A Comparison of Policy Iteration Methods for Solving Continuous-State, Infinite-Horizon Markovian Decision Problems Using Random, Quasi-random, and Deterministic Discretizations

John Rust
Yale University
jrust@gemini.econ.yale.edu
Preliminary Draft, April, 1997

Abstract: This paper compares the performance of the Howard (1960) policy iteration algorithm for infinite-horizon continuous-state Markovian decision processes (MDP's) using alternative random, quasi-random, and deterministic discretizations of the state space, or *grids*. Each grid corresponds to an embedded finite state Markovian decision process whose solution is used to approximate the solution to the original continuous-state Markovian decision process. Extending a result of Rust (1997), I show that policy iteration using random grids succeeds in breaking the curse of dimensionality involved in approximating the solution to a class of continuous-state discrete-action MDP's known as discrete decision processes (DDP's). I compare this "random policy iteration algorithm" (RPI) with policy iteration algorithms using deterministically chosen grids including uniform grids and "quadrature grids" both of which are subject to the "curse of dimensionality". I also compare the RPI algorithm to deterministic policy iteration algorithms based on quasi-random or "low discrepancy grids" such as the Sobol' and Tezuka sequences. While an analysis of the "worst case" computational complexity of the DDP problem shows that any deterministic solution method is subject to an inherent curse of dimensionality, my numerical comparisons reveal that in the test problems considered, policy iteration using the deterministic, low discrepancy grids were superior to the RPI algorithm. The RPI algorithm in turn, outperformed deterministic policy iteration using methods based on uniform and quadrature grids even in one and two dimensional test problems when the transition density in the MDP problem is sufficiently smooth, but can be inferior to the latter methods in problems where the transition density has large discontinuities or spikes, violating regularity conditions needed to establish the uniform convergence of the RPI algorithm. This finding suggests that policy iteration algorithms that use low discrepancy grids may succeed in breaking the curse of dimensionality in "average case" settings, since in multivariate problems the rate of convergence of these methods exceeds the rate of convergence of methods based on random grids and other deterministically chosen grids, and thus tends to outperform these methods in problems where there are no large spikes or discontinuities in the transition density of the MDP problem.[1]

[1] I am grateful to Anargyros Papageorgiou and Joseph Traub for providing the *FINDER* software that generated the Sobol and Tezuka sequences used in this paper.

1. Introduction

This paper reports the results of a numerical comparison of the speed and accuracy of alternative policy iteration algorithms for solving a class of infinite horizon continuous state dynamic programming (DP) problems associated with optimal control of *Markovian decision processes* (MDPs).[2] The (infinite horizon) *MDP problem* is to find an *optimal decision rule* α and a *value function* V that solve the optimization problem

$$V(s) \equiv \max_{\alpha} E_{\alpha}\left\{\sum_{t=0}^{\infty} \beta^t u(s_t, a_t) | s_0 = s\right\}, \tag{1.1}$$

where $s \in S$ denotes the *state* of the MDP, S is the *state space*, $\beta \in (0,1)$ is the discount factor, and E_{α} denotes the conditional expectation operator for the controlled stochastic process $\{s_t, a_t\}$ induced by the decision rule α. Except in rare and highly specialized cases, it is impossible to derive closed-form solutions to MDPs so we must generally make due with numerically computed approximate solutions to (1.1). This is especially true in MDPs with continuous state spaces (e.g. where S is a subset of a Euclidean space with non-empty interior), since for these problems V and α are functions defined on a domain with a continuum of elements and can therefore generally only be approximated.

This paper presents results of a numerical comparison of several alternative discrete approximation methods for the continuous state MDP problem. Following the literature in computer science literature, I consider algorithms that require only a finite amount of *information* about the underlying continuous MDP problem. All of the methods I compare in this paper involve computing an approximate value function and decision rule (V_N, α_N) to an embedded N-state MDP defined on a finite subset of points $\{s_1, \ldots, s_N\}$ of S which I will henceforth refer to as a *grid.* The solution (V_N, α_N) to this finite-state MDP — a pair of vectors in R^N — can be computed exactly using the method of *policy iteration* introduced by Howard (1960). Once (V_N, α_N) is computed, I show how to use a variety of interpolation or approximation methods to extend (V_N, α_N) to points $s \in S$ which are not on the grid. Any particular interpolation or approximation method yields a unique extension of the "data" (V_N, α_N) to the entire continuous state space S, so one can think of (V_N, α_N) either as a pair of vectors in R^N or as a pair of functions mapping S to R.

