on a worst case basis. A number of important developments in theoretical computer science in the last twenty years (see, e.g. Garey and Johnson, 1983, Traub and Woźniakowski, and Nemirovsky and Yudin, 1983) have enabled formal proofs of lower bounds on the *computational complexity* of solving various continuous multivariate mathematical problems such as nonlinear optimization, numerical integration, function approximation, and recently, MDP problems. There are two main branches of complexity theory, corresponding to discrete and continuous problems. Discrete (or algebraic) computational complexity applies to finite problems that can be solved exactly such as matrix multiplication, the travelling salesman problem, linear programming, and discrete MDP's.[7] The *size* of a discrete problem is indexed by an integer $k$ and the (worst case) complexity, $comp(k)$, denotes the minimal number of computer operations necessary to solve the hardest possible problem of size $k$, (or $\infty$ if there is no algorithm capable of solving the problem). Continuous computational complexity theory applies to continuous problems such as multivariate integration, function approximation, nonlinear programming, and continuous MDP problems. None of these problems can be solved exactly, but in each case the true solution can be approximated to within an arbitrarily small tolerance $\epsilon$. Problem size is indexed by an integer $k$ denoting the dimension of the space that the continuous variable variable lives in (typically $R^k$), and the complexity, $comp(\epsilon, k)$, is defined as the minimal computational cost of solving the hardest possible $k$-dimensional problem to within a tolerance of $\epsilon$. Complexity theory also provides a simple way of formalizing the curse of dimensionality: we say that a discrete MDP problem is in the class $CD$ if $comp(k) \geq O(N^k)$ for some integer $N > 1$, i.e. if its complexity grows exponentially fast as the dimension $k$ increases. Similarly a continuous MDP problem is in the class $CD$ if $comp(\epsilon, k) \geq O(1/\epsilon^k)$. In

[7] In the latter two problems we abstract from rounding error computer arithmetic.

the computer science literature problems in the class $CD$ are called *intractable.*[8] On the other hand, if complexity is bounded by a polynomial in $k$ we say that the MDP problem is in the class $P$ of polynomial-time problems. Computer scientists refer to polynomial-time problems as *tractable.*[9] Since a rudimentary knowledge of some basic results of complexity theory can provide important guidance into the design of practical solution methods, we provide a brief, self-contained review of the relevant terminology and results from discrete and continuous computational complexity theory in section 3.[10]

The discrete and continuous complexity bounds $comp(k)$ and $comp(\epsilon, k)$ depend on the model of computation used (parallel vs. serial, real vs. Turing), the type of algorithm used (deterministic vs. stochastic), the relevant metric for measuring the error $\epsilon$ in the approximate solution (worst case vs. average case complexity), and the class of problems being considered (i.e. general MDP's versus restricted subclasses where $u$ and $p$ satisfy certain properties). The main results can be summarized as follows. Discrete MDP's with $S$ states and $D$ decisions can be solved exactly in polynomial time using a variety of algorithms.[11] Certain subclasses of continuous MDP's such as linear quadratic MDP's can also be solved in polynomial time using the algebraic Ricatti equation methods outlined in section 4. In section 5.1 we show that an upper bound on the complexity of solving a discrete MDP with $S$ states and $D$ decisions is $cTDS^2$ where $c$ is the time cost per arithmetic operation. One can do substantially better than this by using fast matrix multiplication algorithms of Pan 1983 and others. However the lower limit on the complexity of matrix multiplication is not known at the present time, so it is difficult to put a lower bound on the complexity of solving finite MDP's. Solutions to MDP's can also be speeded up by using massive parallel processors such as the Connection Machine. Although the dynamic programming algorithm is inherently sequential, the main work involved in solving finite horizon problems is the computation of the conditional expectation of next period's value function for each possible value of the current state $s$. This can be carried out efficiently in parallel, say, with $S$ separate processors computing the conditional expectations for each state $s = 1, \ldots, S$. However, Papadimitriou and Tsitsiklis 1987 argued that infinite horizon discrete MDP's cannot be efficiently parallelized by showing that the problem is *P-complete* which

---

[8] We prefer the terminology "curse of dimensionality" since the common use of the term "intractable" connotes a problem that can't be solved. Computer scientists have a specific terminology for problems that can't be solved in any finite amount time: these problems have infinite complexity, and are classified as *non-computable*. However even though intractable problems are computable problems in the computer science terminology, as the problem grows large the lower bound on the solution time grows so quickly that these problems are not computable in any practical sense.

[9] Here again it is important to note the difference between the common meaning of the term "tractable" and the computer science definition. Even so-called "tractable" polynomial-time problems can quickly become computationally infeasible if complexity satisfies $comp(k) \geq O(k^b)$ for some large exponent $b$. However it seems to be a fortunate act of nature that the maximum exponent $b$ for most common polynomial time problems is fairly small; typically $b \in [2, 3]$.

[10] For a recent survey of applications of complexity theory to other economic problems see Norman, 1993. Tsitsiklis 1993 provides a review of applications of complexity theory to MDP's. Traub 1993 presents a review of recent developments in complexity theory for solving more general continuous problems such as multivariate integration and function approximation.

[11] This result assumes that we index the size of the MDP by $(S, D)$. However, section 5.1 shows that the MDP problem is in the class $CD$ of exponential time problems if we index the size of the MDP by $(k_s, k_d)$ where $k_s$ is the number of state variables and $k_d$ is the number of control variables.

means, roughly speaking, that the MDP problem is at least as difficult as any other polynomial-time problem such as linear programming (in fact the LP approach turns out to be a relatively efficient way to solve MDP's as we will see in section 5.4). Polynomial-time problems that can be effectively parallelized are said to be in the class *NC:* formally, *NC* consists of all problems that can be solved in poly-log time (i.e. a polynomial of $log(S)$, where $S$ is the size of the problem) using an expandable massive parallel processor whose processors equal some polynomial of $S$. The P-completeness of the MDP problem "means, intuitively, that such problems are as hard as any problem in $P$; thus, they can be massively parallelized only if $NC = P$, that is if *all* problems in $P$ can, and thus there are no inherently sequential problems. Since this event is considered unlikely, $P$-completeness is taken as evidence that a problem is not *NC*, and thus cannot be satisfactorily paralellized." (Papadimitriou and Tsitsiklis, 1987, p.442). We argue that this result may be an artifact of the use of the Turing model of computation and the algebraic notion of computational complexity. In section 5.3 we show that the MDP problem can be effectively parallelized by presenting a massively parallel policy iteration algorithm that uses the parallel Newton algorithm of Pan and Reif 1985 to approximately solve a system of $S$ linear equations in $S$ unknowns in $O(log(S)^2)$ time using $S^\omega$ processors where $\omega \leq 2.376$ is the best current lower bound on the complexity of matrix multiplication.

Complexity bounds for continuous MDP's are of substantial interest in view of a recent paper by Chow and Tsitsiklis 1991 that proves that a "one way multigrid" discretization algorithm is "approximately optimal." They do this by showing that the multigrid algorithm (which involves solving a discretized version of the MDP by successive approximations using a sequence of successively finer grids) asymptotically attains the lower bound on complexity. This worst case complexity bound was previously established by Chow and Tsitsiklis 1989 and is bounded by

$$\frac{c_1}{\left((1-\beta)^2)\epsilon\right)^{2k_s+k_d}} \leq comp(\epsilon, k_s, k_d, \beta) \leq \frac{c_2}{\left((1-\beta)^2\epsilon\right)^{2k_s+k_d}},$$

for some constants $0 \leq c_1 \leq c_2$. It is an open question whether parametric approximation methods (such as polynomial approximations of the value function) also attain this complexity bound. Some promising recent results of Barron 1992 and Hornik, Stinchombe and White 1993 show that neural networks break the curse of dimensionality problem in the sense that the error in approximating certain families of smooth functions by a single layer feedforward neural network with $N$ connection weight coefficients $\alpha$ decreases at rate $1/\sqrt{N}$ independently of the number of variables $k$ entering the function. However the achilles heel of this approach is the computational complexity of finding an $N$-dimensional vector $\alpha$ that approximately minimizes the distance between the output of the neural net and the function being approximated. We conclude that once we move outside of the class of LQ-MDP's (which are solvable in polynomial time) one cannot avoid the curse of dimensionality for a reasonably general class of nonlinear MDP's – at least on a worst case basis.

However it turns out that we can break the curse of dimensionality if we use random instead of deterministic algorithms or measure approximation error and computer runtime on an average instead of a worst case basis. An example is multivariate integration of a function $f$ defined on the $k$-dimensional unit cube $[0,1]^k$ that is $r > 0$ times differentiable. The worst-case (deterministic) complexity of this problem $comp(\epsilon, k) = O(1/\epsilon^m)$ where $m = k/r$, so the problem is in the class *CD*. However consider monte carlo integration of $f$ using random uniform draws from $[0,1]^k$. It is easy to see that the strong law of large numbers and central limit theorem imply that worst case *randomized complexity* of the multivariate integration problem is $comp^{wor\text{-}ran}(\epsilon, k) = O(1/\epsilon^2)$, so the integration problem becomes tractable once randomization is allowed. However randomization does not always succeed in breaking the curse of dimensionality: Nemirovsky and Yudin 1983 showed that randomization doesn't help in solving general multivariate nonlinear programming problems, Traub, Wasilikowski and Woźniakowski 1987 showed that randomization doesn't help in multivariate function approximation, and Werschulz 1992 showed that randomization doesn't help in solving multivariate elliptic partial differential equations or Fredholm integral equations of the second kind.

The other way to break the curse of dimensionality is to redefine the problem and instead of measuring approximation error and computer runtime on a worst case basis, one measures it on an average case basis with respect to some prior probability distribution over the space of possible problem instances. Woźniakowski 1991 and 1992 has shown that a number of linear multivariate problems such as integration and function approximation that are subject to the curse of dimensionality on a worst case basis turn out to be tractable on average. Indeed, Woźniakowski 1991 has shown that the multivariate integration problem is *strongly tractable:* i.e. the average amount of time required to approximately integrate a function over $R^k$ to within a mean square error of $\epsilon$ is *independent* of the dimension $k$. Another practical example is the simplex algorithm for linear programming: although one can construct a sequence of worst case problems for which the simplex method for solving linear programming problems requires an exponentially increasing amount of time to find a solution, the simplex method typically behaves as a polynomial-time algorithm for most problems encountered in practice. However average case complexity results are clearly weaker than worst case complexity bounds since they require us to specify a "prior probability measure" over the space of possible MDP problems and only guarantee that the mean square error of a candidate solution is less than $\epsilon$ (where the expectation is taken with respect to the prior probability measure). Specification of an appropriate prior can be a tricky business since its support equals the infinite-dimensional space of problem inputs $(\beta, u, p)$. Randomized complexity results are slightly stronger n the sense that we don't have to specify a prior distribution over potential problem instances and can still measure approximation error and computer run-time on a worst case basis. However we do have to be content with a somewhat weaker assurance about accuracy: instead of being guaranteed of generating an approximate solution that is within $\epsilon$ of the true solution with probability 1, a randomized algorithm produces an approximate solution that is within $\epsilon$ of the true solution with probability arbitrarily close to 1. Randomized algorithms are also subject to the criticism that

actual computers are only able to generate psuedo-random numbers using deterministic algorithms. However many computer scientists believe that "pseudo-random computation may be viewed as a close approximation of random computation, and that randomness is a very powerful tool for computation even if implemented on deterministic computers" (Traub, Wasilikowski and Woźniakowski, 1988, p. 414).[12]

In section 5 we present recent results from Rust (1994) that proves: 1) randomization does succeed in breaking the curse of dimensionality for the class of *discrete choice processes* (DDP's), i.e. MDP's with finite choice sets, although 2) randomization cannot break the curse of dimensionality for MDP's with continuous choice sets unless the problem has further structure, e.g. convexity. Rust's upper bound on the worst case randomized complexity of a finite horizon DDP problem with $D$ possible choices, $T$ time periods and a $k_s$-dimensional state variable $s_t$ is given by:

$$comp^{wor\text{-}ran}(\epsilon, k_s) = O\left(\frac{1}{\epsilon^4}\right). \tag{1.1}$$

The reason why randomization cannot break the curse of dimensionality for general continuous MDP's with continuous choice sets is quite simple: since the general nonlinear programming problem is a special case of the MDP problem when $\beta = 0$, the general MDP problem must be at least as hard as the general (static) nonlinear programming problem. However Nemirovsky and Yudin 1978 have proven that nonlinear programming problems are in the class *CD* regardless of whether deterministic or random algorithms are employed. In section 5.4 we conjecture that recent results of Woźniakowski (1991,1992) imply that DDP's are also tractable on average, i.e. their complexity is bounded by a polynomial function of the dimension $k_s$ where the prior probability measure on possible problem inputs is a Wiener sheet measure. The algorithm that we believe will succeed in breaking the curse of dimensionality uses the optimal integration algorithm of Woźniakowski 1991 (which evaluates integrands at the set of shifted Hammersley points) and Woźniakowski's multivariate function approximation algorithm (which is based in turn on Temlyakov's 1987 kernel estimator for smooth periodic functions).

While complexity theory provides a useful general guide to the analysis of numerical methods for solving general MDP's, most of the complexity bounds are best viewed as asymptotic results that show how solution time increases as $\epsilon \to 0$ and $k \to \infty$, but which rarely provide an explicit lower bound on solution time for fixed $(k, \epsilon)$. Numerical examples in section 5.4 show that an exponential time algorithm such the simplex method can be significantly faster than a polynomial time algorithm such as successive approximations for solving sufficiently small infinite horizon MDP problems when the discount factor is close to 1. A simple analogy to econometrics may prove helpful here. Although one can prove that the maximum likelihood estimator $\hat{\theta}_n$ based on $n$ observations is asymptotically efficient

---

[12] Recent theoretical analyses of random number generation show that there are pseudo-random generators whose results are indistinguishable from truly random numbers from the standpoint of "any observer who does not have enough computational power to 'understand' them. More specifically, pseudorandom generation is perhaps the most general and natural method to reduce or eliminate the randomness required by algorithms. Pseudorandom sequences may replace random sequences required by algorithms without changing the results in any way." (Nisan, p. 1).

as $n \rightarrow \infty$, any estimator of the form $\tilde{\theta} = \hat{\theta}_n + o_p(1/\sqrt{n})$ is also asymptotically efficient, where the second term denotes noise that goes to zero in probability faster than $1/\sqrt{n}$. The $o_p(1/\sqrt{n})$ term represents noise from use of other non-maximum likelihood procedures (e.g. Bayesian posterior mean) or from noise in the approximate calculation of the maximizing value of the likelihood. Just as we are concerned with evaluating the finite sample behavior of various asymptotically equivalent estimators, we are also concerned with the relative efficiency and accuracy of various numerical solution methods for fixed problems. Thus, section 5 presents the results of numerical tests comparing the speed and accuracy of various discrete and parametric approximation methods for the suite of continuous MDP test problems introduced in section 2.6. The test problems are some of the rare examples of continuous MDP's that have analytic solutions for $(V, \delta)$, making it quite convenient to directly compare the speed and accuracy of the various algorithms.[13]

Our review of MDP's and numerical methods is necessarily selective given the space constraints of this chapter. To provide some perspective of how our review fits into the larger literature on numerical dynamic programming and stochastic control, it is useful to briefly review the main variations of control problems encountered in the literature:

- deterministic vs. stochastic
- Markovian vs. non-Markovian
- discrete vs. continuous time
- finite vs. infinite horizon
- linear/quadratic vs. general nonlinear problems
- discrete vs. continuous state
- discrete vs. continuous control
- discounted vs. long-run average rewards
- perfect vs. imperfect state information

This chapter focuses on stochastic control since deterministic control problems are a special case and can generally be effectively solved using the same algorithms developed for stochastic problems. [14] We also focus on Markovian decision processes with additively separable utility functionals. Although the method of dynamic programming can be extended in a straightforward manner to solve finite-horizon problems with non-Markovian uncertainty and general utility functionals (see Hinderer 1970 or Gihman and Skorohod, 1979), these methods require keeping track of complete histories which make them computationally intractable for solving all but the smallest

[13] Although many of the solution methods we present have associated error bounds that allow us to bound the maximum distance between the true and approximate solutions, in many cases these bounds involve unknown constants which can be very difficult to evaluate.

[14] This is not completely true. Papadimitrious and Tsitsiklis 1987 have shown that certain deterministic MDP's are in the class *NC* whereas the general stochastic MDP problem is P-complete.

problems. We also focus on MDP's formulated in discrete rather than continuous time. While there is an elegant theory of continuous time dynamic programming in which the value function $V$ can be shown to be the solution to a partial differential equation known as the *Hamilton-Bellman-Jacobi* (HBJ) equation (see, e.g. Doshi, 1976, and Fleming and Soner, 1993), under very general conditions one can show that a continuous time MDP can be approximated arbitrary closely by a discrete time MDP when the time interval is sufficiently small (van Dijk, 1984). Indeed, the predominant solution method in the OR literature involves solving approximate discrete time formulations of the MDP problem rather than attempting to numerically solve the HBJ equation, which frequently reduces to a rather difficult parabolic partial differential equation (see Kushner, 1990 and Kushner and Dupuis, 1992).[15] For this reason, the complexity of solving continuous time MDP's is given by the same bounds as for discrete time MDP's, but with a discount factor that is close to 1. Similarly, although there is a special theory for the solution of MDP problems under the long-run average reward criterion, we focus on solving problems with discounted returns since a generalization of a classical theorem due to Abel (see Bhattacharya and Majumdar, 1989, or Dutta, 1989) shows that under weak regularity conditions $(1-\beta)V$ converges to the optimal long-run average reward as $\beta$ tends to 1. This implies that we can approximate the stationary optimal decision rule under long-run average rewards by solving a discounted MDP with $\beta$ close to 1. We also focus on problems of perfect state information, since one can always reduce a problem with imperfect state information to a problem with perfect state information by showing that the conditional distribution of the unobserved components of the state variable given the observed components is a sufficient statistic, and hence is the relevant state variable for a reformulated problem with perfect state information (see, e.g. Bertsekas, 1987). However while this Bayesian approach is conceptually simple, the reformulation falls prey to the curse of dimensionality since the new state variable in the reformulated problem is a conditional problability distribution, a multi-dimensional continuous state variable on the $k$-dimensional simplex. Papadimitriou and Tsitsiklis have shown that even discrete MDP's with partially observed states are significantly harder problems than discrete MDP's with fully observed states. Unlike the fully observed MDP which is in the class $P$, Papadimitriou and Tsitsiklis showed that the partially observed MDP problem is *NP-complete*.[16] Sections 4.4 and 5.6 of our review discuss non-Bayesian learning and control algorithms for solving LQ and general MDP problems with unknown parameters, respectively. It is an open question, however, whether these non-Bayesian methods avoid the curse of dimensionality inherent in the Bayesian approach. The remaining subject divisions: finite vs. infinite horizon, discrete vs. continuous states and controls, and methods for LQ vs. general nonlinear MDP's are all covered in separate sections of this chapter.

---

[15] Chapter 9 of Fleming and Soner reviews closely related numerical methods based on solving finite-difference approximations to the HJB equations, which Kushner and Dupuis (1992) show are closely related to methods based on solving approximate discrete time MDP's. For a review of the general literature on numerical methods for solving PDE's, see Ortega and Voigt, 1988.

[16] The class *NP* consists of problems that can be solved in polynomial time using a "non-deterministic" computer. Essentially, this means that one can verify the correctness of any conjectured solution in polynomial time. Many computer scientists believe that NP-problems have exponential worst case complexity, but the answer to this question depends on the resolution of the "$P = NP$ problem" which is a famous unsolved problem in computer science. If $P = NP$ then all problems that were previously thought to be exponential time problems (e.g. the "travelling salesman" problem) can actually be solved in polynomial time.

A final caveat is that the MDP framework involves the implicit assumption that decision-makers have time-separable preferences and are expected utility maximizers. Experimental tests of human decision-making show that neither of these assumptions may be valid (e.g. the famous "Allais paradox", see Machina, 1987). Recently, a new theory of sequential decision-making has emerged that allows for time and state non-separable preferences (see Epstein and Zin, 1989, and Ozaki and Streufert, 1992), and it's possible that this generalized dynamic choice theory could result in more realistic economic models. Apart from some special cases that have been solved numerically (e.g. Hansen and Sargent, 1993) there is virtually no theory or well developed numerical procedure for solving general versions of these problems. However these problems have a recursive structure, so it is quite likely that many of the methods outlined in this chapter can ultimately be adapted to solve these more general SDP's. We content ourselves with the observation, formalized in Rust (1994), that the class of MDP's is already sufficiently broad to enable one to generate virtually any type of decision rule (behavior) via an appropriate specification of preferences $u$ and law of motion $p$.

## 2. Solving MDP's via Dynamic Programming: A Brief Review

This section reviews the main theoretical results on dynamic programming in finite and infinite horizon problems. Readers who are already familiar with this theory may simply wish to skim over this section to determine the notation we are using, and skip directly to the presentation of numerical methods in sections 3 and 4.