All of the methods for choosing grids and an associated embedded finite-state MDP considered in this paper have the property that the continuous extension of the solution (V_N, α_N) to the finite-state problem provides a uniform approximation to the true solution (V, α). Therefore I measure the accuracy of a solution by the magnitude of the approximation errors $\|\alpha_N - \alpha\|$ and $\|V_N - V\|$, where

$$\|V_N - V\| = \sup_{s \in S} |V_N(s) - V(s)|, \tag{1.2}$$

[2] There is a large theoretical literature on the general properties of and existence of solutions to MDPs: Bertsekas (1995) and Puterman (1994) are excellent introductions to this theory. Because MDPs offer a very general framework for modeling optimal decision-making over time and under uncertainty there have been numerous applications in engineering, computer science, economic theory and econometrics. See Stokey and Lucas, 1994 for a recent survey of applications of MDPs in economic theory and Rust, 1994 for a recent survey applications of MDPs in econometrics.

and $\|\alpha_N - \alpha\|$ is defined similarly. There are two components to these approximation errors: 1) the error involved in finding an approximate solution to (V_N, α_N) for the embedded N-state MDP, and 2) the approximation error involved in extending (V_N, α_N) to a function defined at points $s \in S$ that are not on the grid. In this paper I evaluate these approximation errors using two "test problems" that have analytic solutions for (V, α): a machine replacement problem and a job search problem. Both of these problems are examples of "discrete decision problems" where the decision-maker has only a finite number of possible actions to choose from. For these problems, the method of policy iteration can be used to calculate exact solutions to the embedded N-state MDP problem.[3] Furthermore, the amount of cpu time necessary to solve the embedded MDP problem depends on the number of grid points N but is approximately the same regardless of the particular grid or embedded MDP problem considered for any fixed value of N. Thus, we can compare the relative efficiencies of various methods for choosing grids and embedded MDPs by calculating the error functions $\|\alpha_N - \alpha\|$ and $\|V_N - V\|$: the method that yields the smallest error will constitute the most efficient method for the two test problems considered in this paper.

I compare the performance of five alternative grids for various values of N:

1. uniform grids
2. quadrature grids
3. Sobol' grids
4. Tezuka grids
5. random grids.

Each of these grids can be viewed as providing different ways of solving the *integration subproblem* involved in approximating the solution to an MDP: the details of how these grids are generated will be described in section 2. The integration subproblem is particularly hard in continuous-state MDP problems since the true solution V requires the calculation of potentially infinitely many integrals, one for each $s \in S$.

In section 2 I show how error bounds for $\|V - V_N\|$ depend on a bound on the maximum absolute error over a particular class of multidimensional integrals known as the *discrepancy.* For each of the grids considered I am able to derive upper bounds on the discrepancy, so the discrepancy bounds determine the rate at which $\|V - V_N\|$ tends to 0 as $N \to \infty$.[4] This analysis yields the following bounds on the approximation errors $\|V - V_N\|$ *for solving a*

[3] The solution are "exact" modulo the rounding errors inherent in finite precision arithmetic on digital computers.

[4] A slightly different argument is used to derive errors for methods based on quadrature grids and for uniform grids in problems where we have *exact integration,* i.e. problems where we can exactly integrate any spline approximation V_N of V. In this case, the approximation error then derives solely from the maximum error in approximating V by functions V_N that are piece-wise polynomials or *splines* and where the grid now determines the "knot points" used to form the spline approximation.

particular MDP problem using the various grids:

$$\|V - V_N\| = \begin{cases} O\left(\frac{1}{N^{\frac{1}{d}}}\right) & \text{for uniform grid} \\ O\left(\frac{1}{N^{\frac{1}{d}}}\right) & \text{for quadrature grid} \\ O\left(\frac{1}{\sqrt{N}}\right) & \text{for random grids} \\ O\left(\frac{\log(N)^d}{N}\right) & \text{for Sobol', Tezuka, and other "low discrepancy" grids} \end{cases} \tag{1.3}$$

One can show that the lower bounds on the rates of convergence for uniform and quadrature grids differ from the upper bounds only by a factor of proportionality, so that approaches based on uniform and quadrature grids are subject to an inherent *curse of dimensionality:* their rate of convergence is inversely proportional to the dimension d. However methods based on random and low discrepancy grids are not subject to the curse of dimensionality. It is important to note that the convergence rates quoted above are for a *fixed* MDP problem and therefore do not contradict the complexity bounds of Chow and Tsitsiklis (1989) which imply that *any* deterministic method for solving the MDP problem (including methods based on deterministic low discrepancy sequences) is subject to the curse of dimensionality. The reason is that the Chow and Tsitsiklis complexity bounds are based on a worst case analysis for a *class* of MDP problems satisfying certain Lipschitz conditions. This class is sufficiently large that one can always find a problem in the class that will perform poorly for any given deterministically chosen set of grid points. In this study I compare rates of convergence for *fixed* MDP problems and therefore the rates of convergence I observe empirically are generally much faster than the worst case lower bounds derived for a broad class of MDP problems. However I do make some attempt to study what happens in "worst case scenarios" by choosing test problems that violate standard Lipschitz regularity conditions used to establish convergence and complexity bounds in other contexts. While I show that $\|V - V_N\|$ still converges to 0 despite the existence of discontinuities in the test problems, but the curse of dimensionality appears for all of the methods when there is a sufficient level of discontinuity or "irregularity" in the problem.

For more "regular" problems the results of numerical comparisons reported in sections 3 and 4 basically confirm the theoretically predicted convergence rates quoted above. In the test problems considered, policy iteration using the deterministic, low discrepancy grids uniformly outperformed policy iteration using random grids, which I henceforth refer to as *random policy iteration* (RPI). The RPI algorithm in turn, outperformed deterministic policy iteration using uniform or quadrature grids in one and two dimensional test problems provided the transition density of the MDP problem is sufficiently smooth. However RPI can be inferior to deterministic policy iteration algorithms using uniform or quadrature grids in problems where the transition density has large discontinuities or spikes, violating regularity conditions needed to establish the uniform convergence of the RPI algorithm and policy iteration algorithms based

on low discrepancy grids. These results suggests that deterministic policy iteration using low discrepancy grids may succeed in breaking the curse of dimensiona lity in "average case" settings, since in multivariate problems the rate of convergence of these methods exceeds the rate of convergence of methods based on random discretizations and other deterministic discretization procedures, and thus tends to outperform these methods in problems where there are no large spikes or discontinuities in the transition density of the MDP problem.

Section 2 reviews some key results from the theory of MDPs, Bellman and Markov operators, describes the policy iteration algorithm, and presents the key inequalities used to derive the errors bounds presented above for the various choices of grid points. Section 3 introduces the test problems used in the numerical comparisons and describes an idea due to Kenn Judd that I used to generate a family of multidimensional test problems from unidimensional test problems with closed-form solutions. Section 4 presents results of a numerical comparison of methods in several one dimensional test problems. Section 5 presents results of a numerical comparison of methods in several one dimensional test problems. Section 6 presents some concluding remarks including conjectures and suggestions for further research.

2. Background on MDPs, Policy Iteration, and Approximation Error Bounds

This section reviews some basic facts about Markovian decision processes (MDP's), introduces a subclass of MDP's with continuous state variables and discrete decision variables known as *Discrete decision processes* (DDP's) that form the test problems in this study. Sections 2.2 and 2.3 review some key results on Bellman operators, Markov operators, and the policy iteration algorithm. Section 2.4 discusses how to approximate the solution to a continuous-state MDP problem by solving an embedded finite-state MDP problem defined over a finite grid of points in S. I derive bounds for $\|V - V_N\|$ in terms of the error between the true Bellman operator Γ and a discretized Bellman operator Γ_N. Next I derive bounds on the error $\|\Gamma(V) - \Gamma_N(V)\|$ via bounds on the error between the true Markov operators E_a and a discretized Markov operator $E_{a,N}$. Finally I derive error bounds for $\|E_a(V) - E_{a,N}(V)\|$ using error bounds for a class of multivariate integrals known as the *discrepancy*, D_N^*, so the rate of convergence of D_N^* provides on upper bound on the rate of convergence of $\|V - V_N\|$.

2.1 Definitions of MDPs and DDPs

Definition 2.1: *A (stationary infinite-horizon) Markovian Decision Process (MDP) consists of the following objects:*

- A *state space* S,
- An *action space* A,
- A family of *constraint sets* $s \to A(s) \subseteq A$,
- A *utility function* $u(s, a)$,
- A Markov *transition density* $p(s'|s, a)$,
- A *discount factor* $\beta \in (0, 1)$.

Definition 2.2: *A Discrete Decision Process (DDP) is an MDP with an action space A that contains a finite number of elements.*

In order to derive error bounds, additional regularity conditions need to be placed on (u, p): typically S is assumed to be a compact subset of R^d (with non-empty interior), p will be a density with respect to Lebesgue measure on S and $u(s, a)$ and $p(s'|s, a)$ are assumed to be a Lipschitz continuous functions of their first arguments for each $s \in S$ and each $a \in A(s)$. However in the test problems we consider in this paper the transition density p is discontinuous, and indeed in the search problem a transition *density* does not even exist. Despite the failure of these regularity conditions, I show that the discrete approximation methods can still be applied and still converge to the true solution.

2.2 Bellman's Equation and Bellman Operators

Equation (1.1) presented the optimization problem that we are trying to solve and noted that the solution consists of two objects, the optimal value function V and the optimal decision rule α. As is well known (see e.g. Bellman, 1957, Blackwell, 1965 or Denardo, 1967), the optimal decision rule α (which may not be uniquely defined) can be recovered from the value function V, which is the unique solution to *Bellman's equation* $V = \Gamma(V)$, where Γ is a contraction mapping known as the *Bellman operator:*

$$\Gamma\Big(V\Big)(s) \equiv \max_{a \in A(s)} [u(s,a) + \beta \int V(s')p(s'|s,a)ds']. \tag{2.1}$$

A standard procedure for solving infinite horizon MDPs is via *backward induction* which is equivalent to finding a fixed point of Bellman's equation by the method of successive approximations. However in some cases, especially when the discount factor β is close to 1, it can be more efficient to solve the MDP by the method of *policy iteration* which I describe in the next section.