**Definition 2.1:** *A (discrete-time)* ***Markovian Decision Process*** *consists of the following objects:*

- A *time index* $t \in \{0, 1, 2, \ldots, T\}$, $T \leq \infty$
- A *state space* $S$
- A *decision space* $D$
- A family of *constraint sets* $\{D_t(s_t) \subseteq D\}$
- A family of *transition probabilities* $\{p_{t+1}(\cdot|s_t, d_t) : \mathcal{B}(S) \Rightarrow [0, 1]\}$[17]
- A *discount factor* $\beta \in (0, 1)$ and family of *single period utility functions* $\{u_t(s_t, d_t)\}$ such that the utility functional $U_T$ has the additively separable decomposition: [18]

$$U_T(\mathbf{s}, \mathbf{d}) = \sum_{t=0}^{T} \beta^t u_t(s_t, d_t). \tag{2.1}$$

The agent's optimization problem is to choose an optimal decision rule $\delta^* = (\delta_0^T, \ldots, \delta_T^T)$ to solve the following problem:

$$\max_{\delta=(\delta_0,\ldots,\delta_T)} E_\delta\{U_T(\mathbf{s}, \mathbf{d})\} \equiv \int_{s_0} \cdots \int_{s_T} \left[\sum_{t=0}^{T} \beta^t u_t(s_t, \delta_t(s_t))\right] \prod_{t=1}^{T} p_t(ds_t|s_{t-1}, \delta_{t-1}(s_{t-1}) p_0(ds_0), \tag{2.2}$$

where $p_0$ is a probability distribution over the initial state $s_0$. Stated in this form, the optimization problem is extremely daunting. We must search over an infinite-dimensional space of sequences of potentially history-dependent functions $(\delta_0, \ldots, \delta_T)$, and each evaluation of the objective function in (2.2) requires $(T + 1)$-fold multivariate integration. We now show how the method of dynamic programming can be used to vastly simplify this potentially intractable optimization problem.

---

[17] $\mathcal{B}(S)$ is the *Borel $\sigma$-algebra* of measurable subsets of $S$. For simplicity, the rest of this chapter avoids measure-theoretic details since they are superfluous in the most commonly encountered case where both the state and control variables are discrete or under the topological and smoothness assumptions we will be imposing later in this chapter.

[18] The boldface notation denotes sequences: $\mathbf{s} = (s_0, \ldots, s_T)$. We will subsequently impose explicit toplogical structure on $S$ and $D$ and smoothness conditions on $\{u_t\}$ and $\{p_t\}$ later in this section and in sections 4 and 5.

## 2.1 Finite Horizon Dynamic Programming and the Optimality of Markovian Decision Rules

In finite-horizon problems ($T < \infty$), dynamic programming amounts to calculating the optimal decision rule $\delta^* = (\delta_0^T, \ldots, \delta_T^T)$ by backward induction starting at the terminal period, $T$. The backward recursion must be done for each time period $t = T, T-1, \ldots, 0$ and for each possible state $s_t$. In principle, the optimal decision at each time $t$ might depend not only on the current state $s_t$, but on the entire previous history of the process, i.e. $d_t = \delta_t^T(H_{t-1}, s_t)$ where $H_t = (s_0, \ldots, s_{t-1})$.[19] However since we are dealing with MDP's it is easy to see that the Markovian structure of $p$ and the additive separability of $U_T$ imply that it is unnecessary to keep track of the entire previous history: the optimal decision rule depends only on the current time $t$ and the current state $s_t$: $d_t = \delta_t^T(s_t)$. For example, starting in period $T$ we have:

$$\delta_T(H_{T-1}, s_T) = \underset{d_T \in D_T(s_T)}{argmax} \; U_T(H_{T-1}, s_T, d_0, \ldots, d_{T-1}, d_T), \tag{2.3}$$

where $U_T$ can be rewritten as:

$$\begin{aligned} U_T(H_{T-1}, s_T, d_0, \ldots, d_T) &= \sum_{t=0}^{T} \beta^t u_t(s_t, d_t) \\ &= \sum_{t=0}^{T-1} \beta^t u_t(s_t, d_t) + \beta^T u_T(s_T, d_T). \end{aligned} \tag{2.4}$$

It is clear that the optimal choice of $d_T$ is independent of previous history $H_{T-1}$ since $d_T$ appears only in the final term $u_T(s_T, d_T)$ on the right hand side of (2.4). Continuing the dynamic programming recursion, it is straightforward to verify that at each time $t$ the optimal decision rule $\delta_t^T$ depends only on $s_t$ and the current time $t$. A decision rule that depends on the past history of the process only via the current state $s_t$ and time $t$ is called *Markovian*. Notice also that the optimal decision rule can always be chose to be a deterministic function of $s_t$. Randomization can never help and typically reduces expected utility whenever the maximizing value of $d_T$ in (2.3) is unique.[20]

The *value function* is the expected discounted value of utility over the remaining horizon *assuming an optimal policy is followed in the future.* Economists recognize the value function as "shadow price" that facilitates intertemporal decentralization of the decision process. Others have provided more colorful characterizations of the value function, e.g. Howard (1960) described it as a "portable genius". Regardless of how it is described, the value function is a key to obtaining the optimal decision rule as well as being of fundamental interest in its own right. The method of dynamic

---

[19] If past decisions are known, measurable functions of past states, $d_j = \delta_j(H_j)$, $j = 0, \ldots, t-1$, there is no increase in information in including the realized decisions in $H_t$.

[20] Uniqueness of the optimal decision is a generic property in the sense that whenever multiple values of $d_T$ maximize (2.4), a slight perturbation of $u_T$ yields a similar problem where the maximizing value is unique.

programming calculates the value function and the optimal policy recursively as follows. In the terminal period $V_T^T$ and $\delta_T^T$ are defined by:

$$\delta_T^T(s_T) = \underset{d_T \in D_T(s_T)}{argmax} \left[u_T(s_T, d_T)\right], \tag{2.5}$$

$$V_T^T(s_T) = \underset{d_T \in D_T(s_T)}{max} \left[u_T(s_T, d_T)\right]. \tag{2.6}$$

In periods $t = 0, \ldots, T-1$, $V_t^T$ and $\delta_t^T$ are recursively defined by:

$$\delta_t^T(s_t) = \underset{d_t \in D_t(s_t)}{argmax} \left[u_t(s_t, d_t) + \beta \int V_{t+1}^T\left(s_{t+1}, \delta_{t+1}^T(s_{t+1})\right) p_{t+1}(ds_{t+1}|s_t, d_t)\right], \tag{2.7}$$

$$V_t^T(s_t) = \underset{d_t \in D_t(s_t)}{max} \left[u_t(s_t, d_t) + \beta \int V_{t+1}^T\left(s_{t+1}, \delta_{t+1}^T(s_{t+1})\right) p_{t+1}(ds_{t+1}|s_t, d_t)\right]. \tag{2.8}$$

It's straightforward to verfiy that at time $t = 0$ the value function $V_0^T(s_0)$ represents the maximized expected discounted value of utility in all future periods. Since dynamic programming has recursively generated the optimal decision rule $\delta^* = (\delta_0^T, \ldots, \delta_T^T)$, it follows that

$$V_0^T(s) = \underset{\delta}{max}\, E_\delta\{U_T(\tilde{\mathbf{s}}, \tilde{\mathbf{d}})|s_0 = s\}. \tag{2.9}$$

These basic results can be formalized as follows:

**Theorem 2.1:** *Given a finite horizon MDP ($T < \infty$) that satifisfies certain weak regularity conditions (see Gihman and Skorohod, 1979):*

1. *An optimal, non-randomized decision rule $\delta^*$ exists,*
2. *An optimal decision rule can be found within the class of non-randomized Markovian strategies,*
3. *The optimal decision rule $\delta^* = (\delta_0^T, \ldots, \delta_T^T)$ can be computed by backward induction according to the recursions (2.5), ..., (2.8).*
4. *The initial value function $V_0^T$ computed from the dynamic programming recursions satisfies the identity (2.9).*

## 2.2 Infinite Horizon Dynamic Programming and Bellman's Equation

In the infinite horizon case $T = \infty$ and there is no "last" period from which to start the backward induction to carry out the dynamic programming algorithm described in the previous section. However if the per period utility functions $u_t$ are uniformly bounded and the discount factor $\beta$ is in the $(0,1)$ interval, then we can approximate the infinite horizon utility functional $U_\infty(\mathbf{s},\mathbf{d})$ arbitrarily closely by a finite horizon utility functional $U_T(\mathbf{s},\mathbf{d})$ for $T$ sufficiently large. This is a basic idea underlying many numerical methods for solving infinite horizon MDP's such as successive approximations, all of which will be described in more detail in sections 3 and 4.

However almost all infinite-horizon MDP's formulated in economic applications have the further characteristic of *stationarity:* i.e. the transition probabilities and utility functions are the same for all $t$. In the finite horizon case the time homogeneity of $u$ and $p$ does not lead to any significant simplifications since there still is a fundamental nonstationarity induced by the fact that remaining utility $\sum_{j=t}^{T}\beta^j u(s_j,d_j)$ depends on $t$. However in the infinite-horizon case, the stationary Markovian structure of the problem implies that the future looks the same whether the agent is in state $s_t$ at time $t$ or in state $s_{t+k}$ at time $t+k$ provided that $s_t = s_{t+k}$. In other words, the only variable that affects the agent's view about the future is the value of his current state $s$. This suggests that the optimal decision rule and corresponding value function should be time invariant: i.e. for all $t \geq 0$ and all $s \in S$, $\delta_t^\infty(s) = \delta(s)$ and $V_t^\infty(s) = V(s)$. Removing time subscripts from equation (2.7), we obtain the following equation for the optimal stationary decision rule $\delta$:

$$\delta(s) = \underset{d\in D(s)}{argmax}\big[u(s,d) + \beta\int V(s')p(ds'|s,d)\big], \tag{2.10}$$

where $V$ is the solution to:

$$V(s) = \max_{d\in D(s)}\big[u(s,d) + \beta\int V(s')p(ds'|s,d)\big]. \tag{2.11}$$

Equation (2.11) is known as *Bellman's Equation*[21] and most of the efficient numerical methods described in section 5 focus on direct solution of this equation rather than attempting to solve an approximate finite-horizon version of the the problem by the backward induction. Note that Bellman's equation defines $V$ in terms of itself: i.e. it is a recursive equation. In mathematical terminology, Bellman's equation is a *functional equation* and $V$ is a *fixed point* to this functional equation. Bellman's equation has the further important property, apparently first noted by Denardo (1967), that $V$ is the fixed point to a *contraction mapping*.

---

[21] Bellman was not the first to discover this equation (for example versions of it appear in prior work by Arrow *et. al.* on optimal inventory policy), however the equation bears his name due to Bellman's systematic application of the approach to solving a wide variety of problems.

**2.3 Bellman's Equation, Contraction Mappings, and Blackwell's Theorem**

To establish the existence and uniqueness of a solution to Bellman's equation, assume for the moment the following regularity conditions hold: 1) $S$ and $D$ are complete metric spaces, 2) $u(s,d)$ is jointly continuous and bounded in $(s,d)$, 3) $s \rightarrow D(s)$ is a continuous correspondence. Let $B$ denote the vector space of all measurable, bounded functions $f: S \rightarrow R$ under the (essential) supremum norm, $|f| = sup_{s \in S} |f(s)|$. Then $B$ is a Banach Space, i.e. a complete normed linear space.[22] Define the *Bellman operator* $\Gamma: B \rightarrow B$ by:

$$\Gamma(W)(s) = \max_{d \in D(s)} \left[ u(s,d) + \beta \int W(s')p(ds'|s,d) \right]. \tag{2.12}$$

Bellman's equation can then be re-written in operator notation as:

$$V = \Gamma(V), \tag{2.13}$$

i.e. $V$ is a fixed point of the mapping $\Gamma$. It is not difficult to show that given any $V, W \in B$ we have:

$$|\Gamma(V) - \Gamma(W)| \leq \beta |V - W| \tag{2.14}$$

An operator that satisfies inequality (2.14) for some $\beta \in (0,1)$ is said to be a *contraction mapping*. The theory of contraction mappings allows us to establish the existence and uniqueness of the solution $V$ to Bellman's equation. In addition, since contraction mappings have extremely nice mathematical properties, the theory has spawned many of the solution methods and error bounds developed in section 5. Because of its simplicity and its role in the development of various solution methods, we repeat the proof often-stated contraction mapping theorem below.

**Banach Contraction Mapping Theorem:** *If $\Gamma$ is a contraction mapping on a Banach Space $B$, then $\Gamma$ has a unique fixed point $V \in B$.*

The uniqueness of the fixed point is a direct consequence of the contraction property (2.14): if $W$ and $V$ are fixed points to $\Gamma$ then (2.14) implies that

$$|V - W| = |\Gamma(V) - \Gamma(W)| \leq \beta |V - W|. \tag{2.15}$$

Since $\beta \in (0,1)$ then only possible solution to (2.15) is $|V - W| = 0$. The existence of a fixed point is a result of the completeness of the Banach space $B$. Starting from any initial element of $B$ (such as 0), the contraction property (2.14) implies that the following sequence of successive approximations forms a Cauchy sequence in $B$:

$$\{0, \Gamma(0), \Gamma^2(0), \Gamma^3(0), \ldots, \Gamma^n(0), \ldots\}. \tag{2.16}$$

---

[22] Recall that a metric space $S$ (which includes the normed linear space space $B$ as a special case) is said to be *complete* if every Cauchy sequence in $S$ converges to a point in $S$.

Since a the Banach space $B$ is complete, the Cauchy sequence converges to a point $V \in B$, so existence follows by showing that $V$ is a fixed point of $\Gamma$. To see this, note that a contraction $\Gamma$ is uniformly continuous, so:

$$V = \lim_{n\to\infty} \Gamma^n(0) = \lim_{n\to\infty} \Gamma(\Gamma^{n-1}(0)) = \Gamma(V), \tag{2.17}$$

i.e., V is indeed the required fixed point of $\Gamma$.

**Blackwell's Theorem:** *The stationary, Markovian, infinite-horizon policy given by $\delta^* = (\delta, \delta, \ldots)$ where $\delta$ is defined in (2.10) and (2.11) constitutes an optimal decision rule for the infinite-horizon MDP problem (2.2).*

This result follows by showing that the unique solution $V(s)$ to Bellman's equation coincides with the optimal value function $V_0^\infty$ defined in (2.9), which we restate below for convenience:

$$V_0^\infty(s) \equiv \max_\delta \; E_\delta\Big\{\sum_{t=0}^{\infty} \beta^t u(s_t, d_t)\Big| s_0 = s\Big\}. \tag{2.18}$$

Consider approximating the infinite-horizon problem by the solution to a finite-horizon problem with value function:

$$V_0^T(s) = \max_\delta \; E_\delta\Big\{\sum_{t=0}^{T} \beta^t u(s_t, d_t)\Big| s_0 = s\Big\}. \tag{2.19}$$

Since $u$ is bounded and continuous, $\sum_{t=0}^{T} \beta^t u(s_t, d_t)$ converges to $\sum_{t=0}^{\infty} \beta^t u(s_t, d_t)$ for any sequences $\mathbf{s} = (s_0, s_1, \ldots)$ and $\mathbf{d} = (d_0, d_1, \ldots)$. Using the Lebesgue dominated convergence theorem and properties of suprema, one can show that for each $s \in S$, $V_0^T(s)$ converges to the infinite-horizon value function $V_0^\infty(s)$:

$$\lim_{T\to\infty} V_0^T(s) = V_0^\infty(s) \qquad \forall s \in S. \tag{2.20}$$

But the contraction mapping theorem also implies that this same sequence converges uniformly to $V$ (since $V_0^T = \Gamma^T(0)$), so it follows that $V = V_0^\infty$. Since $V$ is the expected present discounted value of utility under the policy $\delta^*$ (a result we demonstrate in section 2.4), the fact that $V = V_0^\infty$ immediately implies the optimality of $\delta^*$.

The optimality of the infinite horizon policy $\delta^*$ can be proved under weaker conditions that allow $u(s, d)$ to be an unbounded, upper semicontinuous function of $s$ and $d$, see Bhattarcharya and Majumdar, 1989. Although problems with unbounded state spaces $S$, decision spaces $D$, and utility functions $u$ arise frequently in economic applications, in practice most of the computational methods presented in section 5 require bounded domains and utility functions. We can approximate the solution to a problem with unbounded state space, decision space, and utility function via a sequence of bounded problems where the bounds tend to infinity. Furthermore, the solution to problems with upper-semicontinuous utility functions can be approximated as the limit of a sequence of solutions to problems with continuous utility functions, see Gihman and Skorohod 1979, Lemma 1.8.

It turns out that many of the solution methods for solving stationary, infinite horizon MDP's described in section 5 can be interpreted more broadly as numerical methods for computing fixed points to general contraction mappings. Indeed, several of the methods involve computation of fixed points to "nearby" contraction mappings $\Gamma_\sigma$ that can be viewed as perturbed versions of the Bellman operator $\Gamma$ in the sense that for each $W \in B$:

$$\lim_{\sigma \to 0} \Gamma_\sigma(W) = \Gamma(W). \tag{2.21}$$

The parameter $\sigma$ can be thought of as indexing a level of "noise" intended to "smooth out" potential non-differentiabilities caused by the $max$ operation in the Bellman operator. For example consider the operator $\Gamma_\sigma : B \to B$ defined by:

$$\Gamma_\sigma(V)(s) = \int \max_{d \in D(s)} \left[ u(s,d) + \sigma\eta(d) + \beta \int V(s')p(ds'|s,d) \right] q(d\eta|s), \tag{2.22}$$

where $q(d\eta|s)$ is a continuous distribution function on $R^{k_s}$ where $k_s$ is the number of elements in $D(s)$. This operator can be viewed as providing the expected value of the value function when choices are perturbed by small stochastic shocks $\eta$. In the special case where $\eta$ has a (shifted) multivariate extreme value distribution, the operator $\Gamma_\sigma$ is given by:

$$\Gamma_\sigma(V)(s) = \sigma \, log \left[ \sum_{d \in D(s)} exp \left\{ \frac{u(s,d) + \beta \int V(s')p(ds'|s,d)}{\sigma} \right\} \right]. \tag{2.23}$$

It is straightforward to verify that for any $\sigma > 0$ the $\Gamma_\sigma(V)$ is a contraction mapping and a Fréchet differentiable function of $V$ that converges to $\Gamma(V)$ as $\sigma$ tends to zero. The following lemma shows that the fixed point $V_\sigma$ to the contraction mapping $\Gamma_\sigma$ converges to the fixed point $V$ of the contraction mapping $\Gamma$ as $\sigma$ tends to zero.

**Lemma 2.1:** *Suppose $\{\Gamma_\sigma\}$ are a family of contraction mappings on a Banach space $B$ indexed by a single real parameter $\sigma$ that satisfies condition (2.21). Then the fixed point $V_\sigma = \Gamma_\sigma(V_\sigma)$ tends to the fixed point $V = \Gamma(V)$ as $\sigma \to 0$ and satisfies the error bound:*

$$|V_\sigma - V| \leq \frac{|\Gamma_\sigma(V) - \Gamma(V)|}{(1-\beta)}. \tag{2.24}$$

The proof of this lemma is a simple application of the triangle inequality:

$$\begin{aligned} |V_\sigma - V| &= |\Gamma_\sigma(V_\sigma) - \Gamma(V)| \\ &\leq |\Gamma_\sigma(V_\sigma) - \Gamma_\sigma(V)| + |\Gamma_\sigma(V) - \Gamma(V)| \\ &\leq \beta |V_\sigma - V| + |\Gamma_\sigma(V) - \Gamma(V)|. \end{aligned} \tag{2.25}$$

A closely related example of a perturbed problem arises from homotopy or "path following" algorithms that compute value functions $V_\beta$ to MDP's for different values of $\beta$. When $\beta = 0$ we have a static optimization that is generally relatively easy to solve (such as is the case whenever $D(s)$ is a finite set for each $s \in S$), and as $\beta$ increases monotonically towards 1 we get a sequence of increasingly difficult dynamic problems. The idea behind the path

following algorithm is to solve the value function $V_{\beta_n}$ for a sequence $\beta_n \uparrow \beta$ using the solution $V_{\beta_{n-1}}$ as the starting value in some algorithm for finding the solution $V_{\beta_n}$ at $\beta_n$. We conclude this section by presenting several fundamental error bounds for contraction mappings that will be key to establishing error bounds and convergence of several of the numerical methods presented in section 5.