In continuous state MDPs, Γ is a contraction mapping on the infinite-dimensional Banach space B of bounded measurable functions from S to R, and its fixed point V is also a member of B. In this paper I use *discrete approximation* to find a computable, finite-dimensional approximation to the generally non-computable infinite-dimensional fixed point problem. Consider the problem of finding a fixed point $V_N = \Gamma_N(V_N)$ to a *discretized Bellman operator* Γ_N defined by:

$$\Gamma_N\Big(V\Big)(s) \equiv \max_{a \in A(s)} \left[u(s,a) + \beta \sum_{i=1}^{N} V(s_i) p_N(s_i|s,a) \right], \tag{2.2}$$

where p_N is a discrete probability distribution over a finite grid $\{s_1, \ldots, s_N\}$ in S. It is easy to see that when the state variable s in (2.2) is restricted to the grid the fixed point equation $V_N = \Gamma_N(V_N)$ is simply Bellman's equation for an *embedded finite state MDP problem.* Since the state space for this problem has N elements it is clear that Γ_N is a contraction mapping on R^N, and its fixed point V_N will also be a vector in R^N. However in order to construct an approximation to $V \in B$ we need to extend the vector V_N to a function defined over all $s \in S$. One obvious way to do this is via *interpolation:* if $s \in S$ can be expressed as a convex combination of grid points $\{s_1, \ldots, s_N\}$, then $V_N(s)$ can be expressed as the corresponding convex combination of the components of V_N:

$$V_N(s) = \sum_{i=1}^{N} \mu_i V_N(s_i) \quad \text{where} \quad s = \sum_{i=1}^{N} \mu_i s_i \quad \text{and} \quad \sum_{i=1}^{N} \mu_i = 1,\ \mu_i \geq 0, \tag{2.3}$$

where $V_N(s_i)$ denotes the i^{th} component of the vector V_N. Although I include interpolation as one of the methods evaluated in this paper, I note several of its shortcomings here: 1) the interpolated solution may not be uniquely defined: uniqueness is typically imposed by requiring s to be a convex combination of its "adjacent" or "nearest neighbor" grid points, but for arbitrary grids it may not be easy to find the nearest neighbors of an arbitrary $s \in S$,

2) some $s \in S$ may be "outside the grid" and are therefore not expressable as a convex combination of grid points, so auxiliary extrapolation methods must be used to determine $V_N(s)$ for such points. One could also use a variety of other "smoothing" or approximation methods, treating the vector $\big(V_N(s_1), \ldots, V_N(s_N)\big)$ as N "data points" and using parametric or nonparametric regression methods to find a function V_N defined over all of S that best fits the data. Unfortunately any approach which requires us to solve an auxiliary interpolation/approximation problem wil be subject to an inherent curse of dimensionality, at least on a worst case basis. Furthermore, it has been proven that use of randomization cannot break the curse of dimensionality of this approximation problem (see, e.g. Traub, Wasilkowski and Woźniakowski, 1988, or Novak 1988).

A better approach to extending V_N to a function defined over all $s \in S$ is to use a method that avoids the need to solve an auxiliary interpolation/approximation problem. A standard approach is *Nyström's method*, a technique well known in the literature on numerical solution of Fredholm integral equations (see, e.g. Press, *et. al.* 1992). In the present context Nyström's method amounts to the observation that if $u(s,a)$ and $p_N(s_i|s,a)$ are uniformly bounded and measurable functions defined for all $s \in S$, then Γ_N given in (2.2) has a dual interpretation: it can be viewed as a contraction mapping on R^N when the state space is restricted to the grid $\{s_1, \ldots, s_N\}$, but it can also be viewed as a contraction mapping on the infinite-dimensional Banach space B when $\Gamma_N(V)$ is regarded as a function defined for all $s \in S$. It then follows that the unique fixed point V_N to Γ_N also has a dual interpretation as an element of R^N (when we restrict attention to its values on the grid $\{s_1, \ldots, s_N\}$) or as an element of B (when we consider V as a function defined for all $s \in S$). Undert the latter interpretation we can use the triangle inequality, the contraction-mapping property, and the fact that Γ and Γ_N have common modulus β, to derive the following error bound for $\|V - V_N\|$:

$$\|V - V_N\| \leq \frac{\|\Gamma(V) - \Gamma_N(V)\|}{(1-\beta)}. \tag{2.4}$$

Thus, the problem of uniform approximation of V is reduced to the problem of uniform approximation of the Bellman operator. Rust (1997) described the property that $\Gamma_N(V)$ and V_N have a dual interpretation as elements of R^N or elements of B as the property of *self-approximation.* Self-approximation implies that it is not necessary to employ auxiliary function approximation or smoothing algorithms to extend the solution to the embedded finite state MDP, $\big(V_N(s_1), \ldots, V_N(s_N)\big)$, to a function V_N defined over all of S. He showed that the self-approximation property is the key to a proof that randomization succeeds in breaking the curse of dimensionality of the DDP problem. The strategy of his proof is to show that approximation error bounds for the nonlinear operator Γ can be further decomposed into error bounds for a finite number of linear operators E_a known as *Markov operators.* The approximation error bounds for these infinite-dimensional operators can the be reduced to well-known error bounds for the maximum error in a particular class of multivariate integrals known as the *discrepancy.* Rust (1997) used empirical process methods and *maximal inequalities* to bound the expected value of the discrepancy. However this approach is not applicable in cases where the grid $\{s_1, \ldots, s_N\}$ is deterministically rather than randomly chosen. Indeed one might question the

relevance of Rust's results on the grounds that it is not possible to generate truly random sequences on any digital computer. Rust (1997) argued that this objection is of more ph ilisophical than practical relevance since it deterministic random number generators faithfully mimic the key properties of truly random sequences, the most important of which is the property that the sequences are *uniformly distributed.*[5] In this paper I address the objection head-on by deriving error bounds for $\|V - V_N\|$ that explicitly account for the fact that we are using deterministic rather than random grids. I do this by employing a deterministic error bound known as the *Koksma-Hlawka inequality* rather than a maximal inequality.

2.3 Markov Operators and Policy Iteration

As I noted in the introduction, there are three subproblems that must be confronted in order to solve infinite horizon, continuous state MDP problems: 1) a fixed point subproblem, 2) a maximization subproblem, and 3) an integration subproblem. Since this paper focuses on DDPs, I can solve maximization subproblem exactly. Furthermore, in this section I show that by using policy iteration I can solve the fixed point problem exactly, computing the exact fixed point $V_N = \Gamma_N(V_N)$ of the discretized Bellman equation (a vector in R^N). Thus, the only subproblem the remains to be solved is the integration subproblem, i.e. the problem of approximating the conditional expectations operator (integral) entering the Bellman operator Γ. In equation (1.1) the integration subproblem is represented by E_α, which denotes the conditional expectation operator for the controlled stochastic process $\{s_t, a_t\}$ induced by the stationary decision rule $a_t = \alpha(s_t)$ and the transition density $p(s_{t+1}|s_t, a_t)$. However in the remainder of this paper I will use E_α to denote the one-step-ahead or *Markov operator,* i.e. the linear operator defined by

$$E_\alpha V(s) = \int V(s')p(s'|s, \alpha(s))ds'. \tag{2.5}$$

When the state space S contains infinitely many elements, E_α is an operator on the infinite-dimensional Banach space B of bounded, measurable functions from S into R. It follows that evaluation of $E_\alpha V$ involves computation of infinitely many integrals, one for each $s \in S$. We can write Bellman's equation using this Markov operator notation as:

$$V(s) = \max_{a \in A(s)} [u(s, a) + \beta E_a V(s)] = u(s, \alpha(s)) + \beta E_\alpha V(s), \tag{2.6}$$

where $\alpha(s)$ is the optimal decision in state s. The value function V can be recovered from α via *policy evaluation,* i.e.

$$V = u_\alpha + \beta E_\alpha V = (I - \beta E_\alpha)^{-1} u_\alpha, \tag{2.7}$$

where u_α is the function defined by $u_\alpha(s) = u(s, \alpha(s))$, i.e. the reward or utility function implied by the optimal decision rule α. Equation (2.7) shows that V is the unique solution to an infinite-dimensional linear operator equation,

[5] See also Traub and Woźniakowski (1992) who provide sufficient conditions for sequences from deterministic linear congruential generators to behave equivalently to truly random sequences.

a *Fredholm integral equation of the second kind.* Existence and uniqueness of a solution follow from the Fredholm theory and the fact that E_α is a Markov operator and $\beta \in (0,1)$: it is straigthforward to show that the inverse operator $(I - \beta E_\alpha)^{-1}$ exists and has the following infinite series or *Neumann representation:*

$$(I - \beta E_\alpha)^{-1} = \sum_{t=0}^{\infty} [\beta E_\alpha]^t. \tag{2.8}$$

It follows that the methods in this paper are also applicable to approximating solutions to Fredholm integral equations, since each policy evaluation step of the policy iteration algorithm described below involves solution of a Fredholm integral equation of the second kind.[6]

The duality between V and α is the basis for the *policy iteration algorithm* which consists of iterations consisting of alternating policy evaluation and policy improvement steps. Suppose that at iteration k of this algorithm we have a candidate decision rule α_k. Then we perfrom a policy evaluation step, solving equation (2.7) to find the value function V_{α_k} corresponding to policy, α_k. Then at iteration $k+1$ we form a new policy α_{k+1} via a *policy improvement step:*

$$\alpha_{k+1}(s) = \underset{a \in A(s)}{argmax}[u(s,a) + \beta E_a V_{\alpha_k}(s)]. \tag{2.9}$$

It is easy to see that if $\alpha_{k+1} = \alpha_k$, then $V_{\alpha_{k+1}} = V_{\alpha_k} = V$, where V is the unique solution to Bellman's equation (2.6). It follows that if policy iteration converges, it converges to the solution of the MDP problem. One can show (see, e.g. Puterman and Brumelle, 1979) that policy iteration is equivalent to Newton's method for finding V as a zero of Bellman's equation, $0 = V - \Gamma(V)$. Furthermore, policy iteration yields a monotonic sequence of candidate value functions, i.e. $V_{\alpha_k} \leq V_{\alpha_{k+1}}$, $k = 0, 1, \ldots$. In finite state DDPs there are only a finite number of possible policies, the monotonicity property guarantees that policy iteration converges to the optimal solution in a finite number of iterations.