**Lemma 2.2:** *Let $\Gamma$ be a contraction mapping on a Banach space $B$ with fixed point $V = \Gamma(V)$. If $W$ is an arbitrary element of $B$ the following inequalities hold:*

$$|W - \Gamma(W)| \leq (1+\beta)|V - W| \tag{2.26}$$

$$|\Gamma^n(V) - V| \leq \beta^n |\Gamma(W) - W|/(1-\beta) \tag{2.27}$$

$$|\Gamma^{n+1}(V) - V| \leq \beta^n |\Gamma^{n+1}(V) - \Gamma^n(V)|/(1-\beta) \tag{2.28}$$

$$|\Gamma_{n+1}(V) - V| \leq \beta |\Gamma^n(V) - V| \tag{2.29}$$

Error bound (2.26) shows that if $W$ is close to the fixed point $V$ then $W$ must also be close to $\Gamma(W)$. Error bound (2.26), known as an *a priori* error bound, shows the converse result: the maximum error in a sequence of successive approximations $\{V_n\}$ starting from $W$ is a geometrically declining function of the initial error $|W - \Gamma(W)|$. These two inequalities will be the keys to establishing the convergence of the various parametric approximation methods presented in section 5. Inequality (2.28) is referred to as an *a posteriori* error bound: it enables us to bound the maximum distance of $V_{n+1}$ to the fixed point $V$ in terms of the (observable) change in the value function from iteration $n$ to $n+1$. We state this bound for completeness, since in section 5 we will show that in the special case where $\Gamma$ is the Bellman operator one can derive a much sharper *a posteriori* error bound known as the *McQueen-Porteus error bound*. The final bound, (2.29), provides an upper bound on the rate of convergence of the method of successive approximations. Generally, the following limiting relationship obtains:

$$\lim_{n\to\infty} \frac{|V_{n+1} - V|}{|V_n - V|} = \beta. \tag{2.30}$$

and we say that the method of successive approximations *converges linearly to $V$ with convergence ratio $\beta$*. The final lemma is a useful result that implies that the fixed point to Bellman's equation always lies within a maximum distance $K$ of the origin.

**Lemma 2.3:** *Let $\Gamma$ be a Bellman operator. Then we have:*

$$\Gamma : B(0,K) \to B(0,K), \tag{2.31}$$

*where $B(0,K) \subset B$ is the ball centered at 0 with radius $K$ given by:*

$$K \equiv \max_{t\in\{1,\ldots,T\}} \sup_{s\in S} \sup_{d\in D_t(s)} \frac{|u_t(s,d)|}{(1-\beta)}. \tag{2.32}$$

## 2.4 A Geometric Series Representation for MDP's

Stationary, discounted infinite-horizon MDP's have a beautiful algebraic structure that directly generalizes simple geometric series, prompting Lucas's (1978) observation that "a little knowledge of geometric series goes a long way." (p. 1443). Since many of the numerical methods presented in section 5 directly exploit this algebraic structure, we provide a brief review here. To avoid technicalities, we first present the basic results for the case of finite MDP's where, without loss of generality, both the state space $S$ and the decision space $D$ can be identified as finite sets of integers, $S = \{1, \ldots, N\}$ and $D(s) = \{1, \ldots, \#D(s)\}$. In this framework a feasible stationary decision rule $\delta$ is an $N$-dimensional vector satisfying $\delta(s) \in \{1, \ldots, D(s)\}, s = 1, \ldots, N$, and the value function $V$ is an $N$-dimensional vector in the Euclidean space $R^N$. Given $\delta$ we can define a vector $u_\delta \in R^N$ whose $i^{th}$ component is $u(i, \delta(i))$, and an $N \times N$ *transition probability matrix* $E_\delta$ whose $(i, j)$ element is $p(j|i, \delta(i)) = Pr\{s_{t+1} = j | s_t = i, d_t = \delta(i)\}$. The Bellman equation $V = \Gamma(V)$ simplifies to:

$$V(s) = \Gamma(V)(s) \equiv \max_{1 \le d \le D(s)} \Big[u(s, d) + \beta \sum_{s'=1}^{N} V(s') p(s'|s, d)\Big]. \tag{2.33}$$

Given a stationary, Markovian decision rule $\delta^* = (\delta, \delta, \ldots)$, define the associated value function $V_\delta \in R^N$ as the vector of expected discounted utilities under policy $\delta$. It is straightforward to show that $V_\delta$ is the solution to a system of $N$ linear equations in $N$ unknowns, or in matrix form:

$$V_\delta = G_\delta(V_\delta) \equiv u_\delta + \beta E_\delta V_\delta. \tag{2.34}$$

It turns out that this equation can always be solved by simple matrix inversion:

$$\begin{aligned} V_\delta &= [I - \beta E_\delta]^{-1} u_\delta \\ &= u_\delta + \beta E_\delta u_\delta + \beta^2 E_\delta^2 u_\delta + \beta^3 E_\delta^3 u_\delta + \cdots \end{aligned} \tag{2.35}$$

The last equation in (2.35) is simply a geometric series expansion for $V_\delta$ in powers of $\beta$ and $E_\delta$. As is well known $E_\delta^k = (E_\delta)^k$ is simply the *k-stage transition probability matrix*, whose $(i, j)$ element equals $Pr\{s_{t+k} = j | s_t = i, \delta\}$, where the presence of $\delta$ as a conditioning argument denotes the fact that all intervening decisions satisfy $d_{t+j} = \delta(s_{t+j}), j = 0, \ldots, k$. Since $\beta^k E_\delta^k u_\delta$ is the expected discounted utility received in period $k$ under policy $\delta$, formula (2.35) can be thought of as a vector generalization of a geometric series, showing explicitly how $V_\delta$ equals the sum of expected discounted utilities under $\delta$ in all future periods. Note that since $E_\delta^k$ is a transition probability matrix (i.e. all of its elements are between 0 and 1, and its rows sum to unity), it follows that $lim_{k \to \infty} \beta^k E_\delta^k = 0$, which guarantees that the geometric series representation in (2.35) is convergent and therefore that tha $S \times S$ matrix $[I - \beta E_\delta]$ is invertible for any Markovian decision rule $\delta$ and all $\beta \in [0, 1)$. Although this is indeed a simple result in geometric

series, economists will recognize that this same algebraic structure appears in input-output theory (Leontief, 1966) and mathematicians will recognize this as a special case of the more general theory of *M-matrices.*[23]

In fact, the basic geometric series representation for $V_\delta$ in (2.35) also holds in problems where the state and decision spaces are infinite sets, although mathematicians refer to the series expansion (2.35) as a *Neumann series* instead of a geometric series in this case. When the state space $S$ has infinitely many elements $E_\delta$ is no longer represented by a transition probability matrix, but by a special kind of linear operator known as a *Markov operator* given by:[24]

$$E_\delta(V)(s) = \int V(s')p(ds'|s, \delta(s)) \tag{2.36}$$

Using the standard definition of a norm $|L|$ of a linear operator $L : B \to B$,

$$|L| = \sup_{V \neq 0} \frac{|L(V)|}{|V|}, \tag{2.37}$$

it is straightforward to verify that the linear operator $E_\delta$ has unit norm, $|E_\delta| = 1$. It is also easy to verify that the operator $E_\delta^2 = E_\delta(E_\delta)$ is also a Markov operator. By induction, it follows that $E_\delta^k = E_\delta(E_\delta^{k-1})$ is also Markov operator. Furthermore using the definition of the operator norm it is easy to see that

$$|E_\delta^k| \leq |E_\delta|^k, \tag{2.38}$$

which implies that the Neumann series expansion in equation (2.35) is convergent. Banach's Theorem (Theorem 4.4.3 of Kantorovich and Aikilov, 1982) then implies that the inverse operator $[I - \beta E_\delta]^{-1}$ exists and is a bounded linear operator on $B$. Thus, the value function $V_\delta$ is well-defined even when $S$ is infinite, a result that also follows directly from the fact that the operator $G_\delta$ defined in (2.34) is a contraction mapping.

In section 5 we will establish error bounds and prove the convergence of discretization methods as the number of states $S$ used in the finite approximations tend to infinity. The basic idea is to approximate the Markov operator $E_\delta$ on an infinite-dimensional function space $B$ by an $N \times N$ transition probability matrix on the ordinary Euclidean space $R^N$, and show that this implies that the solution $V_\delta \in B$ from equation (2.35) can be approximated by appropriately interpolating the vector solution for $V_\delta$ on $R^N$.

---

[23] See Berman and Plemmons, 1993 for a general characterization of necessary and sufficient conditions for matrices of the form $I - A$ to be invertible. These theorems subsume the Hawkins-Simon 1949 condition that guarantees that $lim_{n\to\infty} A^n = 0$ provided that certain conditions on the principal minors of $A$ are satisfied.

[24] If the state space consists of a countably infinite number of points, then $E_\delta$ can be represented as an infinite matrix.

## 2.5 Examples of Bellman's Equation for Specific MDP's

We now provide several concrete illustrations of MDP's that arise in typical economic problems, showing how the theory in the previous sections can be used to solve them. Although we stressed in the introduction that analytical solutions to DP problems are rare and non-robust (in the sense that small perturbations of the problem formulation leads to a problem with no analytic solution), we present analytical solutions in order to provide a "test bed" of problems to compare the accuracy and speed of various numerical solution methods presented in section 5. Examples 2 and 3 reflect two polar extreme types of infinite-horizon problems that arise in economic applications: *continuous choice* versus *discrete choice* problems. Since continuous choice problems require optimization over a continuum of possible decisions $d \in D(s)$, they are typically much more difficult to solve than discrete choice problems. In order to differentiate these two cases we will use the notation $d \in D(s)$ to denote the case of discrete choice (where $D(s)$ contains a finite or countably infinite number of possible values), and $c \in D(s)$ to denote the case of continuous choice (where $D(s)$ contains a continuum of possible values, such as a convex subset of Euclidean space).

**Example 1: A trivial problem.** Consider a problem where $u(s,d) = 1$ for all $d \in D(s)$ and all $s \in S$. Given that the utility function is a constant, it is reasonable to conjecture that $V^*$ is a constant also. Substituting this conjecture into Bellman's equation we obtain:

$$V^* = \max_{d \in D(s)} \left[1 + \beta \int V^* p(ds' | s, d)\right], \tag{2.39}$$

the unique solution to which is easily seen to be $V^* = 1/(1-\beta)$. This is the well known formula for a geometric series, $V^* = [1 + \beta + \beta^2 + \cdots]$ which is clearly equal to expected utility in this case since $u$ is identically equal to 1. This provides a simple and basic test of any solution method for infinite horizon MDP's: the method should return a value function identically equal to $1/(1-\beta)$ whenever we input a utility function that is identically 1.

**Example 2: A problem with continuous state and control variables.** Consider the problem of optimal consumption and savings analyzed by Phelps (1962). In this case the state variable $s$ denotes a consumer's current wealth, and the decision $d$ is how much to consume in the current period. Since consumption is a continuous decision, we will use $c_t$ rather than $d_t$ to denote the values of the control variable, and let $w_t$ to denote the state variable wealth. The consumer is allowed to save, but is not allowed to borrow against future income. Thus, the constraint set is $D(w) = \{c | 0 \leq c \leq w\}$. The consumer can invest his savings in a single risky asset with random rate of return $\{\tilde{R}_t\}$ is an *IID* process (i.e. idependently and identically distributed over time) withi marginal distribution $F$. Thus, $p(dw_{t+1} | w_t, c_t) = F\big((dw_{t+1}/(w_t - c_t)\big)$. Let the consumer's utility function be given by $u(w,c) = log(c)$. Then Bellman's equation for this problem is given by:

$$V^*(w) = \max_{0 \leq c \leq w} \left[log(c) + \beta \int_0^\infty V^*\big(R(w-c)\big) F(dR)\right] \tag{2.40}$$

Working backward from an initial conjecture $V = 0$ we see that at each stage $V_t$ has the form, $V_t(w) = A_t\, log(w) + B_t$ for constants $A_t$ and $B_t$. Thus, it is reasonable to conjecture that this form holds in the limit as well. Inserting the conjectured functional form $V^*(w) = A_\infty\, log(w) + B_\infty$ into (2.40) and solving for the unknown coefficients $A_\infty$ and $B_\infty$ we find:

$$\begin{aligned} A_\infty &= 1/(1-\beta) \\ B_\infty &= log(1-\beta)/(1-\beta) + \beta\, log(\beta)/(1-\beta)^2 + \beta E\{log(\tilde{R})\}/(1-\beta)^2, \end{aligned} \tag{2.41}$$

and the optimal decision rule or *consumption function* is given by:

$$\delta^*(w) = (1-\beta)w. \tag{2.42}$$

Thus, the logarithmic specification implies that a strong form of the *permanent income hypothesis* holds in which optimal consumption is independent of the distribution $F$ of investment returns.

**Example 3: A problem with discrete control and continuous state variable.** Consider the problem of optimal replacement of durable assets analyzed in Rust (1985,1986). In this case the state space $S = R_+$, where $s_t$ is interpreted as a measure of the accumulated utilization of the durable (such as the odometer reading on a car). Thus $s_t = 0$ denotes a brand new durable good. At each time $t$ there are two possible decisions {keep,replace} corresponding to the binary constraint set $D(s) = \{0, 1\}$ where $d_t = 1$ corresponds to selling the existing durable for scrap price $\underline{P}$ and replacing it with a new durable at cost $\overline{P}$. Suppose the level of utilitization of the asset each period has an exogenous exponential distribution. This corresponds to a transition probability $p$ is given by:

$$p(ds_{t+1}|s_t, d_t) = \begin{cases} 1 - exp\{-\lambda(ds_{t+1} - s_t)\} & \text{if } d_t = 0 \text{ and } s_{t+1} \geq s_t \\ 1 - exp\{-\lambda(ds_{t+1} - 0)\} & \text{if } d_t = 1 \text{ and } s_{t+1} \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2.43}$$

Assume the per-period cost of operating the asset in state $s$ is given by a function $c(s)$ and that the objective is to find an optimal replacement policy to minimize the expected discounted costs of owning the durable over an infinite horizon. Since minimizing a function is equivalent to maximizing its negative, we can define the utility function by:

$$u(s_t, d_t) = \begin{cases} -c(s_t) & \text{if } d_t = 0 \\ -[\overline{P} - \underline{P}] - c(0) & \text{if } d_t = 1. \end{cases} \tag{2.44}$$

Bellman's equation takes the form:

$$\begin{aligned} V^*(s) = max\Big[ &-c(s) + \beta \int_s^\infty V^*(s')\lambda\, exp\{-\lambda(s'-s)\}ds', \\ &-[\overline{P} - \underline{P}] - c(0) + \beta \int_0^\infty V^*(s')\lambda\, exp\{-\lambda(s')\}ds'\Big]. \end{aligned} \tag{2.45}$$

Observe that $V^*$ is a non-increasing, continuous function of $s$ and that the second term on the right hand side of (2.45), the value of replacing the durable, is a constant independent of $s$. Note also that $\overline{P} > \underline{P}$ implies that it is never optimal to replace a brand-new durable $s = 0$. Let $\gamma$ be the smallest value of $s$ such that the agent is indifferent between keeping and replacing. Differentiating Bellman's equation (2.45), it follows that on the *continuation region*, $[0, \gamma)$, $V^*$ satisfies the differential equation:

$$V^{*'}(s) = -c'(s) + \lambda c(s) + \lambda(1-\beta)V^*(s). \tag{2.46}$$

This is known as a *free boundary value problem* since the boundary condition:

$$V^*(\gamma) = [\overline{P} - \underline{P}] + V^*(0) = -c(\gamma) + \beta V^*(\gamma) = \frac{-c(\gamma)}{1-\beta}, \tag{2.47}$$

is determined endogenously. Equation (2.46) is a linear first order differential equation that can be integrated to yield the following closed-form solution for $V^*$:

$$V^*(s) = max\Big[\frac{-c(\gamma)}{1-\beta}, \frac{-c(\gamma)}{1-\beta} + \int_s^\gamma \frac{c'(y)}{1-\beta}[1 - \beta e^{-\lambda(1-\beta)(y-s)}]dy\Big], \tag{2.48}$$

where $\gamma$ is the unique solution to:

$$[\overline{P} - \underline{P}] = \int_0^\gamma \frac{c'(y)}{1-\beta}[1 - \beta e^{-\lambda(1-\beta)y}]dy. \tag{2.49}$$

It follows that the optimal decision rule is given by:

$$\delta^*(s) = \begin{cases} 0 & \text{if } s \in [0, \gamma] \\ 1 & \text{if } s > \gamma. \end{cases} \tag{2.50}$$

### 2.6 Discrete and Continuous Decision Processess

In this chapter the terms "discrete" and "continuous" refer to both the state and control variable. However for analytical purposes, there is a major difference in the nature of solution strategy for MDP's with discrete control variables versus MDP's with continuous control variables. We now define two subclasses of MDP's: *discrete decision processes* (DDP's) and *continuous decision processes* (CDP's) and derive a *stochastic Euler equation* characterizing the optimal decision rule in the latter class of problems.

**Definition 2.2:** *A **Discrete Decision Process** is an MDP satisfying:*

- The decision space $D$ is a finite set, and for each $s \in S$ we have $D(s) \subset D$.

**Definition 2.3:** *A **Continuous Decision Process** (CDP) is an MDP satisfying:*[25]

[25] This is a slight generalization of a class of problems analyzed in chapters 9 and 10 of Stokey and Lucas, 1989.

- The decision space $D$ is a subset of $R^M$.

- The state space $S$ is the product space $S = Y \times Z$, where $Y$ is a closed subset of $R^J$ and $Z$ is a closed subset of $R^K$.

- $D(s) = D(y,z)$ is an upper hemicontinuous correspondence which is increasing in its first argument: $y \leq y' \Rightarrow D(y,z) \subset D(y',z)$.

- $D(\cdot,\cdot)$ is convex in its first argument: i.e. for all $z \in Z$ and all $y, y' \in D$ if $c \in D(y,z)$ and $c' \in D(y',z)$ then $\theta c + (1-\theta)c' \in D(\theta y + (1-\theta)y', z)$ for all $\theta \in [0,1]$.

- The transition probability $p(ds_{t+1}|s_t, c_t)$ factors as:

$$p(ds_{t+1}|s_t, c_t) = p(dy_{t+1}, dz_{t+1}|y_t, z_t, c_t) = I\{dy_{t+1} = r(y_t, c_t, z_{t+1})\}q(dz_{t+1}|z_t). \tag{2.51}$$

  where $q(\cdot|z)$ is a weakly continuous function of $z$, and $r$ is a continuously differentiable function of $(y,c)$.

- For each (y,z,c) there exists an $M \times J$ matrix $h(y,c)$ satisfying:

$$\frac{\partial r(y,c,z)}{\partial y} = \frac{\partial r(y,c,z)}{\partial c} h(y,c). \tag{2.52}$$

- The utility function $u(s,c) = u(y,z,c)$ is a strictly concave function of its first and third arguments for each $z \in Z$ and strictly increasing in its first argument for each $(z,c) \in Z \times D$.

The interpretation of this class of problems is that the the state variable $s_t = (y_t, z_t)$ consists of two components: an "endogenous" state variable $y_t$ and an "exogenous" state variable $z_t$. The exogenous state variable is so named because its law of motion $q(dz_{t+1}|z_t)$ is unaffected by the agent's decisions $c_t$. The exogenous state variable $y_t$ is affected by the agent's decisions, but in a particular way: with probability 1 the value of $y_{t+1}$ is given by $y_{t+1} = r(y_t, c_t, z_{t+1})$.[26] Bellman's equation for this class of problems is given by:

$$V^*(s) = V^*(y,z) = \max_{c \in D(y,z)} \left[u(y,z,c) + \beta \int V^*\big(r(y,c,z'), z'\big) q(dz'|z)\right]. \tag{2.53}$$

**Lemma 2.4:** *Let $\delta^*$ be the optimal decision rule and $V^*$ be the value function for the continuous decision process given in (2.53). If $y \in int\, Y$ and $\delta(y,z) \in int\, D(y,z)$, then $V(y,z)$ is a continuously differentiable in $y$ at the point $(y,z)$ with derivative given by:*

$$\frac{\partial V(y,z)}{\partial y} = \frac{\partial u\big(y,z,\delta^*(y,z)\big)}{\partial y} - \frac{\partial u\big(y,z,\delta^*(y,z)\big)}{\partial c} h\big(y,\delta^*(y,z)\big). \tag{2.54}$$

---

[26] The Stokey-Lucas (1989) model is a special case when $y_{t+1} = c_t$ with probability 1.

The proof of the Lemma 2.4 is a straightforward extension of the proof of Theorem 9.10 of Lucas and Stokey (1989). The intuition behind (2.54) is quite simple: it is just a special case of the "envelope theorem". To see this, note that the first order conditions for optimality from (2.53) are:

$$\frac{\partial u(y,z,c)}{\partial c} + \beta \int \frac{\partial V(r(y,c,z'),z')}{\partial y} \frac{\partial r(y,c,z')}{\partial c} q(dz'|z) = 0, \tag{2.55}$$

where $c = \delta^*(y,z)$ is the maximizing value of $c$ in (2.53). Differentiating the value function in (2.53) with respect to $y$ and substituting in the optimal decision rule $\delta^*(y,z)$ on the right hand side we obtain:

$$\begin{aligned}\frac{\partial V(y,z)}{\partial y} = &\left(\frac{\partial u(y,z,c)}{\partial c} + \beta \int \frac{\partial V(r(y,c,z'),z')}{\partial y} \frac{\partial r(y,c,z')}{\partial c} q(dz'|z)\right) \frac{\partial \delta^*(y,z)}{\partial y} \\ &+ \frac{\partial u(y,z,c)}{\partial y} + \beta \int \frac{\partial V(r(y,c,z'),z')}{\partial y} \frac{\partial r(y,c,z')}{\partial y} q(dz'|z).\end{aligned} \tag{2.56}$$

The first order condition (2.55) implies that the first term of the second equation in (2.56) is 0 when evaluated at $c = \delta^*(y,z)$. Using assumption (2.52) and the first order condition (2.55), we can substitute for the last term in (2.56) to yield the result (2.54).

If equation (2.54) holds for *all* points $(y,z) \in Y \times Z$, we can substitute the right hand side of formula (2.54) for the term $\partial V(r(y,c,z'),z')/\partial y$ appearing inside the integral in (2.55) to obtain a general form of the *stochastic Euler equation:*

$$0 = \frac{\partial u(y,z,c)}{\partial c} + \beta \int \left[\frac{\partial u(y',z',c')}{\partial y} - \frac{\partial u(y',z',c')}{\partial c} h(y',c')\right] \frac{\partial r(y',c,z')}{\partial c} q(dz'|z), \tag{2.57}$$

where $c = \delta^*(y,z)$, $y' = r(y,\delta^*(y,z),z')$, and $c' = \delta^*(y',z')$. The Euler equation is simply a first order necessary condition characterizing the optimal decision rule $\delta^*(y,z)$. It says that the change in expected discounted expected utility from a small change in $c$ will be zero when $c = \delta^*(y,z)$. The remarkable feature of the first order condition for this class of problems is that the impact on expected discounted utility can be evaluated only in terms of the value of marginal utility in period $t$ and the expected discounted value of marginal utility in period $t+1$: it is not necessary to consider the impact on marginal utility on all future periods $t+2, t+3, \ldots$ Notice that (2.57) is a recursive formula characterizing the optimal decision rule $\delta^*$, similar to the recursive characterization of $V^*$ in Bellman's equation (2.53). It is not immediately apparent, however, that this particular functional equation is any simpler to solve than the functional equation (2.53) characterizing $V^*$. Indeed, the solution to (2.57) generally can't be formulated as the fixed point to a contraction mapping, and therefore may not have the nice mathematical solution properties that contraction mappings enjoy. Nevertheless we will show in section 5 that the stochastic Euler equation can serve as a basis for effective computational methods. The key restriction that limits the applicability of the method is the assumption that $\delta^*(c,z)$ is always an interior point of the constraint set. In cases when the maximizing value of $c$ in (2.53) occurs on

the boundary of the constraint set, the first order conditions to (2.53) take the form of *inequalities* rather than equalities, so that the standard form of the Euler equation (2.57) is no longer valid.

**Example 1** Consider Phelp's model of optimal consumption and saving in example 2 of the previous section. This is an example of a continuous decision process with exogenous shocks since the *IID* investment return process $\{R_t\}$ is a special case of the Markovian exogenous state variable $\{z_t\}$ given in (3.38), where $q(dz'|z) = F(dz')$. The endogenous state variable is current wealth $w$. Thus the state variable in this problem is $s = (w, z)$, and the control variable $c$ is consumption, and the law of motion for the endogenous state variable is $w_{t+1} = r(w_t, c_t, z_{t+1}) = z_{t+1}(w_t - c_t)$. The $h$ function in this case is given by $h(w, c) = -1$, and the utility function is given by $u(c)$, independent of the state $s$. One can also verify that Phelp's problem also satisfies all of the other regularity conditions listed in our definition of a CDP. The stochastic Euler equation (2.57) for this problem takes the form:

$$u'(\delta^*(w)) = \beta \int z' u'\big(\delta^*(z'(w - \delta^*(w)))\big) F(dz'). \tag{2.58}$$

Note that in this case, the assumption that the exogenous state variable is *IID* implies that the optimal decision rule $\delta^*$ is a function only of $w_t$. It is straightforward to verify that the stochastic Euler equation does indeed hold in the special case $u(c) = log(c)$. Substituting $log(c)$ and the optimal decision rule $\delta^*(w_t) = (1 - \beta)w_t$ into (2.58) we get:

$$\begin{aligned} 0 &= \frac{1}{(1-\beta)w} - \beta \int \frac{z'}{(1-\beta)w'} F(dz') \\ &= \frac{1}{(1-\beta)w} - \beta \int \frac{z'}{(1-\beta)\big(z'(w-(1-\beta)w)\big)} F(dz') \\ &= \frac{1}{(1-\beta)w} - \frac{1}{(1-\beta)w}. \end{aligned} \tag{2.59}$$

**Example 2** Consider the Brock-Mirman (1972) stochastic growth model. This problem has been a sort of canonical test problem for the effectiveness of various numerical methods in a problem with multidimensional continuous state variables. The state variable for this problem is $s_t = (k_t, z_t)$ where the endogenous state variable $k_t$ is the current capital stock and the exogenous state variable $z_t$ is interpreted as a "technology shock" with Markov transition probability $q(dz_{t+1}|z_t)$. Output is produced by a production function $f(k, z)$ which is increasing and strictly concave in $k$ and increasing in $z$. Capital and output are assumed to be the same commodity, so the existing capital stock $k$ and current period output $f(k, z)$ can be either reinvested or used for current consumption $c$. Thus the constraint set is given by $D(k, z) = \{c | 0 \leq c \leq f(k, z) + k\}$, and the law of motion $r$ for the next period's capital stock is $k_{t+1} = r(k_t, c_t, z_{t+1}) = f(k_t, z_{t+1}) + k_t - c_t$. The value function $V^*$ for this problem can be written as:

$$V^*(s) = V^*(k, z) = \max_{0 \leq c \leq f(k,z)+k} \big[u(c) + \beta \int V^*(f(k, z') + k - c, z') q(dz'|z)\big]. \tag{2.60}$$

Since this specification of the stochastic growth problem meets all the regularity conditions of the previous Lemma, the value function is differentiable and the stochastic Euler equation (2.57) for this problem takes the form:

$$u'\big(\delta^*(k,z)\big) = \beta \int u'\big(\delta^*(f(k,z)+k-\delta^*(k,z),z')\big)h\big(f(k,z)+k-\delta^*(k,z),z'\big)\,q(dz'|z), \tag{2.61}$$

where in this case the function $h$ defined in (2.52) is given by $h(k,z) = -[1+\partial f(k,z)/\partial k]$. In certain cases the Euler equation can be used to derive closed-form solutions for $\delta^*$. For example suppose that $u(c) = c$, $f(k,z) = zk^\alpha$, $\alpha \in (0,1)$, and specifying the technology shock $\{z_t\}$ as a log-normal process $log(z_{t+1}) = \rho\, log(z_t) + \epsilon_t$, where $\epsilon_t$ is distributed $N(0,\sigma^2)$. Then as Tauchen (1990) noted, we can solve the stochastic Euler equation for $\delta^*$ to obtain:

$$\delta^*(k,z) = zk^\alpha + k - \left(\frac{z^\rho \sigma^2 \beta\alpha}{2(1-\beta)}\right)^{\frac{1}{1-\alpha}}. \tag{2.62}$$

Note that the Euler equation has enabled us to obtain a closed-form solution for the optimal decision rule in a problem with two continuous state variables and one continuous control variable, even though it would not have been possible to deduce this directly from Bellman's equation. We will use this particular case as a further test problem in section 5 since it provides a direct way of judging the accuracy of various solution methods in the case of a multi-dimensional problem.

## 3. Numerical Solution Methods for General MDP's

Applying dynamic programming to solve general MDP problems forces us to confront a number of complications that are absent in LQ problems. First, the certainty equivalence principle no longer holds, so we need to specify the full conditional probability distribution $p(s'|s,d)$ instead of only its first two conditional moments in the LQ case. Second, since the value function $V(s)$ is an unknown nonlinear function of $s$ (as opposed to a quadratic function in the LQ case), virtually all solution methods require some sort of numerical integration to compute its conditional expectation $\int V(s')p(ds'|s,d)$. Third, the optimal decision rule $d = \delta(s)$ is generally not a linear function of $s$, so in general we need to resort to numerical optimization to compute the maximizing value of $d$ for any given state $s \in S$.

This section surveys the main solution methods for general MDP's. The nature of solution methods differ according to whether the problem involves a finite or infinite horizon and whether the state and control variables are continuous or discrete. We will consider each of these cases separately, beginning with the easiest case first. The final part of this section will consider "learning algorithms" for solving MDP's with partially observed states or unknown parameters in preferences $u$ or beliefs $p$.

### 3.1 Discrete Finite Horizon MDP's

The main numerical algorithm for solving finite horizon MDP's is simple backward recursion using equations $(2.5),\ldots,(2.8)$ of section 2. The integration operator in the finite state case reduces to simple summation over the $S$ possible states in the state space:

$$V_t^T(s) = \max_{d \in D_t(s)} \left[ u_t(s,d) + \beta \sum_{s=1}^{S} V_{t+1}^T(s') p(s'|s,d) \right]. \tag{3.1}$$

Assume for simplicity that the choice sets $D(s)$ contain a common finite number of possible decisions $D$. Then it is easy to see that computation of (3.1) for each state $s$ requires a total of $2(DS + D)$ additions, multiplications, and comparison operations, or a total of $2(DS^2 + DS)$ operations to compute the entire time $t$ value function $V_t^T$. Thus the total operation count in carrying out the dynamic programming procedure is is $2T(DS^2 + DS)$, which is dominated by the squared term for problems where $S$ is large. Note that the storage requirements are also $O(TDS^2)$, representing the space required to store the transition probabilities $\{p_t(s'|s,d)\}$. Thus the complexity of solving discrete finite horizon MDP's is $O(TDS^2)$, which implies that it is a member of the class $P$ of polynomial time problems if we measure the size of the problem by the number of discrete states $S$. Papadimitriou and Tsitsiklis 1987 proved that the MDP problem is *P-complete* (in the Turing model of computation) by showing that a known P-complete problem, the *Circuit Value Problem,* reduces to and thus can be solved as a special case of a general discrete finite horizon MDP problem. As discussed earlier, this implies that the MDP problem is a member of an equivalence class of the "hardest" polynomial time problems including linear programming. Since many computer scientists believe that $P \neq NC$,

the finding that the MDP problem is P-complete is taken as strong evidence that the problem cannot be effectively massively parallelized. However if we measure the size of the discrete MDP problem by the *dimension* of the state variable $s_t$ rather than by the total number of states $S$, then it is easy to see that the MDP problem is subject to the curse of dimensionality, and thus in the class $CD$ of exponential-time problems. For example, suppose there are $k_s$ state variables each taking on $S$ possible values. Then the total number of states is $S^{k_s}$, which increases exponentially in $k_s$. Similarly if there are $k_d$ control variables each of which takes on $D$ possible values, then the total number of decisions is $D^{k_d}$ which also increases exponentially in $k_d$. When problem size is measured this way, the MDP problem is not even NP-complete since the time required to verify a candidate solution produced by a "nondeterministic computer" increases exponentially rather than polynomially in $k_d$ or $k_s$. We summarize this discussion as:

**Theorem 5.1:** *The finite horizon discrete MDP problem is P-complete if problem size is measured by the pair $(D, S)$ representing the total number of possible decisions and states in the MDP. If problem size is measured by the pair $(k_d, k_s)$, resulting in an MDP problem with $(D^{k_d}, S^{k_s})$ decisions and states for some $D$ and $S > 0$, then the MDP problem is in the class CD of exponential-time problems. In the latter case, the MDP problem is not even NP-complete since the amount of time required to verify a candidate solution to the MDP increases exponentially in $(k_d, k_s)$.*

Notice that the main bulk of the work required to solve a discrete finite horizon MDP is computing the conditional expectations of the value function for each possible combination of the state $s$, decision $d$, and time period $t$: the remaining summation and maximization operations are of order $O(TDS)$ which are negligible compared to the $O(TDS^2)$ operations needed to compute the conditional expectations. There are four main ways to speed up the latter calculations: 1) exploiting special "sparsity structure" of the $\{p_t(s'|s,d)\}$ arrays, 2) using fast matrix multiplication algorithms, 3) using massive parallel processors, and 4) using parameteric methods to approximate solutions to MDP's with huge numbers of states and controls.

Many economic problems such as the optimal replacement problem and the optimal consumption and saving problems presented in section 2 have transition probabilities that are sparse and often have a highly recursive structure. For example, by restricting the distribution of investment returns to a finite interval one obtains a discrete representation of the consumption/savings problem with a banded transition probability matrix $p$. In general it is important to fully exploit one's *a priori* knowledge of the economic problem to reduce the time and storage requirements of the DP algorithms but determining the best way to do this gets more complicated in problems with multidimensional state variables. In order to apply formula (3.1) one must recode the multidimensional state vector $s_t$ to have a linearly ordered discrete state representation $s = 1, \ldots, S$. By doing this coding in the right way one can obtain a representation for the finite state transition probability matrix that has a desirable *sparsity pattern* that substantially reduces the burden of computing the conditional expectation of the value functions in (3.1). Rust (1991) provides an example of this approach in the case of an MDP model of optimal retirement behavior where the state variable $s_t$ has seven components,

i.e. $k_s = 7$. It turns out that for this problem any coding procedure yields a matrix representation for $p_t(s'|s,d)$ that is a direct product of a circulant matrix $C$, a banded matrix $B$, and a dense matrix $D$. Depending on how one orders these component matrices to form the overall transition probability matrix $p$ one obtains various sparsity patterns which are more or less amenable to rapid computation on parallel and vector computers. It turns out that the optimal ordering for a vector processor like the Cray-2 is $p = C \otimes B \otimes D$, which yields an upper block triangular representation for $p$ (for details see Rust, 1991). This strategy is even more effective for solving infinite horizon problems by the policy iteration approaches presented in the next section since policy iteration involves solution of systems of $S$ equations in $S$ unknowns, which takes $O(S^3)$ time using standard methods. Relative to naive policy iteration methods that treat $p$ as a dense matrix, Rust 1991 shows that using the optimal ordering $p = C \otimes B \otimes D$ fully exploits the sparsity pattern of $p$ and results in speedups of $\Theta(c^3)$ where $c$ is the order of the circulant matrix $C$. For the retirement problem considered in Rust 1991, this amounts to a speed up of 27,000 times relative to naive policy iteration algorithms that don't exploit the structure of $p$.

The other ways of speeding up the solution to discrete MDP problems - massive parallelism and fast matrix multiplication - also speed up the solutions to discrete infinite horizon MDP's and will be discussed in more detail when we discuss algorithms for solving infinite horizon MDP's in section 5.4. The use of parametric methods to approximate a discrete MDP with a large number of states and controls will be deferred to our discussion of solution methods for continuous MDP's in section 5.2. However it is perhaps more honest to speak of *potential* speedups since these approaches have not yet been widely applied, at least in economic applications. The use of fast matrix multiplication algorithms has been especially limited even though the speed-ups from using these algorithms are getting increasingly large. For example, the number of operations required to multiply two $n \times n$ matrices using standard matrix multiplication algorithms is $2n^3 - n^2$. Strassen's 1972 algorithm computes the product in $4.7n^{2.807}$ operations, Pan's 1980 algorithm requires $O(n^{2.795})$ operations, and as of 1987 the best known algorithm is due to Coppersmith and Winograd and requires only $O(n^{2.376})$ operations. However these fast matrix multiplication algorithms have larger space requirements and higher fixed costs relative to the conventional matrix multiplication algorithm. For example, Strassen's algorithm requires $11/3n^2$ memory locations which exceeds the $3n^2$ locations needed for conventional matrix multiplication, and Pan's algorithm requires 24 operations to multiply two $2 \times 2$ matrices compared to only 8 operations required by the conventional algorithm. However when $n = 128$ the traditional algorithm requires 2,097,152 operations versus 823,723 for Strassen's algorithm and 797,184 for Pan's algorithm.[27] The other not to be neglected fixed cost is the programming effort required implement these more complicated methods: this may in fact be *the* major reason why these methods have not been applied so far.

---

[27] These figures are taken from table 2.4.2 of Kronsjö, 1985.

Massive parallel processing is just beginning to be applied in computational economics. Note that the backward recursion approach to solving DP problems is inherently sequential and cannot be parallelized. However, the majority of the work is the $O(DS^2)$ operations necessary to compute the conditional expectations of the value function, and this can be parallelized. For example, if one has access to an expandable massive parallel processor with $S$-processors, then each separate processor can compute the (3.1) for each state $s = 1, \ldots, S$, so that the MDP problem can now be solved in $O(TDS)$ time on a parallel processor as opposed to $O(TDS^2)$ time on a standard "von Neumann machine". However it is an open question as to whether having access to more than $S$ processors will allow further acceleration in the solution time. The results of Papadimitriou and Tsitsiklis 1987 suggests that significant additional speedups are not possible, if by "significant" we mean an algorithm that can solve the MDP problem in polylog time using a polynomial number of processors. However in section 5.3 we show that we can in fact solve discrete infinite horizon MDP problems in $O(log(S)^2)$ time using $O(S^\omega)$ processors, where $\omega$ is the best available lower bound on fast matrix multiplication (currently $\omega = 2.376$). This does not contradict Papadimitriou and Tsitsiklis since the algorithm we present solves the MDP problem approximately rather than exactly and is based on the real instead of the Turing model of computation.

### 3.2 Continuous Finite Horizon MDP's

Recall that continuous MDP's are problems for which the state $s_t$ or control $d_t$ can assume a continuum of possible values. For concreteness, we will assume here that the state space $S$ is a compact, convex subset of $k_s$-dimensional Euclidean space $R^{k_s}$, and each constraint set $D(s)$ is a compact, convex subset of a $k_d$-dimensional Euclidean space $R^{k_d}$. The solution $\{V, \delta\}$ (see equations (2.5) to (2.8) of section 2) lies in an infinite-dimensional space, and clearly can't be stored in a digital computer. At best the solution can be approximated at a finite number of points, perhaps using interpolation to determine the solution at other points. As mentioned in the introduction, there are two main approximation strategies: discrete approximation and parametric approximation. Since the former approach is conceptually simpler, we will present it first.

Discrete approximation methods compute the value functions $V_t^T$ and decision rules $\delta_t^T$ at a finite grid of points in the state space $S$ and constraint sets $D_t(s)$. There are many variants of discrete approximation, but the general structure can be described as follows. Assume that we have carried out the discretization from time periods $T, T-1, \ldots, t+1$ and we now wish to compute an estimate of the value function $\hat{V}_t^T$ at time $t$. Discretization amounts to replacing the continuous-state version of the backward induction step

$$V_t^T(s) = \Gamma(V_{t+1}^T)(s) \equiv \max_{d \in D_t(s)} [u_t(s,d) + \beta \int V_{t+1}^T(s') p_{t+1}(ds'|s,d)], \tag{3.2}$$

by a discretized equivalent

$$\hat{V}_t^T(s) = \hat{\Gamma}_N(\hat{V}_{t+1}^T)(s) \equiv \max_{d \in D_t(s)} [u_t(s,d) + \beta \sum_{k=1}^{N} \hat{V}_{t+1}^T(s_k)\hat{p}_{t+1}(s_k|s,d)]. \tag{3.3}$$

Thus, we can view discrete approximation as replacing the true Bellman operator $\Gamma$ by an approximate "discrete" Bellman operator $\hat{\Gamma}_N$, but otherwise the backward induction process is identical. If we restrict attention to the finite set of grid points $\{s_1, \ldots, s_N\}$, then $\hat{\Gamma}_N$ can be viewed as a contraction mapping on $R^N$ that generates a discretized value function $\hat{V}_t^T$ which lives in $R^N$. However, one can also view the approximate operator $\hat{\Gamma}_N$ as a mapping on the infinite-dimensional space $B$ since $\hat{\Gamma}_N$ is *self-interpolating:* i.e. we can evaluate $\hat{\Gamma}_N(\hat{V}_t^T)$ at any point $s$ in the continuous state space $S$ and not just at the finite set of grid points $\{s_1, \ldots, s_N\}$. Thus, if $\hat{p}_{t+1}(s_k|s,d)$ satisfies weak regularity properties (i.e. is a measurable or continuous function of $s$ for each grid point $s_k$) then the approximate Bellman operator is mathematically completely well defined as a contraction mapping directly on $B$: $\hat{\Gamma}_N : B \to B$. This interpretation turns out to be a key insight to establishing the consistency and rate of convergence of various discrete approximation methods.

In the course of backward induction approximation errors generally compound: i.e. $|\hat{V}_t^T - V_t^T| \geq |\hat{V}_{t+1}^T - V_{t+1}^T|$. The following Lemma provides a simple sufficient condition guaranteeing that $|\hat{V}_t^T - V_t^T| \leq \epsilon$ uniformly for all $t \in \{1, \ldots, T\}$.

**Lemma 5.1:** *Suppose there exists an integer $\overline{N}(\epsilon, \beta)$ such that for all $N\overline{N}(\epsilon, \beta)$ we have:*

$$\left|\hat{\Gamma}_N(W) - \Gamma(W)\right| \leq (1-\beta)\epsilon, \tag{3.4}$$

*uniformly for all $W \in B$ satisfying:*

$$|V| \leq K \equiv \max_{t \in \{1,\ldots,T\}} \sup_{s \in S} \sup_{d \in D_t(s)} \frac{|u_t(s,d)|}{(1-\beta)}. \tag{3.5}$$

*If we begin the backward induction using any estimate of the terminal value function $\hat{V}_T^T$ satisfying:*

$$\begin{aligned} |\hat{V}_T^T - V_T^T| &\leq \epsilon, \\ |\hat{V}_T^T| &\leq K/(1-\beta), \end{aligned} \tag{3.6}$$

*then $\hat{V}_t^T$ is uniformly within $\epsilon$ of $V_t^T$ for all $t$:*

$$\max_{t \in \{1,\ldots,T\}} |\hat{V}_t^T - V_t^T| \leq \epsilon. \tag{3.7}$$

**Proof:** We prove the result by induction on $t$. Starting at $t = T-1$ suppose we choose $N$ and $\hat{V}_T^T$ satisfying (3.4), (3.5), and (3.6). Then we have:

$$\begin{aligned}\left|\hat{V}_{T-1}^T - V_{T-1}^T\right| &= \left|\hat{\Gamma}_N(\hat{V}_T^T) - \Gamma(V_T^T)\right| \\ &= \left|\hat{\Gamma}_N(\hat{V}_T^T) - \hat{\Gamma}_N(V_T^T) + \hat{\Gamma}_N(V_T^T) - \Gamma(V_T^T)\right| \\ &\leq \left|\hat{\Gamma}_N(\hat{V}_T^T) - \hat{\Gamma}_N(V_T^T)\right| + \left|\hat{\Gamma}_N(V_T^T) - \Gamma(V_T^T)\right| \\ &\leq \beta\epsilon + (1-\beta)\epsilon \\ &= \epsilon.\end{aligned} \tag{3.8}$$

This argument can be repeated for each $t = T-1, T-2, \ldots, 1$ provided we can show that for each $t$ we have $|\hat{V}_t^T| \leq K/(1-\beta)$. However this follows follows from Lemma 2.3.