2.4 Discrete Markov Operators and Embedded MDPs

As noted above, all of the discrete approximation methods analyzed in this paper, including iterative methods based on discretization of the Bellman operator, can be viewed in a unified framework as arising from solutions to an embedded finite state MDP determined by the grid $\{s_1, \ldots, s_N\}$ in S. Given any policy α, for the embedded N-state MDP, there is a corresponding *discrete Markov operator* $E_{\alpha,N}$ given by:

$$E_{\alpha,N} V(s) = \sum_{i=1}^{N} V(s_i) p_N(s_i | s, \alpha(s)) \tag{2.10}$$

---

[6] Strictly speaking, the methods are applicable only to a subset of Fredhom integral equations where the Fredholm operator K has the representation $K = \beta E$ where E is a Markov operator and β is a constant between 0 and 1. In a subsequent paper I will address the issue of whether the methods in this paper can be extended to a wider class of Fredholm integral equations.

where $\{s_1, \ldots, s_N\}$ is some predetermined grid in S and p_N is a discrete probability density over points on this grid. If α and p_N are defined for all points $s \in S$ which may not necessarily lie on the grid, then $E_{\alpha,N}$ also has a dual interpretation as a Markov operator on either the infinite-dimensional space B or on the finite-dimensional Euclidean space R^N. In the latter case $E_{\alpha,N}$ can be represented by an $N \times N$ transition probability matrix given by:

$$E_{\alpha,N}[i,j] = \left[p_N(s_i|s_j, \alpha(s_j))\right]. \tag{2.11}$$

Just as in the continuous state case, the discretized Bellman operator Γ_N can be represented as the maximum of a finite number of Markov operators:

$$\Gamma_N(V) = \max_{a \in A(s)} \left[u(s,a) + \beta E_{\alpha,N} V(s)\right]. \tag{2.12}$$

The solution (V_N, α_N) to the embedded finite state MDP can be computed from the fixed point to the discretized Bellman equation, $V_N = \Gamma_N(V_N)$, and this fixed point can be computed exactly in a finite number of iterations by the policy iteration algorithm. In direct analogy to the infinite dimensional case the policy evaluation step is given by:

$$V_{\alpha_k} = u_{\alpha_k} + \beta E_{\alpha_k,N} V = \left(I - \beta E_{\alpha_k,N}\right)^{-1} u_{\alpha_k} \tag{2.13}$$

where V_{α_k} is an $N \times 1$ vector whose i^{th} element is $V_{\alpha_k}(s_i)$, the expected discounted utility of policy α_k in state s_i, and u_{α_k} is $N \times 1$ vector whose i^{th} element is given by $u_\alpha(s_i) = u(s_i, \alpha(s_i))$, the utility received in state s_i under policy α_k. Also in direct analogy to the infinite-dimensional case, the policy improvement step is given by:

$$\alpha_{k+1}(s_i) = \underset{a \in A(s_i)}{argmax} \left[u(s_i,a) + \beta E_a V_{\alpha_k}(s_i)\right], \quad i = 1, \ldots, N. \tag{2.14}$$

As noted above, the policy iteration algorithm is guaranteed to converge to the optimal policy in a finite number of iterations. The dominant work in policy iteration are the $O(N^3)$ operations necessary to carry out the policy evaluation step, (2.13). As a result, policy iteration tends to become impractical when N exceeds several thousand unless one has access to supercomputers.[7]

---

[7] Supercomputers can solve dense systems of $N = 10000$ equations in a matter of seconds. However larger systems can be solved using approximate methods, such the "GMRES" method described in Rust, 1996.

2.5 Deriving Error Bounds for MDPs from Error Bounds for Markov Operators

Using Lemma 3.1 of Rust (1997) we can derive an error bound for $\|V - V_N\|$ in terms of the errors $\|E_a - E_{a,N}\|$:

$$\|V - V_N\| \leq \frac{max_{a \in A} \|E_a(V) - E_{a,N}(V)\|}{(1-\beta)}. \tag{2.15}$$

So to complete our analysis, we need to derive bounds on $\|E_a(V) - E_{a,N}(V)\|$. To do this, we need to be more specific about how $E_{a,N}$ is formed by specifying the discrete Markov transition probability p_N. One strategy for forming p_N is to make it equal to a normalized ratio of the underlying transition density $p(s'|s,a)$:

$$p_N(s_k|s,a) = \frac{p(s_k|s,a)}{\sum_{i=1}^{N} p(s_i|s,a)}, \tag{2.16}$$

assuming the denominator in (2.16) is not equal to 0. A sufficient condition for this is that $p(s'|s,a)$ has full support S for each $a \in A$ and $s \in S$. Notice that this normalization ensures that p_N is a discrete probability density, and if it is defined for all $s \in \{s_1, \ldots, s_N\}$, then $\{u, p_N, \beta\}$ is a well-defined N-state MDP. I used (2.16) to construct p_N and $E_{\alpha,N}$ for the random, Sobol', and Tezuka grids.

A slightly different construction was used for quadrature grids following the approach of Tauchen and Hussey (1981). Their approach is based on Gauss-Legendre quadrature which is used to approximate an integral of a function on the unit interval as:

$$\int_0^1 f(s)ds \simeq \sum_{i=1}^{N} w_i f(s_i), \tag{2.17}$$

where the $\{w_i\}$ are the *quadrature weights* and the $\{s_i\}$ are *quadrature abscissae*. These $2N$ unknowns are chosen so that the approximate integral of the right hand side of (2.17) is exact for all polynomials of degree $2N-1$ or less, and there are standard routines for calculating the (w_i, s_i) recursively (see e.g. Press, *et. al.* 1992). Given the N quadrature weights and N quadrature abscissa, we can define a discrete transition probability p_N by the formula

$$p_N(s_k|s,a) = \frac{w_k p(s_k|s,a)}{\sum_{i=1}^{N} w_i p(s_i|s,a)}. \tag{2.18}$$

Since the quadrature weights are required to integrate constant functions exactly, they must sum to 1: one can also show that they must also be nonnegative. It follows that (2.18) also defines a valid probability density, and if $p_N(s_k|s,a)$ is defined for all $a \in A$ and s in the N quadrature abscissa then $\{u, p_N, \beta\}$ is also a well-defined N-state MDP. In multidimensional problems a *product rule* can be employed: the quadrature grid is formed as a cartesian product of unidimensional quadrature grids and the weights for each of these grid points are simply a product of the corresponding weights for the unidimensional grids. Otherwise the corresponding discrete Markov transition probability p_N has the same form as equation (2.18). It is easy to see that multidimensional quadrature by product rules are subject to the curse of dimensionality: if there are N points used in each of d dimensions, the quadrature grid for S consists of N^d points.

I consider one final approach to constructing p_N, *exact integration*, which can be used in cases where the grid defines a *triangulation* of the state space S (i.e. line segments connecting the grid points define a partition of S into a finite number of non-intersecting polyhedra). For example in the case where $S = [0,1]$ a uniform discretization with N points divides S into N equal intervals of length $1/N$ with midpoints $s_i = \frac{1}{2N} + \frac{(i-1)}{N}$, $i = 1, \ldots, N$. The corresponding p_N is given by:

$$p_N(s_i|s,a) = \int_{\frac{(i-1)}{N}}^{\frac{i}{N}} p(s'|s,a)ds' \quad s \in (\frac{(i-1)}{N}, \frac{i}{N}], \quad i = 1, \ldots, N. \tag{2.19}$$

Note that in many problems that it is not possible to compute exact integrals as in equation (2.19), but in the two test problems studied in this paper $p(s'|s,a)$ is an exponential and gaussian distribution, respectively, so it is possible to perform virtually "exact" integrations as in (2.19). In fact, using integration by parts it is possible to find closed-form expressions for $E_a V(s)$ provided that V is any piecewise polynomial function with fixed coefficients for each partition element. Any spline approximation to V with knot points at the N grid points is an example of a piecewise polynomial function. Although the corresponding $E_{a,N}$ operator no longer has the simple representation as in (2.10) and we lose the interpretation of the discretized MDP as an embedded finite state MDP (so the direct application of finite state policy iteration is no longer applicable), we can still derive error bounds for $\|V - V_N\|$ and use successive approximations to compute an approximate fixed point $V_N = \Gamma_N(V_N)$: see Santos and Vigo (1996) for examples of this approach. In this paper I restrict attention to discretizations that yield N-state MDPs and whose discrete Markov operators have the representation in (2.10) so I can use policy iteration. However in section 2.6 I discuss error bounds for exact integration approaches using "higher order" spline approximations such as considered in Santos and Vigo (1996).

2.6 Stochastic Error Bounds Via Maximal Inequalities

Now that I have explicitly defined the $E_{a,N}$ operators, I can derive bounds on the approximation error $\|E_a - E_{a,N}\|$, yielding error bounds for the object of interest, $\|V - V_N\|$, via inequality (2.15). Consider first discrete Markov operators $E_{a,N}$ derived from p_N of the form given in (2.18). For simplicity, I will assume in what follows that $S = [0,1]^d$, i.e. S is the d-dimensional hypercube. As long as S is compact it will be contained in some cube and via a change of variables we can map this cube back to the case $S = [0,1]^d$. Thus, the only thing that will change in the error bounds will be that the proportionality constants will be increased accordingly. If the grid $\{s_1, \ldots, s_N\}$ is "truly random" (i.e. *IID* draws from $lambda(s)$, the Lebesgue or uniform measure on $[0,1]^d$), the analysis of Rust (1997) applies. Using a maximal inequality for empirical processes due to Pollard (1989), Rust's Theorem 3.3 provides the following bound on the expected error in the random variable $\|E_a(V) - \tilde{E}_{a,N}\|$ that holds for each $N \geq 1$:

$$E\left\{\|E_a(V) - \tilde{E}_{a,N}\|\right\} \leq \sqrt{\frac{\pi}{2}} \frac{[1 + d\sqrt{\pi}C]K_p\|V\|}{\sqrt{N}}, \tag{2.20}$$

where K_p is a Lipschitz bound on $p(s'|s,a)$, and C is an absolute constant independent of N, V, and p. Note that this bound, together with error bound (2.15) implies that the expected approximation error in the random policy iteration algorithm, $E\{\|V - V_N\|\}$, converges to zero at rate $1/\sqrt{N}$ *independent of the dimension* d *of the state space.* This immediately implies that randomization breaks the curse of dimensionality of the DDP problem.