Lemma 5.1 shows that the problem of proving the consistency of various discrete approximation methods reduces to the problem of proving that the approximate Bellman operator $\hat{\Gamma}_N$ converges to $\Gamma$ uniformly for $V \in B(0,K)$:

$$\lim_{N\to\infty} \sup_{|V|\leq K} \left|\hat{\Gamma}_N(V) - \Gamma(V)\right| = 0. \tag{3.9}$$

In order to compare the computational requirements of various discrete approximation methods we will also need to determine the *rate of convergence* of $\hat{\Gamma}_N$ to $\Gamma$.

We now provide a brief overview of four different discrete approximation methods that we analyze in this section. The key problems in finding an effective discretization procedure are 1) choosing the $N$ grid points $\{s_1, \ldots, s_N\}$ at which to evaluate $V_{t+1}^T$ in the summation in (3.3), and 2) choosing an appropriate discretized version $\hat{p}_{t+1}$ of the continuous transition probability $p_{t+1}$. It turns out that the first problem is the most difficult. We will consider four different strategies for selecting the grid points $\{s_1, \ldots, s_N\}$: 1) uniform grid, 2) quadrature abscissa, 3) random draws from $S$, and 4) "optimal" grid points. The first two methods are subject to the curse of dimensionality of multivariate integration discussed in section 3: i.e. the number of grid points $N$ required to approximate the Bellman operator $\Gamma$ defined in (3.2) to within a maximum error of $\epsilon$ is $N = O(1/\epsilon^{k_s})$ where $k_s$ is the dimension of the state vector $s$. Furthermore the worst case complexity bounds established by Chow and Tsitsiklis 1989 show that the general MDP problem is in the class *CD*. That is, it is impossible to get around the curse of dimensionality, at least worst case basis using deterministic methods. The last two discrete approximation methods do succeed in breaking the curse of dimensionality, however only for the subclass of *discrete decision processes* (DDP's), i.e. MDP's with finite choice sets and continuous multidimensional state space $S$. Rust (1994) developed a simple monte carlo integration algorithm and proved that the (randomized) worst case complexity of finite horizon continuous DDP's is $comp^{wor\text{-}ran}(\epsilon, k_s) = O(1/\epsilon^4)$. Notice that $1/\epsilon^4 = 1/\epsilon^2 \times 1/\epsilon^2$ where $1/\epsilon^2$ is the order of the number of operations required by monte carlo integration in order to estimate an integral to within an expected absolute error of $\epsilon$. It is

interesting to note that the MDP problem, which is both nonlinear (due to the presence of the $max$ operator in the Bellman equation), and infinite-dimensional (since it effectively requires calculation of a continuum of integrals for different conditioning values $(s,d)$) nevertheless enjoys the same rate of convergence as is achieved in monte carlo integration, i.e. the expected absolute error decreases (uniformly) at rate $1/\sqrt{N}$ independent of the dimension $k_s$. We have not yet established the lower bound on the complexity of DDP's using random algorithms. However Bakhvalov 1959 and Novak 1987 have proven that the lower bound on the randomized complexity of multivariate integration of $r$-times differentiable functions is $comp^{wor\text{-}ran}(\epsilon, k_s) = \Theta(1/\epsilon^m)$ where $m = 2k_s/(2r + k_s)$. Note that $r = 1$ for the class of Lipschitz continuous DDP problems that we consider, so that use of Bakhvalov's integration algorithm yields a reduction in the exponent $m$ from 2 to $2/(2/k+1)$. We conjecture that these results can be used to derive lower bounds on the complexity of DDP's. The final approach, "optimal grid points" uses deterministic instead of random placement of the grid points. We shall consider two possibilities for $\{s_1, \ldots, s_N\}$: 1) the Hammersley points, and 2) the hyperbolic cross points. Woźniakowski 1991 and Paskov 1992 have shown that these grid points are asymptotically "optimal" in the sense that they attains the average case complexity bound for multivariate integration in the limit as $N \to \infty$. Recall that calculation of average case complexity bounds requires a specification of a prior distribution over problem elements. The standard choice of prior in this literature is "Wiener sheet" measure. Under this prior, the average case complexity of multivariate integration is given by:

$$comp^{avg\text{-}det}(\epsilon, k_s) = \Theta\left(\frac{1}{\epsilon} log\left(\frac{1}{\epsilon}\right)^{(k_s-1)/2}\right). \tag{3.10}$$

We show that Wiener sheet measure can also be adapted to provide a prior distribution over utility functions $u_t$ and transition probabilities $p_t$, and we conjecture that these average case complexity bounds can be adapted to derive lower bounds on the average complexity of DDP's.

In order to prove the consistency and asymptotic properties of the various discretization procedures presented below we need to impose some additional regularity conditions on the state space $S$ and the functions $\{u_t, p_t | t = 1, \ldots, T\}$. The following assumptions are stronger than necessary to prove many of the results stated below, but we will impose them here in order to simplify the proofs and unify the exposition. In particular, the assumption that the state and control spaces are subsets of the $k_s$ and $k_d$-dimensional unit cubes is without loss of generality: as long as the original $S$ and $D$ sets are compact, one can always do a change of coordinates to map these sets into the unit cubes.

**(A1)** *$S$ is a Borel subset of $[0,1]^{k_s}$; $D_t(s)$ is a compact subset of $[0,1]^{k_d}$ $\forall s \in S$, and $\forall t \in \{1, \ldots, T\}$.*

**(A2)** *$p_t(ds'|s,d)$ has a continuous and uniformly bounded density with respect to Lebesgue measure on $S$ for each $t \in \{1, \ldots, T\}$, $d \in D_T(s)$ and $s \in S$.*

**(A3)** $\{u_t | t = 1 \ldots, T\}$ *is Lipschitz continuous family of functions with Lipschitz bound* $K_u$.[28]

**(A4)** $\{p_t | t = 1, \ldots, T\}$ *is a Lipschitz continuous family of functions, with Lipschitz bound* $K_p$.

**(A5)** *The mapping* $s \to D(s)$ *is a Lipschitz continuous correspondence.*[29]

**Uniform grid points.** Following the recent approach of Chow and Tsitsiklis 1991 partition the $k_s$-dimensional cube $[0,1]^{k_s}$ into equal subcubes of length $h$ on each side. Assuming that $1/h$ is an integer, this results in a total of $N = (1/h)^{k_s}$ subcubes. Let $S_h$ denote the partition of $S$ induced by this partition of the unit cube. If $D_t(s)$ contains an infinite number of points for any $s$, do a similar discretization of the choice sets, using a partition of $[0,1]^{k_d}$ into $N = (1/h)^{k_d}$ equal subcubes of length $h$ on each side, which induces a corresponding partition on $D_t(s)$ which we denote as $D_{t,h}(s)$. Finally, let $s_k$ denote an abritrary element (grid point) in the $k^{th}$ partition element of $S_h$ and let $k(s)$ denote the partition element of $S_h$ that contains a given point $s \in S$. Define discretized utility and transition probabilities by:

$$\begin{aligned} u_{t,h}(s,d) &= u_t(s_{k(s)}, d) \\ p_{t,h}(s'|s,d) &= \frac{p_t(s_{k(s')}|s_{k(s)}, d)}{\int p_t(s_{k(s')}|s_{k(s)}, d)ds'} \end{aligned} \tag{3.11}$$

where the normalization of the second equation insures that $p_{t,h}$ is a well defined transition probability density on $S$. Note that (3.11) defines $u_{t,h}$ and $p_{t,h}$ as discrete step function approximations to $u_t$ and $p_t$.[30] This is primarily for notational convenience: in practice one would probably want to use spline interpolations so that $u_{t,h}$ and $p_{t,h}$ are also Lipschitz continuous functions of $(s', s, d)$. Given these objects, we can define the discretized Bellman operator by

$$\begin{aligned} \hat{\Gamma}_h(V)(s) &= \max_{d \in D_{t,h}(s)} [u_{t,h}(s,d) + \beta \int V(s') p_{(t+1),h}(s'|s,d)ds'] \\ &= \max_{d \in D_{t,h}(s)} [u_{t,h}(s,d) + \frac{\beta}{N} \sum_{k=1}^{N} V(s_k) p_{(t+1),h}(s_k|s_{k(s)}, d)], \end{aligned} \tag{3.12}$$

where the second equality holds whenever $V$ is a step function taking value $V(s_k)$ in partition element $k$.[31] The following consistency result for the uniform discretization procedure, adapted from Theorem 3.1 of Chow and Tsitsiklis 1991, is given below:

---

[28] Lipschitz continuity means that $|u_t(s,d) - u_t(s',d')| \leq K_u |(s,d) - (s',d')|$, for all feasible pairs $(s,d)$ and $(s',d')$.

[29] This means that for any $s, s' \in S$ and any $d' \in D_t(s')$ there exists some $d \in D_t(s)$ such that $|d - d'| \leq K|s - s'|$.

[30] In some cases we will need to do numerical integration just to determine the values of $p_{t,h}$ as can be seen from formula (3.11).

[31] Note that in this case $\hat{\Gamma}_h$ should be regarded as an operator on the Banach space $B$ of bounded measurable functions of $s$ under the essential supremum norm rather than the space of all continuous, bounded functions of $s$.

**Theorem 5.2:** *There exist constants $K_1$ and $K_2$ such that for all $h$ sufficiently small and all $V \in B$ we have:*

$$|\Gamma(V) - \hat{\Gamma}_h(V)| \leq \frac{(K_1 + \beta K_2 |V|)}{h}. \tag{3.13}$$

Notice that in order to obtain consistency, we must choose $h$ sufficiently small, say $h < \epsilon$. This implies that the discretized MDP requires $S = O(1/\epsilon)^{k_s}$ states and $D = O(1/\epsilon)^{k_d}$ decisions. Recalling that $O(TDS^2)$ operations are required to solve a discrete, finite horizon MDP, it follows that using the uniform discretization to compute an $\epsilon$-approximation to the continuous finite horizon MDP problem requires $O(T/\epsilon^{(2k_s+k_d)})$ operations, so this method is subject to the curse of dimensionality.

**Quadrature abscissa.** Following the approach of Tauchen and Hussey 1991, consider the problem of approximating the integral of a function $g : S \to R$ with respect to a weighting function $\omega : S \to R$ by choosing $N$ grid points $\{s_1, \ldots, s_N\}$ and corresponding weights $\{w_1, \ldots, w_N\}$ such that

$$\int g(s)\omega(s)ds \simeq \sum_{k=1}^{N} g(s_k)w_k. \tag{3.14}$$

The weighting function will typically be a probability density function representing an "importance sampling" weighting reflecting the analyst's *a priori* knowledge about the parts of the state space that are most important for calculating the integral. In the special case $k_s = 1$ the Gaussian quadrature formula determines $\{s_1, \ldots, s_N\}$ and $\{w_1, \ldots, w_N\}$ by forcing (3.14) to be exact for all polynomials of degree less than or equal to $2N - 1$.[32] The resulting grid points and weights depend only on the weighting function $\omega(s)$ and not the function $g(s)$ being integrated. Notice that the weights $\{w_1, \ldots, w_N\}$ generally do not satisfy $w_k = \omega(s_k)$, although they are all guaranteed to be positive. Applying Gaussian quadrature to approximate the conditional expectation of the value function $V$ we obtain

$$\int V(s')p_t(s'|s,d)ds' = \int V(s')\frac{p_t(s'|s,d)}{\omega(s')}\omega(s')ds' \simeq \sum_{k=1}^{N} V(s_k)\frac{p_t(s_k|s,d)}{\omega(s_k)}w_k. \tag{3.15}$$

The quadrature weights and abscissa allow us to define an $N$-state Markov chain with state space equal to the quadrature abscissa $\{s_1, \ldots, s_N\}$ with transition probability $p_{t,N}$ defined probability by a simple normalization:

$$p_t^N(s_k|s_j,d) = \frac{p_t(s_k|s_j,d)w_k/\omega(s_k)}{\sum_{i=1}^{N} p_t(s_i|s_j,d)w_i/\omega(s_i)}, \quad j,k = 1,\ldots,N. \tag{3.16}$$

Now we can define a discretized Bellman operator $\hat{\Gamma}_N : B \to B$ (where in this case $B$ is the Banach space of continuous functions on $S$ under the sup norm) by:

$$\hat{\Gamma}_N(V)(s) = \max_{d \in D_t^N(s)} [u_t(s,d) + \beta \sum_{k=1}^{N} V(s_k)p_{t+1}^N(s_k|s,d)], \tag{3.17}$$

---

[32] Multivariate quadrature, $k_s > 1$ is more involved. See Tauchen and Hussey 1991 and Stroud, 1971 for a description of various extensions of the quadrature approach to multidimensional problems.

where $D_{tN}(s)$ denotes a finite choice set with $N$ points, a discretized version of $D_t(s)$ using the same procedure defined for the uniform discretization procedure above.[33] Adapting Tauchen and Hussey's Theorem 4.2 we have:

**Theorem 5.3:** *There exist constants $K_1$ and $K_2$ such that for all $V \in B = C[0,1]$ and all $N$ sufficiently large* we have:

$$|\Gamma(V) - \hat{\Gamma}_N(V)| \leq (K_1 + \beta K_2)/N. \tag{3.18}$$

Note that this is the same basic bound as obtained in the case of uniform discretization (Theorem 2) with $1/N$ playing the same role as the grid size parameter $h$. Thus, just as in the case of uniform discretization, the quadrature approach produces a consistent estimate of the true value function $V$. Note that although Gaussian quadrature possesses several optimum properties (Davis and Rabinowitz, 1975) the $\epsilon$-complexity of the quadrature algorithm is the same as uniform discretization. More generally, the number of grid points required for a $k_s$-dimensional quadrature procedure to approximate an integral of an arbitrary once differentiable function to within $\epsilon$ is $N = \Theta(1/\epsilon^{k_s})$, the same as for uniform discretization.[34] Thus, use of a quadrature rule does not break the curse of dimensionality of the continuous MDP problem, at least on a worst case basis.

Indeed, the results of Chow and Tsitsiklis 1989,1991 indicate that the $O(T/\epsilon^{(2k_s+k_d)})$ complexity bound is also also a lower bound on the complexity of the general MDP problem and that their uniform discretization procedure is an approximately optimal algorithm.[35] Thus there is *no* algorithm that is capable of breaking the curse of dimensionality of solving general MDP's, at least using deterministic algorithms and measuring complexity on a worst case basis. We can think of their lower bound as consisting of a product of the lower bounds on complexity of three "subproblems" encountered in approximate solution of a general MDP: 1) a $\Theta(1/\epsilon^{k_d})$ worst case complexity bound for approximate solution of the nonlinear optimization problem that determines the optimal decision rule $d = \delta_t^T(s)$ for any $s \in S$, 2) a $\Theta(1/\epsilon^{k_s})$ worst case complexity bound on the approximate calculation of the integral of $V_t^T$ for any $(s,d)$, and 3) a $\Theta(1/\epsilon^{k_s})$ complexity bound on the approximation of the function $\Gamma(V_t^T)$ from information of its values at a finite number of points $\{s_1, \ldots, s_N\}$.

**Random points from $S$.** Results by Yudin and Nemivorski 1978 and Traub, Waslikowski and Woźniakowski 1988 show that randomization does not succeed in breaking the curse of dimensionality for the optimization and approximation subcomponents of the MDP problem, so that use of random algorithms will not succeed in breaking

---

[33] Of course there is no reason beyond notational simplicity to require that the number of points $N$ used for the quadrature abscissa equals the number of points for the choice set discretization as long as both tend to infinity and the maximum grid size of the discretization of $D$ tends to zero as $N \to \infty$.

[34] See Traub, Wasilikowski and Woźniakowski 1987. This result assumes that the function being integrated has are least one derivative that is bounded in $L_p$ norm. If the function has $r > 1$ derivatives, then the number of grid points needed for an $\epsilon$-approximation of the integral is given by $N = O(1/\epsilon^m)$ where $m = k_s/r$.

[35] Chow and Tsitsiklis's results are for infinite horizon MDP's, although their approach extends in a straightforward method to finite horizon MDP's.

the curse of dimensionality of solving general finite horizon MDP's. However randomization does break the curse of dimensionality of the multivariate integration problem. Rust (1994) used this result to construct a variant of backward induction using a "random Bellman operator" that does succeed in breaking the curse of dimensionality for a subclass of DDP's, i.e. the class of MDP's with finite choice sets defined in section 2.6. The basic idea it to use monte carlo integration rather than a deterministic integration algorithm such as quadrature or uniform discretization. Monte carlo integration of a function $g : S \to R$ with respect to a probability density $w : S \to R$ is given by:

$$\int g(s)w(s)ds \simeq \frac{1}{N}\sum_{k=1}^{N} g(s_k), \tag{3.19}$$

where $\{s_1, \ldots, s_N\}$ are *IID* draws from the density $\omega$. Under very weak conditions (i.e. $g$ is absolutely integrable with respect to $\omega$), the strong law of large numbers implies that the sample average converges with probability 1 to its expected value, which is the integral on the left hand side of (3.19). More significantly, the Central Limit Theorem implies that the sample average converges to the sample mean at rate $O_p(1/\sqrt{N})$ independently of the dimension $k_s$ of the state space $S$. For this reason economists have recently been attracted to monte carlo methods. To our knowlege Wolpin and Keane 1993 is the first actual application of the monte carlo approach to compute solutions to large scale continuous finite horizon MDP's. Since their approach also involves the use of parametric approximation of the value function we will defer a discussion of their approach to the section on parametric approximation methods below.

Let $\{s_1, \ldots, s_N\}$ be *IID* draws (with respect to Lebesgue measure) from the $k_s$-dimensional unit cube $[0,1]^{k_s}$. We can use these random draws to define the *random Bellman operator* $\tilde{\Gamma}_N : B \to B$ (where $B = C[0,1]^{k_s}$) by

$$\tilde{\Gamma}_N(V)(s) = \max_{d \in D_t^N(s)} [u_t(s,d) + \frac{\beta}{N}\sum_{k=1}^{N} V(s_k)p_t(s_k|s,d)], \tag{3.20}$$

where $D_t^N(s)$ denotes the discretized choice set used in the uniform discretization approach above. Note that in general that $p_t(s_k|s_j,d)/N$ does not define a transition probability over the sample points $\{s_1, \ldots, s_N\}$: the probability does not sum to 1. In some cases it is helpful to define a modified form of the random Bellman operator, $\hat{\Gamma}_N$, with a legitimate transition probability $p_{t,N}$ defining the conditional expectation operator which is guaranteed to be a contraction mapping with probability 1:

$$\hat{\Gamma}_N(V)(s) = \max_{d \in D_t^N(s)} [u_t(s,d) + \frac{\beta}{N}\sum_{k=1}^{N} V(s_k)p_{(t+1),N}(s_k|s,d)], \tag{3.21}$$

where $p_{t,N}$ is defined by:

$$p_{t,N}(s_k|s,d) = \frac{p_t(s_k|s,d)/N}{\sum_{i=1}^{N} p_t(s_i|s,d)/N}. \tag{3.22}$$

A simple application of the strong law of large numbers shows that the denominator of (3.22) converges with probability 1 to the constant 1, so the modified version of $\tilde{\Gamma}_N$ using the transition probability $p_{t,N}$ converges to the same limit as the original definition in (3.20).

**Theorem 5.4:** *The random operators $\tilde{\Gamma}_N$ and $\hat{\Gamma}_N$ are consistent: i.e. $\forall V \in B$ we have:*

$$\begin{aligned} Pr\left\{\lim_{N\to\infty} |\Gamma(V) - \tilde{\Gamma}_N(V)| = 0\right\} = 1, \\ Pr\left\{\lim_{N\to\infty} |\Gamma(V) - \hat{\Gamma}_N(V)| = 0\right\} = 1. \end{aligned} \tag{3.23}$$

The proof of Theorem 5.4 is given in appendix 1.

Although Theorem 5.4 establishes the consistency of the the randomized discretization method (3.20), it tells us little about its computational complexity. In order to determine this we need to establish the rate of convergence of the randomized estimate $\tilde{\Gamma}_N(V)$ to the true solution $\Gamma(V)$, we need to establish the rate of convergence. The Central Limit theorem can be used to show that for any fixed $(s,d)$ we have

$$\sqrt{N}[\tilde{h}_N(s,d) - h(s,d)] \Rightarrow \tilde{Z}, \tag{3.24}$$

where $\tilde{Z}$ is Gaussian random variable with mean 0 and variance $\sigma^2$. The function $\tilde{h}_N$ and $h$ entering (3.24) are given by

$$\begin{aligned} \tilde{h}_N(s,d) &= u_t(s,d) + \frac{\beta}{N}\sum_{k=1}^{N} V(s_k)p_{t+1}(s_k|s,d) \\ h(s,d) &= u_t(s,d) + \beta\int V(s')p_{t+1}(s'|s,d)ds' \end{aligned} \tag{3.25}$$

Since the (stochastic) rate of convergence of $\tilde{h}_N(s,d)$ to $h(s,d)$, is $O_p(1/\sqrt{N})$ independent of the dimension $k_s$ of the state space $S$, randomization succeeds in breaking the curse of dimensionality associated with multivariate integration problem. However the randomization does not solve the curse of dimensionality of the $k_d$-dimensional constrained optimization problem in (3.20). It is easy to see, therefore, that the randomized algorithm requires $O(T/\epsilon^m)$ operations to compute an $\epsilon$-approximation (which in the randomized case means that $E\{|\tilde{\Gamma}_N(V) - \Gamma(V)|^2\} \leq \epsilon$), where $m = k_d + 2$. This is better than the worst case complexity of the deterministic algorithms which have an exponent $m = k_d + 2k_s$ but still doesn't succeed in breaking the curse of dimensionality for general MDP's with continuous choice sets $\{D_t\}$.

Rust (1994) has shown that randomization does break the curse of dimensionality for the class of DDP's. To simplify exposition, we will assume that there is a common finite choice set $D$ in each state $s \in S$.[36]

**Theorem 5.5:** *Randomization breaks the curse of dimensionality of solving DDP's: i.e. the worst case complexity of the class of randomized algorithms for solving the DDP problems satisfying $(A1),\ldots,(A5)$ is given by:*

$$comp^{wor\text{-}ran}(\epsilon, k_s) = O\left(\frac{T}{\epsilon^4}\right). \tag{3.26}$$

---

[36] This assumption is without loss of generality since we can always transform a DDP problem with state dependent choice sets $D(s)$ into an equivalent problem with a state independent choice set $D$ by defining $u_t(s,d)$ to be a very large negative number for any $d \in D_t(s)$.

### 3.3 Discrete Infinite Horizon MDP's

As mentioned in section 2, the solution to infinite horizon MDP problems is mathematically equivalent to computing a fixed point to Bellman's equation $V = \Gamma(V)$. We briefly review the standard methods use to compute this fixed point for discrete MDP's with finite state space and choice sets.

Successive Approximations. [37] Starting with an initial guess $V_0$ of the solution of Bellman's equation, one simply iterates the Bellman operator

$$V_{k+1} = \Gamma(V_k) = \Gamma^k(V_0), \tag{3.27}$$

stopping the iterations at the smallest $k$ such that $|V_k - V_{k-1}| < \epsilon$ where $\epsilon$ is a pre-defined solution tolerance. In the case wher $V_0 = 0$, this method is equivalent to solving an approximate finite-horizon problem by backward induction. Using the contraction property, it is easy to show that the error at the $k^{th}$ iteration satisfies:

$$|V_k - V^*| \leq \beta^k |V_1 - V_0|/(1-\beta), \tag{3.28}$$

i.e. the sequence $\{V_k\}$ converges geometrically at rate $\beta$ to the fixed point $V^*$.[38] In problems where decision intervals are relatively short (such as monthly or weekly), the relevant value of $\beta$ will be very close to one, implying that successive approximations will converge unacceptably slowly. Indeed, using tha *a priori* error bound (2.27) we find that $T$ successive approximation steps are required to obtain an $\epsilon$-approximation, where $T$ is given by:

$$T = O\left(\frac{1}{log(\beta)} log\left(\frac{1}{(1-\beta)\epsilon}\right)\right). \tag{3.29}$$

**Acceleration via McQueen-Porteus Error Bounds.** In certain circumstances the method of successive approximations can be significantly accelerated by employing the *McQueen-Porteus Error Bounds:*

$$\Gamma^k(V) + \underline{b}_k e \leq V^* \leq \Gamma^k(V) + \bar{b}_k e, \tag{3.30}$$

where $e$ denotes an $S \times 1$ vector of 1's, and:

$$\begin{aligned} \underline{b}_k &= \beta/(1-\beta)\, min[\Gamma^k(V) - \Gamma^{k-1}(V)] \\ \bar{b}_k &= \beta/(1-\beta)\, max[\Gamma^k(V) - \Gamma^{k-1}(V)]. \end{aligned} \tag{3.31}$$

The contraction property guarantees that $\underline{b}_k$ and $\bar{b}_k$ approach each other geomerically at rate $\beta$. That fact that the fixed point $V^*$ is bracketed within these bounds suggests that we can obtain an improved estimate of $V^*$ by terminating the iterations (3.27) when $|\bar{b}_k - \underline{b}_k| < \epsilon$, setting the final estimate of $V^*$ to be the median bracketed value:

$$\hat{V}_k = \Gamma^k(V_0) + \left(\frac{\bar{b}_k + \underline{b}_k}{2}\right) e. \tag{3.32}$$

---

[37] The method also goes by names *value iterations, contraction iterations, backward induction,* or simply *dynamic programming.*

[38] This is sometimes referred to as *linear* convergence since (3.28) implies that the ratios of approximation errors satisfy $|V_k - V^*|/|V_{k-1} - V^*| \leq \beta$.

Bertsekas (1987) p. 195 shows that the rate of convergence of $\{\hat{V}_k\}$ to $V^*$ is geometric at rate $\beta|\lambda_2|$, where $\lambda_2$ is the subdominant eigenvalue of $P_{\delta^*}$. Thus, in cases where $|\lambda_2| < 1$, the use of the error bounds can lead to significant speed-ups in the convergence of successive approximations at essentially no extra computational cost. However in problems where $P_{\delta^*}$ has multiple ergodic sets, $|\lambda_2| = 1$, and the error bounds will not lead to an appreciable speed improvement as illustrated in computational results in table 5.2 of Bertsekas (1987).

**Policy Iteration Methods.** In relatively small scale problems ($S < 500$) the method of *Policy Iteration* is generally the fastest method for computing $V^*$ and the associated optimal decision rule $\delta^*$ provided the discount factor is sufficiently large ($\beta > .95$). The method starts by choosing an arbitrary initial policy, $\delta_0$.[39] Next a *policy valuation* step is carried out to compute the value function $V_{\delta_0}$ implied by the stationary decision rule $\delta_0$. This requires solving the linear system (2.35). Once the solution $V_{\delta_0}$ is obtained, a *policy improvement* step is used to generate an updated policy $\delta_1$:

$$\delta_1(s) = \underset{1 \le d \le D(s)}{argmax} \left[ u(s,d) + \beta \sum_{s'=1}^{S} V_{\delta_0}(s') p(s'|s,d) \right]. \tag{3.33}$$

Given $\delta_1$ one continues the cycle of policy valuation and policy improvement steps until the first iteration $k$ such that $\delta_k = \delta_{k-1}$ (or alternatively $V_{\delta_k} = V_{\delta_{k-1}}$). It is easy to see from (2.35) and (3.33) that such a $V_{\delta_k}$ satisfies Bellman's equation (2.33), so that by Theorem 2.3 the stationary Markovian decision rule $\delta^* = \delta_k$ is optimal. Policy iteration always generates an improved policy, i.e. $V_{\delta_k} \geq V_{\delta_{k-1}}$. To see this, note that by definition of the policy improvement step we have:

$$\begin{aligned} G_{\delta_k}(V_{\delta_{k-1}})(s) &= \max_{1 \le d \le D(s)} \left[ u(s,d) + \beta \sum_{s'=1}^{S} V_{\delta_{k-1}}(s') p(s'|s,d) \right] \\ &\geq \left[ u(s, \delta_{k-1}(s)) + \beta \sum_{s'=1}^{S} V_{\delta_{k-1}}(s') p(s'|s, \delta_{k-1}(s)) \right] \\ &= V_{\delta_{k-1}}(s). \end{aligned} \tag{3.34}$$

Since the operator $G_\delta$ is monotonic, we have:

$$V_{\delta_{k-1}} \leq G_{\delta_k}(V_{\delta_{k-1}}) \leq \cdots \leq G_{\delta_k}^N(V_{\delta_{k-1}}), \tag{3.35}$$

and since $G_\delta$ is a contraction mapping it follows that:

$$\lim_{N \to \infty} G_{\delta_k}^N(V_{\delta_{k-1}}) = V_{\delta_k} \geq V_{\delta_{k-1}}. \tag{3.36}$$

Policy iteration always generates a strict improvement in $V_{\delta_k}$, since if $V_{\delta_k} = V_{\delta_{k-1}}$ the method has already converged. Since there are only a finite number $D(1) \times \cdots \times D(S)$ of feasible stationary Markov policies, it follows that policy iteration always converges to the optimal decision rule $\delta^*$ in a finite number of iterations.

---

[39] One obvious choice is $\delta_0(s) = argmax_{1 \le d \le D(s)}[u(s,d)]$.

Policy iteration is able to discover the optimal decision rule after testing an amazingly small number of trial policies $\delta_k$: the method typically converges in under 20 iterations. However the amount of work per iteration is larger than for successive approximations. Since the number of algebraic operations needed to solve the linear system (2.35) for $V_{\delta_k}$ is of order $S^3$, the standard policy iteration algorithm becomes impractical for $S$ much larger than 1,000.[40] To solve very large scale MDP problems, it seems that the best strategy is to use policy iteration, but to only attempt to approximately solve for $V_\delta$ in each policy evaluation step (2.35).

The method of *modified policy iteration* uses $N$ successive approximation steps to compute an approximate fixed point $V_\delta = G_\delta(V_\delta)$ rather than computing the exact solution $V_\delta = [I - \beta P_\delta]^{-1} u_\delta$. The successive approximations iterations for the operator $G_\delta$ can be further accelerated by using the McQeen-Porteus error bounds described above. The following Theorem of Puterman and Shin (1978) shows that asymptotically, each modified policy iteration is equivalent to performing $N + 1$ successive approximation steps.

**Theorem 5.6:** *Let $\delta^*$ be an optimal policy for a stationary discounted MDP problem, and let $\delta_k$ be the decision rule generated at step $k$ of the modified policy iteration algorithm that uses $N$ successive approximation steps as an approximate solution for $V_{\delta_k}$. If:*

$$\lim_{n\to\infty} |P_{\delta_k} - P^*_\delta| = 0,$$

then

$$\overline{\lim_{n\to\infty}} \frac{|V_{n+1} - V^*|}{|V_n - V^*|} = \beta^{N+1}. \tag{3.37}$$

Thus, modified policy iteration can be thought of as an accelerated form of successive approximations. It can be effective in problems where $\beta$ is relatively low, although in problems where $\beta$ is close to 1 it tends to suffer from the same slow convergence properties as successive approximations. However our numerical results in section 3.5 demonstrate that when the McQueen-Porteus error bounds are used to accelerate the iterations, the method retains the rapid convergence properties of standard policy iteration, but the amount of work required in each (approximate) policy evaluation step is much less.

Policy iteration by *state aggregation* methods computes the exact solution $\overline{v}_\delta$ to a lower dimensional version of (2.35) that groups each of the $S$ elemental states of the original problem into $M$ aggregate states. In an *aggregation step* we choose a partition of the state space $S_1, \ldots, S_M$ (methods for choosing the partition will be described shortly). If aggregate state $i$ has $N_i$ elements, we can define a transition probability matrix $\overline{P}_\delta$ on the aggregate d state space by:

$$\overline{P}_\delta(i,j) = \frac{1}{N_i} \sum_{s\in S_i} \sum_{s'\in S_j} p(s'|s,\delta(s)). \tag{3.38}$$

---

[40] Supercomputers using combinations of vector processing and multitasking can now solve dense linear systems exceeding 1,000 equations and unknowns in under 1 CPU second. See for example, Dongarra (1986).

Let $\overline{u}_\delta$ be an $M \times 1$ vector of the average utilities in each of the $M$ aggregate states. Then if $M$ is sufficiently small, one can use standard Gaussian elimination algorithms to rapidly compute the exact solution $\overline{v}_\delta = [I - \beta \overline{P}_\delta]^{-1} \overline{u}_\delta$ to the $M$-state problem. In a *disaggregation step* we use $\overline{v}_\delta$ of to construct an approximation of the $S$-state value function $V_\delta$. The partition $S_1, \ldots, S_M$ can be represented by an $S \times M$ *partition matrix* $W$ defined by:

$$W_{ij} = \begin{cases} 1 & \text{if } i \in S_j \\ 0 & \text{otherwise.} \end{cases} \tag{3.39}$$

Using $W$ we can then compute an approximate solution $\hat{V}_\delta = W\overline{v}_\delta$ to the original problem (2.35). Notice this approximate solution will be a step function equal to $\overline{v}_\delta(i)$ for each elemental state $s \in S_i$.

Bertsekas and Castañon (1989) have shown that a better approach is to use $W\overline{v}_\delta$ as an additive correction in an iterative procedure that takes an initial estimate $V$ and generates a more accurate solution $\hat{V}$ that avoids the jaggedness of the step function approximation $W\hat{v}_\delta$. In this case the appropriate formula for $\overline{u}_\delta$ becomes:

$$\overline{u}_\delta(i) = \frac{1}{N_i} \sum_{s \in S_i} [G_\delta(V)(s) - V(s)]. \tag{3.40}$$

To see why (3.40) is apprporiate, suppose there exists an $M \times 1$ vector $y$ that solves the equation

$$V_\delta = V + Wy, \tag{3.41}$$

where $V_\delta$ is the solution to (2.35). Using the equations $G_\delta(V) = u_\delta + \beta P_\delta V$ and $V_\delta = G_\delta(V_\delta) = u_\delta + \beta P_\delta V_\delta$ we have:

$$[I - \beta P_\delta](V_\delta - V) = G_\delta(V) - V. \tag{3.42}$$

Multiplying both sides of (3.42) by $(W'W)^{-1}W'$ and substituting $Wy = (V_\delta - V)$ in the left hand side we obtain:

$$(W'W)^{-1}W'[I - \beta P_\delta]Wy = (W'W)^{-1}W'(G_\delta(V) - V). \tag{3.43}$$

Notice that since $W$ is a partition matrix, $(W'W)^{-1}$ is an $M \times M$ diagonal matrix whose $i^{th}$ diagonal element is $1/N_i$. It follows that the solution to equation (3.43) can be written as:

$$\overline{v}_\delta = [I - \beta \overline{P}_\delta]^{-1} \overline{u}_\delta, \tag{3.44}$$

where $I$ is the $M \times M$ identity matrix and $\overline{u}_\delta$ is given by (3.40). Thus, (3.44) is the appropriate form of the aggregation step when the aggregation term $W\overline{v}_\delta$ is treated as an additive correction to an intial $S \times 1$ estimate $V$ as in (3.41). An alternative formula for the disaggregation step can be found by applying $G_\delta$ to both sides of the equation $V_\delta = V + W\overline{v}_\delta$ yielding:

$$V_\delta \simeq G_\delta(V_\delta) + \beta P_\delta W \overline{v}_\delta. \tag{3.45}$$

Bertsekas and Castañon show that by interspersing a number of successive approximation steps $V_{t+1} = G_\delta(V_t)$ between each pair of aggregation/disaggregation steps (3.44) and (3.45), one can guarantee that the method will converge to the fixed point $V_\delta$. Thus, in the initial stages the additive correction term $W\overline{v}_\delta$ succeeds in shifting an initial estimate $V$ to a neighborhood of $V_\delta$, and this estimate is refined by a small number successive approximation steps. This cycle is then repeated, and as the resulting sequence $\{V_t\}$ converges to $V^*$ it is easy to see from (3.40) that the additive corrections $W\overline{v}_\delta$ in (3.41) converge to 0.

The aggregate states can be chosen on basis of a fixed partition of the state space, or can be chosen adaptively after each aggregation step $k$ in order to minimize the residual variation in $G_\delta(V_k) - V_k$. One way to do this is to divide the total variation $\Delta = max[G_\delta(V_k) - V_k] - min[G_\delta(V_k) - V_k]$ into $M$ equal intervals, assigning states with residuals $G_\delta(V_k)(s) - V_k(s)$ in the highest interval to aggregate state 1, states with residuals in the next highest interval to aggregate state 2, and so on, i.e.

$$s \in S_k \quad \text{if} \quad G_\delta(V)(s) - V(s) - \underline{b} - (k-1)\Delta \in (0, \Delta], \tag{3.46}$$

where $\underline{b} = min[G_\delta(V_k) - V_k]$.

### 3.4 Numerical Illustration of Alternative Solution Methods

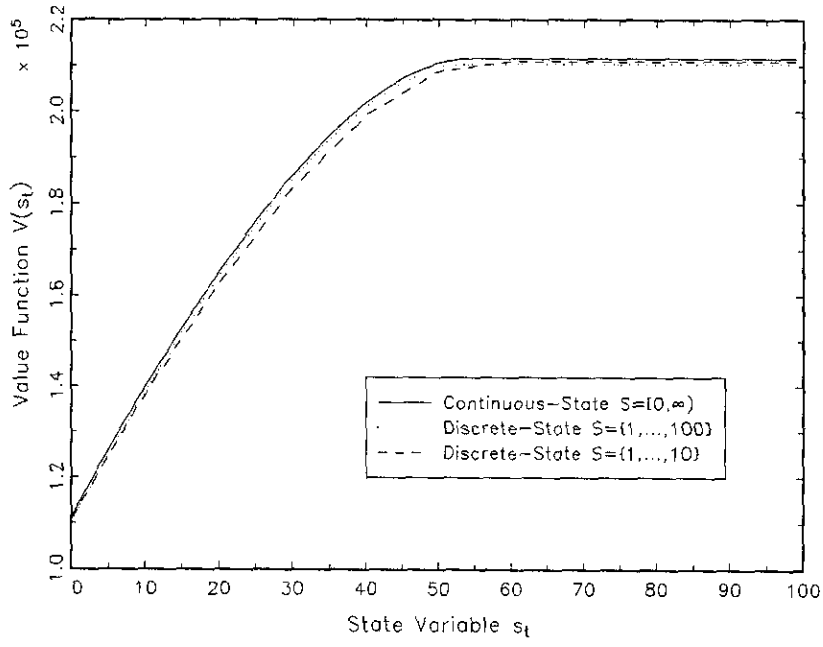


**Figure 3.1:** Comparison of Discrete vs. Continuous Value Functions in Auto Replacement Problem

As an illustration, we used each of the above methods to to solve a finite-state version of the automobile replacement problem presented in Example 3 of section 2.5. The parameters of the finite-state problem were chosen to match the analytical solution given in (2.35) in the case where $\lambda = .5$, $\beta = .95$, $c(s) = 200s$, and $\overline{P} - \underline{P} = 100,000$. Calculating the optimal stopping boundary $\gamma$ in (2.36), we see that it is optimal to replace the automobile when $s_t > \gamma = 52.87$. The corresponding value function is plotted as the solid line in figure 3.1.

An approximate discrete-state version of the problem was solved using $S = 100$ states and the same discount factor, cost function, and replacement costs. The exponential specification (2.30) for $p(\cdot \,|\, s_t, d_t)$ in the continuous case was approximated with a 12-point probability distribution in the discrete case, using a simple continuity correction. Thus, each of the 12 mass points were computed as the probability an exponential random variable falls within plus or minus .5 of the integer values $j$ that $s_t$ assumes in the discrete case:

$$p(s_{t+1} = s_t + j \,|\, s_t, d_t) = \begin{cases} \int_0^{.5} \lambda \, exp^{-\lambda y} \, dy & j = 0 \\ \int_{j-.5}^{j+.5} \lambda \, exp^{-\lambda y} \, dy & j = 1, \ldots, 10 \\ \int_{10.5}^{\infty} \lambda \, exp^{-\lambda y} \, dy & j = 11. \end{cases} \tag{3.47}$$

The discrete-state value function (computed by policy iteration) is plotted as the dotted line in figure 3.1, with one dot for each point in the state space. One can see from figure 3.1 that the discrete-state value function approximates the continuous-state version quite closely: the maximum absolute deviation between the two functions was 1164, and the maximum percentage deviation was just over 1%. Figure 3.1 also plots an interpolated value function solved with only $S = 10$ states. The maximum error in this case is 3553, representing a 1.8% deviation. One can see that even very coarse discretizations are able to do a good job of approximating a continuous underlying value function. This provides some insight into the potential effectiveness of solution methods based on state aggregation.

Tables 3.1 and 3.2 present a comparison of six alternative solution algorithms used to solve the 100 state replacement problem.[41] The tables present the number of iterations and CPU times required by each algorithm to compute an estimate of $V^*$ to within a tolerance of 1164, the maximum deviation between the continuous and discrete-state formulas for $V^*$.[42] The modified policy iteration algorithm used $N = 20$ successive approximation steps to compute an approximate fixed point of $G_\delta$. The McQueen-Porteus error bounds were employed to test for convergence

---

[41] Each of these methods were programmed in Gauss and run on an IBM 386/SX computer. The code, also written in Matlab and C, is available from the author upon request.

[42] Successive approximations were terminated when the crude estimate of the maximum deviation between $V_t$ and $V^*$ (3.28) was less than 1164. Successive approximations with error bounds were stopped when the more refined McQeen-Porteus error bounds (3.30) indicated that the maxium deviation of $V_t$ and $V^*$ was less than 1164. The various policy iteration algorithms were terminated at the first iteration $k$ such that $\delta_k = \delta_{k-1}$, although in the case of the approximate policy iteration algorithms, additional steps were continued until the McQueen-Porteus error bounds for the operator $\Gamma$ indicated that the estimated value function $V_{\delta_k}$ was within 1164 of $V^*$.

| Method | Iterations | CPU Seconds | $\mid V - V^* \mid$ | $\bar{b} - \underline{b}$ | $\mid V - \Gamma(V) \mid$ |
|---|---|---|---|---|---|
| **1.** Successive Approximations | 114 | 48.9 | 571.7 | 26.5 | 30.6 |
| **2.** Error Bounds | 65 | 29.7 | 503.4 | 1141.0 | 48.5 |
| **3.** Policy Iteration | 6 | 46.1 | $1.1E^{-9}$ | $1.1E^{-9}$ | $5.8E^{-11}$ |
| **4.** Modified Policy Iteration | 5 | 21.8 | 52.3 | 301.4 | 8.6 |
| **5.** Fixed State Aggregation | 6 | 31.4 | 20.9 | 336.0 | 8.8 |
| **6.** Adaptive State Aggregation | 8 | 171.8 | 40.4 | 291.2 | 7.7 |

**Table 3.1:** Comparison of Solution Methods $\beta = .95$

| Method | Iterations | CPU Seconds | $\mid V - V^* \mid$ | $\bar{b} - \underline{b}$ | $\mid V - \Gamma(V) \mid$ |
|---|---|---|---|---|---|
| **1.** Successive Approximations | $> 10,000$ | $> 4,600$ | $> 30,000$ | $1.1E^{-8}$ | 3.0 |
| **2.** Error Bounds | 166 | 75.7 | $4.4E^{-1}$ | 114.5 | $5.7E^{-3}$ |
| **3.** Policy Iteration | 8 | 71.0 | $2.9E^{-7}$ | $2.9E^{-7}$ | $1.5E^{-11}$ |
| **4.** Modified Policy Iteration | 11 | 50.0 | 174.9 | 219.4 | $2.9E^{-2}$ |
| **5.** Fixed State Aggregation | 10 | 51.9 | 3.4 | 93.8 | $4.7E^{-3}$ |
| **6.** Adaptive State Aggregation | 15 | 1296 | $2.4E^{-1}$ | 58.4 | $2.9E^{-3}$ |

**Table 3.2:** Comparison of Solution Methods $\beta = .9999$

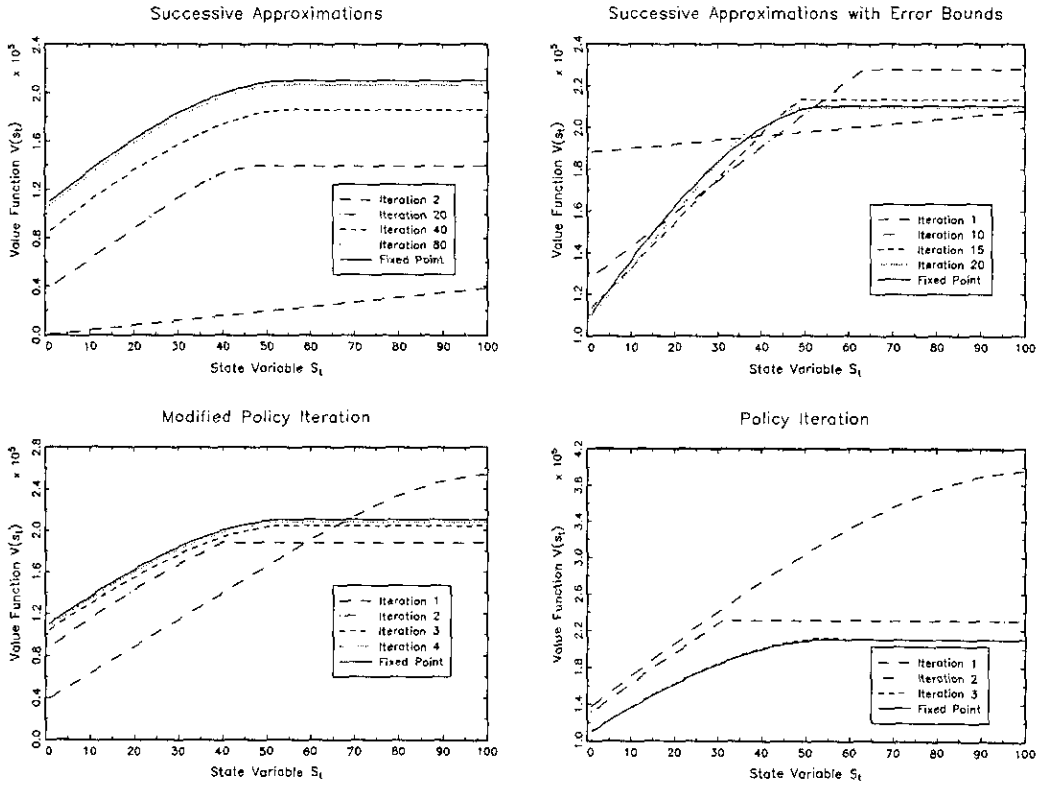


**Figure 3.2:** Convergence of $V_t$ to $V^*$ when $\beta = .95$

and produce an improved final estimate of $V_\delta$.[43] The state aggregation methods used $M = 10$ aggregate states. The fixed state method simply partitioned the states into 10 equal groups, $S_i = \{10*(i-1) + 1, \ldots, 10*i\}$. The adaptive state aggregation method partitioned the states endogenously, according to the magnitude of the residuals $G_\delta(V) - V$ as given in (3.46). In both of the aggregation methods, we set a relatively loose inner convergence tolerance: iterate $V_k$ was deemed to be a sufficiently good approximation of the fixed point $V_\delta = G_\delta(V_\delta)$ if the McQueen-Porteus error bounds $\underline{b}_k$, $\overline{b}_k$ for the operator $G_\delta$ satisfy $\overline{b}_k - \underline{b}_k < 30000(1-\beta)/\beta$. As soon as the policy converged ($\delta_k = \delta_{k-1}$), additional policy evaluation steps were carried out with a tighter convergence tolerance of $500(1-\beta)/\beta$ until the resulting estimate for $V_{\delta_k}$ was within 1164 of $V^*$ (computed via the McQueen-Porteus error bounds for $\Gamma$). Table 3.2 presents a comparison of the six methods in the case where $\beta = .9999$.

Figure 3.2 illustrates the convergence trajectories of 4 of the algorithms. Successive approximations converges to $V^*$ from below, reflecting the fact that it is solving a finite-horizon approximation to the infinite-horizon problem. Policy iteration converges to $V^*$ from above, reflecting the fact that each successive policy $\delta_t$ represents an improvement over the previous policy as established in (3.36). Convergence to $V^*$ under the error bounds and modified policy iteration procedures is not necessarily monotonic. In the former case one can see that the addition of the correction term $(\overline{b} + \underline{b})/2$ quickly translates the iterates $V_t$ into a region much closer to $V^*$ (compare the first iterate $V_1$ using the error bounds correction to the the second iterate $V_2$ under successive approximations). Once in a general neighborhood of $V^*$, successive approximations succeed in "bending" $V_t$ into the precise shape of $V^*$. Modified policy iteration is not monotonic because approximating the fixed point $V_\delta = G_\delta(V_\delta)$ by $N$ successive approximation steps implies that each iterate $V_k = G^N_{\delta_k}(V_{k-1})$ will be an underestimate of the true solution $V_{\delta_k}$. Thus, the sequence $\{V_k\}$ generated by modified policy iteration is essentially the same as under policy iteration, but translated downward. In this case the sequence $\{V_k\}$ ended up converging to $V^*$ from below. Note also the inexactness of the approximate solutions under modified policy iteration imply that more iterations are required to get close to $V^*$ in comparison to policy iteration.

Overall the results indicate that in this problem the method of policy iteration using an approximate rather than exact solution to the inner fixed point problem $V_\delta = G_\delta(V_\delta)$ are generally the fastest solution methods. In the early stages of policy iteration, it is not necessary to solve (2.35) exactly to insure rapid progress towards $V^*$ and the optimal decision rule $\delta^*$. Since the decision rules $\delta_k$ are the results of finite maximizations, $V_t$ does not have to be an extremely precise estimate of $V^*$ to insure that the corresponding policy $\delta_t$ coincides with $\delta^*$. However once $\delta_t = \delta^*$ (which is indicated by the fact that $\delta_t$ does not change in successive steps of the policy iteration) one can set finer tolerances for the approximate solution of $V_{\delta^*} = G_{\delta^*}(V_{\delta^*})$ in (2.35) to generate a more accurate estimate of $V^*$ and insure that the solution $\delta_k$ really does correspond to the optimal decision rule $\delta^*$. The Bertsekas-Castañon

[43] More precisely, one successive approximations step $V_{21} = \Gamma(V_{20})$ was performed using the operator $\Gamma$ after 20 successive approximations steps $V_{t+1} = G_{\delta_k}(V_t)$ were performed with the operator $G_{\delta_k}$. The final estimate of $V_{\delta_k}$ in modified policy iteration step $k$ is then given by: $\hat{V}_k = V_{21} + (\overline{b} + \underline{b})/2$ where $\overline{b}$ and $\underline{b}$ are the McQueen-Porteus error bounds for the operator $\Gamma$.

adapative state aggregation procedure was not effective in this particular problem. We noticed that the adaptively chosen states typically varied quite substantially over successive aggregation steps. The variations in the membership of the aggregate states frequently resulted in an approximate disaggregate solution $G_\delta(V) + \beta P_\delta W \overline{v}_\delta$ that would tend to move away from the fixed point $V_\delta$, requiring a large number of intervening successive approximations steps to move the solution back towards $V_\delta$. Use of a fixed set of aggregate states performed much more effectively in this problem, reflecting our ability to closely approximate a continuous value function using very coarse discretizations as demonstrated in figure 3.1.[44] The McQueen-Porteus error bounds are also a very cost-effective method for accelerating convergence. They can be used in conjunction with methods 4 and 5 to yield improved estimates of $V^*$ between successive policy valuation steps.

### 3.5 Algorithms for Computing Fixed Points to General Contraction Mappings

In section 4 we will encounter contraction mappings $\Psi$ which are closely related to, but distinct from the contraction mapping $\Gamma$ defined by Bellman's equation (2.10). An example of such a mapping is $\Psi: B \to B$ defined by:

$$\Psi(V)(s,d) = u(s,d) + \beta \int log\Big[\sum_{d' \in D(s')} exp\{V(s',d')\}\Big] p(ds'|s,d), \tag{3.48}$$

where $B$ is the Banach space of all measurable, bounded functions $V: S \times D \to R$ under the *essential supremum* norm, $|\cdot|_\infty$.[45] By the Contraction Mapping Theorem, a sure-fire algorithm for computing the fixed point to any contraction mapping $\Psi$ is the method of successive approximations:

$$V_k = \Psi(V_{k-1}) = \Psi^k(V_0). \tag{3.49}$$

Successive approximations for general operators $\Psi$ can be accelerated using the McQueen-Porteus error bounds (3.30), (3.31) just as in the case of the Bellman operator $\Gamma$. However a faster method is to approximate the fixed point $V^*$ using *Newton-Kantorovich* iterations:

$$V_{k+1} = V_k - [I - \Psi'(V_k)]^{-1}(I - \Psi)(V_k), \tag{3.50}$$

where $I$ denotes the identity operator on $B$, and $\Psi'(V)$ denotes the derivative of $\Psi$ evaluated at the point $V \in B$.[47] An argument exactly analogous to the series expansion argument used to proved the existence of $[I - \beta P_\delta]^{-1}$

---

[44] However in less structured problems it may not be clear how to choose the aggregate states, and in problems with multiple ergodic classes use of a fixed set of aggregate states is typically ineffective. In these problems, Bertsekas and Castañon (1989) have found that the adaptive aggregation method can outperform the modified policy iteration algorithm.

[45] The essential supremum is defined by $|V|_\infty = inf\{M: |V(s,d)| \leq M \text{ for almost all } (s,d) \in S \times D\}$.

[47] The (Gateaux) derivative is defined by $\Psi'(V_0)(V) = lim_{t \to 0}[\Psi(V_0 + tV) - \Psi(V_0)]/t$. For each $V_0 \in B$ we require that $\Psi'(V_0) \in L(B,B)$, the space of all continuous linear operators from $B$ into itself. If the above limit is uniform for all $V$ such that $|V| = 1$, $\Psi(V_0)$ is known as the Fréchet derivative (evaluated at the point $V_0$).

can be used to establish that the linear operator $[I - \Psi'(V)]^{-1}$ exists, is continuous, and has a convergent Neumann series expansion provided that $|\Psi'(V)| \in (0,1)$.[49] It is easy to see that when $\Psi$ is a contraction mapping, we have $|\Psi'(V)| \leq \beta$.[50] Thus, the iterations defined in (3.50) are always well-defined. Kantorovich's Theorem guarantees that given a starting point $V_0$ in a *domain of attraction* of the fixed point $V^*$ of $\Psi$, the Newton-Kantorovich iterations will converge to $V^*$ at a *quadratic rate:*

**Theorem 3.1 (Kantorovich):** *Suppose that $\Psi$ has continuous first and second derivatives satisfying $|\Psi''(V)| \leq K$, then for any initial point $V_0$ satisfying $|I - \Psi(V_0)| = \eta \leq (1-\beta)^2/(2K)$, the iterations (3.50) satisfy:*

$$|V_k - V^*| \leq \frac{1}{2^k}\left(\frac{2K\eta}{(1-\beta)^2}\right)^{2^k}(1-\beta)^2/K. \tag{3.51}$$

Equation (3.51) shows that provided we start the Newton-Kantorovich iterations from a point $V_0$ sufficiently close to $V^*$, convergence will be extremely rapid, since (3.51) implies that:

$$|V_{k+1} - V^*| \leq K'|V_k - V^*|^2, \tag{3.52}$$

for a constant K'. The rate of convergence can be further accelerated by using a quasi-Newton method which involves evaluating $\Psi'$ in (3.50) at the intermediate point $W_k = \alpha V_k + (1-\alpha)\Psi(V_t)$ for $\alpha \in (0,1)$. Werner (1984) showed that evaluation at the point $\alpha = 1/2$ typically improves the speed of convergence (by reducing the constant $K'$ in (3.52)) in comparison to the standard Newton-Kantorovich iteration.

Thus, Newton-Kantorovich iterations yield rapid quadratic rates of convergence but are only guaranteed to converge for initial estimates $V_0$ in a domain of attraction of $V^*$ whereas successive approximations yields much slower linear rates of convergence rates but are always guaranteed to converge to $V^*$ starting from any initial point $V_0 \in B$. This suggests the following a hybrid method or *polyalgorithm:* start with successive approximations, and when the McQueen-Porteus error bounds indicate that one is sufficiently close to $V^*$, switch to Newton-Kantorovich iterations to rapidly converge to the solution. It is interesting to note that the policy iteration algorithm is actually equivalent to the Newton-Kantorovich method (Puterman and Brumelle, 1978, 1979). However, in the case of policy-iteration the domain of attraction is the entire space $B = R^S$: no initial successive approximation iterations are necessary to guarantee convergence.

---

[49] This result is known as Banach's Theorem, see Kantorovich and Akilov, 1982, p. 154.

[50] The norm $|\Lambda|$ of a linear operator $\Lambda \in L(B,B)$ is defined by $|\Lambda| = sup\{|\Lambda(V)|: |V| = 1\}$.

## 4. References


de Acosta, A. and E. Giné (1979) "Convergence of Moments and Related Functionals in the General Central Limit Theorem in Banach Spaces" *Z. Wahrschienlichkeitstheorie verw. Gebeite* **48** 213-231.

Ames, W.F. (1977) *Numerical Methods for Partial Differential Equations* Academic Press, New York.

Amman, H.M. (1986) "Are Supercomputers Useful for Optimal Control Experiments?" *Journal of Economic Dynamics and Control* **10** 127-130.

Amman, H.M. (1990) "Implementing Stochastic Control Software on Supercomputing Machines" *Journal of Economic Dynamics and Control* **14** 265-279.

Amman, H.M. and H. Jager (1987) "Optimal Economic Policy under a Crawling Peg Exchange Rate System" In C. Carrato and D. Sartore (eds.) *Developments of Control Theory for Economic Analysis* Kluwer Academic Publishers, Dordrecht.

Amman, H.M. and H,. Jager (1988) "A Constrained Control Algorithm for Nonlinear Control Problems" *International Journal of System Science* **19** 1781-1794.

Amman, H.M. (1989) "Nonlinear control simulation on a Vector Machine" *Parallel Computing* **10** 123-127.

Amman, H.M. and D.A. Kendrick (1991) *A User's Guide for DUAL: A Program for Quadratic-linear Stochastic Control Problems* Technical Paper 90-4, Center of Economics Research, Austin Texas.

Amman, H.M. and D.A. Kendrick (1992) "Parallel Processing for Large-Scale Nonlinear Control Experiments in Economics" *The International Journal of Supercomputer Applications* **5** 90-95.

Amman, H.M. and D.A. Kendrick (1993) "Forward Looking Variables and Learning in Stochastic Control" *The International Journal of Supercomputer Applications* **7** 201-211.

Amman, H.M. and D.A. Kendrick (1993a) "Active Learning; Some Empirical Results" *Journal of Economic Dynamics and Control* **18** 119-124.

Amman, H.M. and L.M.T. van Velden (1992) "Exchange rate uncertainty in imperfect markets: A simulation approach" In H.M. Amman, D.A. Belsley, and L.F. Pau (eds.) *Computational Economics and Econometrics* Kluwer Academic Publshers, Dordrecht.

Anselone, P.M. (1971) *Collectively Compact Operator Approximation Theory and Application to Integral Equations* Prentice Hall Series on Automatic Computation, Englewood Cliffs, New Jersey.

Anselone, P.M. and R. Ansorge (1981) "A Unified Framework for the Discretization of Nonlinear Operator Equations" *Numerical Functional Analysis and Optimization* **4-1** 61-99.

Araujo, A. and Giné, E. (1980) *The Central Limit Theorem for Real and Banach Valued Random Variables* Wiley, New York.

Arnold, L. (1974) *Stochastic differential equations: Theory and Applications* John Wiley and Sons, New York.

Arrow, K.J. (1968) "Applications of Control Theory to Economic Growth" In G.B. Dantzig and A.F. Veinott (eds.) *Mathematics for Decision Sciences* Part 2 American Mathematical Society, Providence.

Arrow, K.J. Harris, T. Marschak, J. (1951) "Optimal Inventory Policy" Econometrica **19-3** 250-272.

Barron, A.R. (1992) "Universal Approximation Bounds for Superpositions of a Sigmoidal Function" forthcoming in *IEEE Transactions on Information Theory*

Basar, T. (1986) "A Tutorial on Dynamic and Differential Games" in T. Basar (ed.) *Dynamic Games and Apllications in Economics* Springer Verlag, Berlin.

Basar, T. (1988) "Solutions to a Class of Nonstandard Stochastic Control Problems with Active Learning" *IEEE Transactions on Atomatic Control* **33** 1122-1129.

Basar, T. (1989a) "Dynamic Optimization of Some Forward Looking Stochastic Models" in A. Blagviere (ed.) *Modeling and Control Systems* Lecture Notes in Control and Information Sciences **121** 315-336. Springer Verlag, Heidelberg.

Basar, T. (1989b) "Some Thoughts on Rational Expectations Models and Alternative Formulations" *Computers, Mathematics and Applications* **18** 591-604.

Becker, R., S. Hall, and B. Rustem (199?) "Robust Optimal Decisions with Stochastic Nonlinear Economic Systems" *Journal of Economics Dynamics and Control* **18** 125-148.

Bellman, R. (1957) *Dynamic Programming* Princeton University Press.

Bellman, R. Kalaba, R. and B. Kotkin (1973) "Polynomial Approximation: A New Computational Technique in Dynamic Programming: Allocation Processes" *Mathematics of Computation* **17** 155-161.

Berge, C. (1963) *Topological Spaces* Oliver & Boyd, Edinburgh.

Bergstrom, A.R. (1987) "Optimal Control in Wide-Sense Stationary Continuous-Time Stochasdtic Models" *Journal of Economic Dynamics and Control* **11** 425-443.

Berman, A. and R.J. Plemmons (1993) *Nonnegative Matrices in the Mathematical Sciences* Academic Press, New York.

Bertsekas, D. (1987) *Dynamic Programming Deterministic and Stochastic Models* Prentice Hall, New York.

Bertsekas, D. (1977) "Monotone Mappings with Application in Dynamic Programming" *SIAM Journal of Control and Optimization* **15-3** 438-464.

Bertsekas, D. (1975) "Convergence of Discretization Procedures in Dynamic Programming" *IEEE Transactions on Automatic Control* **20** 415-419.

Bertsekas, D. Castañon, D. (1989) "Adaptive Aggregation Methods for Infinite Horizon Dynamic Programming" *IEEE Transactions on Automatic Control* **34-6** 589-598.

Bertsekas, D.P. Shreve, S.E. (1978) *Stochastic Optimal Control: The Discrete Time Case* Academic Press, New York.

Bhattacharya, R.N. Majumdar, M. (1989) "Controlled Semi-Markov Models - The Discounted Case" *Journal of Statistical Planning and Inference* **21** 365-381.

Bierens, H. (1990) "A Consistent Conditional Moment Test of Functional Form" *Econometrica* **58-6** 1443-1458.

Blackwell, D. (1962) "Discrete Dynamic Programming" *Annals of Mathematical Statistics* **33** 719-726.

Blackwell, D. (1965) "Discounted Dynamic Programming" *Annals of Mathematical Statistics* **36** 226-235.

Blackwell, D. (1967) "Positive Dynamic Programming" *Proceedings of the* $5^{th}$ *Berkeley Symposium on Mathematical Statistics and Probability* **1** 415-418.

Blanchard, O.J. and C.M. Kahn (1980) "The Solution of Linear Difference Models Under Rational Expectations" *Econometrica* **48** 1305–1311.

Brown, P.N. and Y. Saad (1990) "Hybrid Krylov Methods for Nonlinear Systems of Equations" *SIAM Journal on Scientific and Statistical Computing* **11-3** 450–481.

Burden, R.L. and J.D. Faires (1989) *Numerical Analysis* PWS-KENT Publishing Company, Boston.

Brock, W.A. (1982) "Asset Prices in a Production Economy" in J.J. McCall (ed.) *The Economics of Information and Uncertainty* University of Chicago Press.

Brock, W.A. Mirman, L.J. (1972) "Optimal Economic Growth Under Uncertainty: The Discounted Case" *Journal of Economic Theory* **4** 479–513.

Buchanan, J.L. and Turner, P.R. (1992) *Numerical Methods and Analysis* McGraw-Hill, New York.

Chatelin, F. (1983) *Spectral Approximation of Linear Operators* Academic Press, New York.

Cheney, W. and D. Kincaid (1985) *Numerical Mathematics and Computers* Brooks-Cole Publishing Company, Pacific Grove, California.

Chow, C.S. and Tsitsiklis, J.N. (1989) "The Complexity of Dynamic Programming" *Journal of Complexity* **5** 466–488.

Chow, C.S. and Tsitsiklis, J.N. (1991) "An Optimal Multigrid Algorithm for Continuous State Discrete Time Stochastic Control" *IEEE Transactions on Automatic Control* **36-8** 898–914.

Chow, G.C. (1975) *Analysis and Control of Dynamic Economic Systems* John Wiley, New York.

Chow, G.C. (1979) "Optimum Control of Stochastic Differential Equation Systems" *Journal of Economic Dynamics and Control* 143–175.

Chow, G.C. (1981) *Econometric Analysis by Control Methods* John Wiley, New York.

Chow, G.C. (1992) "Dynamic Optimization without Dynamic Programming" *Economic Modelling* January, 3–9.

Chew, S.H. Epstein, L.G. (1989) "The Structure of Preferences and Attitudes Towards the Timing and Resolution of Uncertainty" *International Economic Review* **30-1** 103–118.

Chui, C.K. (1992) *An Introduction to Wavelets* Academic Press, New York.

Coleman, W. J. (1991) "Equilibrium in a Production Economy With Income Tax" *Econometrica* **59-4** 1091–1104.

Coleman, W.J. (1990) "An Algorithm to Solve Dynamic Models" manuscript, Duke University.

Coleman, T.F. and C. Van Loan (1988) *Handbook for Matrix Computation* SIAM, Philadelphia.

Cryer, C.W. (1982) *Numerical Functional Analysis* Clarendon Press, Oxford.

Daniel, J.W. (1976) "Splines and Efficiency in Dynamic Programming" *Journal of Mathematical Analysis and Applications* **54** 402–407.

Davis, P.J. Rabinowitz, P. (1975) *Methods of Numerical Integration* Academic Press, New York.

den Haan, W. and A. Marcet (1990) "Solving the Stochastic Growth Model by Parameterizing Expectations" *Journal of Business and Economic Statistics* **8-1** 31–34.

Denardo, E.V. (1967) "Contraction Mappings Underlying the Theory of Dynamic Programming" *SIAM Review* **9** 165–177.

Dudley, R.M. (1989) *Real Analysis and Probability* Wadsworth & Brooks/Cole, Pacific Groves, California.

Dynkin, E.B. Juskevic, A.A. (1979) *Controlled Markov Processes* Springer-Verlag, New York.

Epstein, L.G. Zin, S.E. (1989) "Substitution, Risk Aversion, and the Temporal Behavior of Consumption and Asset Returns: A Theoretical Framework" *Econometrica* **57-4** 937–970.

Epstein, L.G. Zin, S.E. (1990) "First-Order Risk Aversion and the Equity Premium Puzzle" *Journal of Monetary Economics* **26** 387–407.

Evtushenko, Yurij G. (1985) *Numerical Optimization Techniques* Optimization Software Division, Springer-Verlag, New York.

Fair, R.C. (1984) *Specification, Estimation and Analysis of Macroeconometric Models* Harvard University Press, Cambridge Massachusetts.

Fair, R.C. and J.B. Taylor (1983) "Solution and Maximum Likelihood Estimation of Dynamic Rational Expectations Models" *Econometrica* **51** 1169–1185.

Fleming, W.H. and H. Meter Soner (1993) *Controlled Markov Processes and Viscosity Solutions* Springer Verlag, New York.

Garey, M.R. Johnson, D.S. (1983) *Computers and Intractibility: A Guide to the Theory of NP-completeness* W.H. Freeman, New York.

Gihman, I.I. Skorohod, A.V. (1974, 1975, 1979) *The Theory of Stochastic Processes* Volumes I, II, III. Springer-Verlag, New York.

Gihman, I.I. Skorohod, A.V. (1979) *Controlled Stochastic Processes* Springer-Verlag, New York.

Hackbusch, W. (1985) *Multi-Grid Methods and Applications* Springer-Verlag, New York.

Hackbusch, W. (1994) *Iterative Solution of Large Sparse Systems of Equations* Volume 95 in Applied Mathematical Sciences Series, Springer-Verlag, New York.

Hakansson, N. (1970) "Optimal Investment and Consumption Strategies Under Risk for a Class of Utility Functions" *Econometrica* **38** 587–607.

Hammersley, J.J. and D.C. Handscomb (1992) *Monte Carlo Methods* Chapman and Hall, London.

Hansen, L.P. Sargent, T.J. (1980a) "Formulating and Estimating Dynamic Linear Rational Expectations Models" *Journal of Economic Dynamics and Control* **2-1** 7–46.

Hansen, L.P. Sargent, T.J. (1980b) "Linear Rational Expectations Models for Dynamically Interrelated Variables" in R.E. Lucas, Jr. and T.J. Sargent (eds.) *Rational Expectations and Econometric Practice* Minneapolis, University of Minnesota Press.

Hansen, L.P. Singleton, K. (1982) "Generalized Instrumental Variables Estimation of Nonlinear Rational Expectations Models" *Econometrica* **50** 1269–1281.

Hansen, L.P. Singleton, K. (1983) "Stochastic Consumption, Risk Aversion, and the Temporal Behavior of Asset Returns" *Journal of Political Economy* **91-2** 249–265.

Hawkins, D. and H.A. Simon (1949) "Note: Some conditions for Macroeconomic Stability" *Econometrica* **17** 245-248.

Hinderer, K. (1970) *Foundations of Nonstationary Dynamic Programming with Discrete Time Parameter* Springer-Verlag, Berlin.

Holbrook, R.S. (1974) "A Practical Method for Controlling a Large Nonlinear Stochastic System" *Annals of Economic and Social Measurement* **3** 155-176.

Holt, C.C. (1962) "Linear Decision Rules for Economic Stabilization and Growth" *Quarterly Journal of Economics* **76** 20-45.

Hornik, K.M. Stinchcombe, M. and H. White (1990) "Universal Approximation of An Unknown Function and Its Derivatives Using Multilayer Feedforward Networks" *Neural Networks* **3** 551-560.

Hornik, K. M. Stinchcombe, H. White, and P. Auer (1993) "Degree of Approximation Results for Feedforward Networks Approximating Unknown Mappings and Their Derivatives" manuscript, Department of Economics, University of California-San Diego.

Howard, R. (1960) *Dynamic Programming and Markov Processes* J. Wiley, New York.

Howard, R. (1971) *Dynamic Probabilistic Systems: Volume 2 — Semi-Markov and Decision Processes* J. Wiley, New York.

Imrohoroğlu, A. and S. Imrohoroğlu (1993) "A Numerical Algorithm for Solving Models with Incomplete Markets" *International Journal of Supercomputer Applications* **7-3** 212-230.

Jamshidi, M. and M. Malek-Zavarei (1986) *Linear Control Systems: A Computer Aided Approach* Pergamon Press, Oxford.

Jerrell, M. (1994) "The Use of Interval Analysis for Optimization" *Computational Economics* **7** forthcoming.

Judd, K. (1990) "Minimum Weighted Residual Methods in Dynamic Economic Models" manuscript, Hoover Institution.

Judd, K. (1991) *Numerical Methods in Economics* manuscript, Hoover Institution.

Kalaith, T. (1980) *Linear Systems* Prentice-Hall, Englewood Cliffs.

Kantorovich, L.V. Akilov, G.P. (1982) *Functional Analysis* Pergamon Press, Oxford.

Keane, M. P. Wolpin, K. I. (1993) "The Solution and Estimation of Discrete Choice Dynamic Programming Models by Simulation: Monte Carlo Evidence" in *Econometrics in Tel Aviv 1993* Tel Aviv University.

Kendrick, D.A. (1973) "Stochastic Control in Macroeconomic Models" *IEEE Conference Publication* **100** 200-207.

Kendrick, D.A. (1978) "Nonconvexities from Probing in Adaptive Control Problems" *Economics Letters* **1** 347-351.

Kendrick, D.A. (1981) "Control Theory with Applications to Economics" in K.J. Arrow and M.D. Intriligator (eds.) *Handbook of Mathematical Economics* Chapter 4 North-Holland, Amsterdam.

Kendrick, D.A. (1981) *Stochastic Control for Economic Models* McGraw-Hill, New York.

Krasnosel'skii, M.A. Vainikko, G.M. Zabreiko, P.P. Rutitskii, Ya. B. and V. Ya. Stetsenko (1972) *Approximate Solution of Operator Equations* D. Louvish, translator. Wolters-Noordhoff Publishing, Groningen.

Kreps, D.M. Porteus, E.L. (1978) "Temporal Resolution of Uncertainty and Dynamic Choice Theory" *Econometrica* **46-1** 185-200.

Kreps, D.M. Porteus, E.L. (1979) "Dynamic Choice Theory and Dynamic Programming" *Econometrica* **47-1** 91-100.

Kushner, H.J. (1990) "Numerical Methods for Stochastic Control Problems in Continuous Time" *SIAM Journal on Control and Optimization* **28-5** 999-1048.

Kushner, H.J. and P.G. Dupuis (1992) *Numerical Methods for Stochastic Control Problems in Continuous Time* Springer Verlag, New York.

Kwakernaak H. and R. Sivan (1972) *Linear Optimal Control Systems* John Wiley, New York.

Leonteif, W. (1966) *Input-Output Economics* Oxford University Press, London.

Lucas, R.E. Jr. (1978) "Asset Prices in an Exchange Economy" *Econometrica* **46** 1426-1446.

Luenberger, D.G. (1969) *Opitimization by Vector Space Methods* Wiley, New York.

Machina, M. J. (1987) "Choice Under Uncertainty: Problems Solved and Unsolved" *Journal of Economic Perspectives* **1-1** 121-154.

Malliaris, A.G. and W.A. Brock (1982) *Stochastic Methods in Economics and Finance* North-Holland, Amsterdam.

Marcet, A. Marimon, R. (1991) "Communication, Committment and Growth" manuscript, Universitat Pompeu Fabra, Barcelona, Spain.

McGrattan, E.R. (1994) "A Note on Computing Competative Equilibria in Linear Models" *Journal of Economics Dynamics and Control* **18** 149-160.

Modi, J.J. (1988) *Parallel Algorithms and Matrix Computation/* Clarendon Press, Oxford.

Nisan, Noam (1992) *Using Hard Problems to Create Pseudorandom Generators* MIT Press, Cambridge.

Ortega, J.M. (1988) *Introduction to Parallel and Vector Solution of Linear Systems* Plenum Press, New York.

Ortega, J.M. Voigt, R.G. (1985) "Solution of Partial Differential Equations on Vector and Parallel Processors" *SIAM Review* **27-2** 149-240.

Ozaki, H. and P. Streufert (1992) "Nonlinear Dynamic Programming for Nonlinear Stochastic Objectives" manuscript, University of Western Ontario.

Pan, V. Reif, J. (1985) "Efficient Parallel Solution of Linear Systems" *Transactions of the ACM* 143-152.

Papadimitriou, C.H. and J.N. Tsitsiklis (1987) "The Complexity of Markov Decision Processess" *Mathematics of Operations Research* **12-3** 441-450.

Petit, M.L. (1990) *Control Theory and Dynamic Games in Economic Policy Analysis* Cambridge University Press, Cambridge UK.

Phelps, E. (1962) "Accumulation of Risky Capital" *Econometrica* **30** 729-743.

Phelan, C. Rust, J. (1991) "U.S. Social Security Policy: A Dynamic Analysis of Incentives and Self-Selection" manuscript University of Wisconsin.

Pinkus, A. (1985) *n-Widths in Approximation Theory* Spinger-Verlag, Berlin.

Porteus, E. L. (1980) "Overview of Iterative Methods for Discounted Finite Markov and Semi-Markov Decision Chains" in R. Hartley et. al. (eds.) *Recent Developments in Markov Decision Processes*, Academic Press.

Powell, M.J.D. (1981) *Approximation Theory and Methods* Cambridge University Press.

Press, W.B.P. Flannery, S.A. Teukolsky and W.T. Vetterling (1986) *Numerical Recipies* Cambridge University Press, Cambridge.

Puterman, M.L. (1990) "Markov Decision Processes" in D.P. Heyman and M.J. Sobel (eds.) *Handbooks in Operations Research and Management Science* Volume 2, North-Holland/Elsevier, Amsterdam.

Rall, L.B. (1969) *Computational Solution of Nonlinear Operator Equations* Wiley, New York.

Rivlin, T.J. (1969) *An Introduction to the Approximation of Functions* Blaisdell Publishing Company, Waltham Massachusetts.

Rust, J. (1985) "Stationary Equilibrium in a Market for Durable Assets" *Econometrica* **53-4** 783–806.

Rust, J. (1986) "When Is It Optimal to Kill Off the Market for Used Durable Goods?" *Econometrica* **54-1** 65–86.

Rust, J. (1987) "Optimal Replacement of GMC Bus Engines: An Empirical Model of Harold Zurcher" *Econometrica* **55-5** 999–1033.

Rust, J. (1988) "Statistical Models of Discrete Choice Processes" *Transportation Research* **22B-2** 125–158.

Rust, J. (1988) "Maximum Likelihood Estimation of Discrete Control Processes" *SIAM Journal on Control and Optimization* **26-5** 1006–1023.

Rust, J. (1989) "A Dynamic Programming Model of Retirement Behavior" in D. Wise (ed.) *The Economics of Aging* University of Chicago Press, 359–398.

Rust, J. (1994) "Structural Estimation of Markov Decision Processes" forthcoming in D. McFadden and R. Engle (eds.) *Handbook of Econometrics* **volume 4** North Holland.

Rust, J. (1994) "Random Bellman Operators and the Curse of Dimensionality of Dynamic Programming Problems" manuscript, University of Wisconsin.

Rust, J. (1995) *Stochastic Decision Processes: Theory, Computation, and Estimation*, manuscript, University of Wisconsin.

Rustem, B. (1990) "Optimal Consistent Robust Feedback Rules under Parameter, Forecast and Behavioral Uncertainty" in N.M. Christadoulakis (ed.) *Dynamic Modeling and Control of National Economies 1989* 195–202 Pergamon Press, Oxford.

Saad, Y. (1984) "Practical Use of Some Krylov Subspace Methods for Solving Indefinite and Nonsymmetric Linear Systems" *SIAM Journal on Scientific and Statistical Computing* **5-1** 203–228.

Saad, Y. (1989) "Krylov Subspace Methods on Supercomputers" *SIAM Journal on Scientific and Statistical Computing* **10-6** 1200–1232.

Saad, Y. and M.H. Schultz (1986) "GMRES: A Generalized Minimum Residual Algorithm for Solving Nonsymmetric Linear Systems" *SIAM Journal on Scientific and Statistical Computing* **7-3** 856–869.

Sargent, T.J. and N. Wallace (1975) "Rational Expectations, the Optimal Monetary Instrument, and the Optimal Money Supply" *Journal of Political Economy* **83** 241–254.

Sargent, T.J. (1987) *Dynamic Macroeconomic Theory* Harvard University Press, Cambridge Massachusetts.

Sargent, T.J. (1978) "Estimation of Dynamic Labor Demand Schedules Under Rational Expectations" *Journal of Political Economy* **86-6**, 1009-1044.

Sargent, T.J. (1981) "Interpreting Economic Time Series" *Journal of Political Economy* **89-2**, 213-248.

Simon, H.A. (1956) "Dynamic Programming Under Uncertainty with a Quadratic Criterion Function" *Econometrica* **24** 74-81.

Singer, I. (1970) *Best Approximation in Normed Linear Spaces by Elements of Linear Subspaces* Springer-Verlag, Berlin.

Smith, A. A. Jr. (1991) "Solving Stochastic Dynamic Programming Problems Using Rules of Thumb" Discussion Paper 818, Institute for Economic Research, Queen's University, Ontario, Canada.

Stern, S. (1991) "Approximate Solutions to Stochastic Dynamic Programming Problems", manuscript, University of Virginia.

Stock, J. Wise, D. (1990) "Pensions, The Option Value of Work and Retirement" *Econometrica* **58-5**, 1151-1180.

Stokey, N.L. Lucas, R.E. Jr. (with Prescott, E.C.) (1989) *Recursive Methods in Economic Dynamics* Harvard University Press, Cambridge, Massachusetts.

Tauchen, G. (1990) "Solving the Stochastic Growth Model by Using Quadrature Methods and Value Function Iterations" *Journal of Business & Economic Statistics* **8-1** 49-51.

Tauchen, G. Hussey, R. (1991) "Quadrature-Based Methods for Obtaining Approximate Solutions to Nonlinear Asset Pricing Models" *Econometrica* **59-2** 371-396.

Taylor, J.B. and H. Uhlig (1990) "Solving Nonlinear Stochastic Growth Models: A Comparison of Alternative Solution Methods" *Journal of Business and Economic Statistics* **8-1** 1-17.

Temlyakov, V.N. (1987) "Approximate Recovery of Periodic Functions of Several Variables" *Math. USSR Sbornik* **56** 249-261.

Traub, J.F. (1994) "Breaking Intractability" *Scientific American* **270-1** 102-107.

Traub, J.F. Wasilkowski, G.W. and Woźniakowsky, H. (1988) *Information-based Complexity* Academic Press.

Traub, J.F. and Woźniakowsky, H. (1980) *A General Theory of Optimal Algorithms* Academic Press, New York.

Trick, M.A. and S.E. Zin (1993) "A Linear Programming Approach to Solving Stochastic Dynamic Programs" manuscript, Carnegie-Mellon University.

Vaughan, D.R. (1970) "A Nonrecursive Algebraic Solution for the Riccati Equation" *IEEE Transactions on Automatic Control* **AC-15** 597-599.

van Dijk, N.M. (1980) *Controlled Markov Processes: Time Discretization* CWI Tract 11, Mathematische Centrum, Amsterdam.

Wald, A. (1947) *Sequential Analysis* Wiley, New York.

Wershculz, A.G. (1991) *The Computational Complexity of Differential and Integral Equations* Oxford University Press, New York.

Wheeler, R.M. Jr. and K.S. Narendra (1986) "Decentralized Learning in Finite Markov Chains" *IEEE Transactions on Automatic Control* **AC-31** 519-526.

Whittle, P. (1982) *Optimization Over Time: Dynamic Programming and Stochastic Control* Volumes I and II. J. Wiley, New York.

Woźniakowski, H. (1991) "Average Case Complexity of Multivariate Integration" *Bulletin of the American Mathematical Society* **24** 185–194.

Woźniakowski, H. (1992) "Average Case Complexity of Linear Multivariate Problems" *Journal of Complexity* **8** (Parts I and II) 372–392.

Zaremba, S.K. (1968) "The Mathematical Basis of Monte Carlo and Quasi Monte Carlo Methods" *SIAM Review* **10** 303–314.

Zeidler, E. (1993) *Nonlinear Functional Analysis and its Applications* Volumes I and IIA Springer-Verlag, New York.

Zijm, W.H.M. (1980) *Nonnegative Matrices in Dynamic Programming* CWI Tract 167, Mathematische Centrum, Amsterdam.