As noted in the introduction, the relevance of this result has been questioned on the grounds that real digital computers are incapable of generating "truly random" sequences: instead computers generate deterministic or "pseudo random" sequences that are intended to mimic many key aspects of randomness. In order to deal with this case I now derive deterministic error bounds for $\|V - V_N\|$ and verify that *for fixed MDP problems* that pseudo random sequences also succeed in breaking the curse of dimensionality. This is true even though lower bounds on the minimax or worst case error for any reasonably broad class of functions (e.g. all uniformly bounded Lipschitz continuous functions (u,p)) will necessarily suffer from the curse of dimensionality (see Chow and Tsitsiklis, 1989 and Novak, 1988). The intuitive explanation for the seemingly paradoxical result is the same as for multivariate integration: given any finite grid and any sufficiently broad class of integrands (e.g. all Lipschitz continuous functions) if the grid contains fewer than N^d points we can show that there are 'worst case" integrands for which the approximation error is greater than $1/N$. In section 3 we will see that this sort of "worst case" scenario also appears for fixed MDP problems when the transition probablity density of the MDP has discontinuities.[8]

2.7 Deterministic Error Bounds Via the Koksma-Hlawka Inequality

The key result for deriving bounds for $\|E_a(V) - E_{a,N}V\|$ in the deterministic case is the *Koksma-Hlawka inequality:*

$$\left| \frac{1}{N} \sum_{i=1}^{N} f(s_i) - \int f(s)\lambda(ds) \right| \leq \mathcal{V}(f) D_N^*(s_1, \ldots, s_N), \tag{2.21}$$

where λ is Lebesgue measure on $[0,1]^d$, $\mathcal{V}(f)$ is the *variation* of f (see p. 67 of Bouleau and Lépingle 1994 for a definition), and D_N^* is the *discrepancy:*

$$D_N^*(s_1, \ldots, s_N) \equiv \sup_{B \in \mathcal{B}} |\lambda_N(B) - \lambda(B)|, \tag{2.22}$$

where $\mathcal{B}$ is the class of (open) suborthants of $[0,1]^d$, ($\mathcal{B} = \{[0,s)^d \subset [0,1]^d | s \in [0,1]^d\}$) and λ_N is the empirical CDF corresponding to the sample points $(s_1, \ldots, s_N)$. Any sequence $\{s_N\}$ that satisfies $D_N^*(\{s_N\}) \to 0$ as $N \to \infty$ is said to be *uniformly distributed.* The Koksma-Hlawka inequality implies that if the sequence $\{s_N\}$ is uniformly distributed, the $\lambda_N \underset{d}{\Longrightarrow} \lambda$, i.e. the empirical CDF for $\{s_N\}$ converges in distribution to Lebesgue measure on $[0,1]^d$.

[8] Note that the deterministic worst case complexity of integration with discontinuous integrands is infinity: i.e. there is no method guaranteed to find an ϵ-approximation to an abitrary function if that function is allowed to be discontinuous. See TWW (1988).

It follows immediately that if f is any bounded function which is continuous except on a set of Lebesgue measure 0 in $[0,1]^d$ we have:

$$\lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} f(s_i) = \int f(s)ds. \tag{2.23}$$

For smoother functions, say if $f : [0,1]^d \to R$ is in C^d (i.e. is d-times continuously differentiable), then one can derive a more explicit formula for the variation of f:

$$\mathcal{V}(f) = \left\| \frac{\partial^d f}{\partial s_1 \cdots \partial s_d} \right\|_1 = \int \cdots \int \left| \frac{\partial^d f(s_1, \ldots, s_d)}{\partial s_1 \cdots \partial s_d} \right| ds_1 \cdots ds_d. \tag{2.24}$$

While formulas such as (2.24) are conceptually useful, in practice it is quite difficult to evaluate $\mathcal{V}(f)$ for use in determining how large to set N to guarantee that a given error tolerance is attained.

Using the Koksma-Hlawka inequality it is now straightforward to derive the follow bound for $\|E_a V - E_{a,N} V\|$:

Lemma 2.1: *Let $S = [0,1]^d$ and let E_a be the Markov operator defined in equation (2.5) and $E_{a,N}$ be the discrete Markov operator defined in equation (2.10), where the density p_N in (2.10) is given in equation (2.18). Then we have:*

$$\|E_a V - E_{a,N}\| \leq \left(K_1(V,p,a) + K_2(p,a)\|V\|\right) D_N^*(s_1, \ldots, s_N), \tag{2.25}$$

where:

$$K_1(V,p,a) = \sup_{s\in[0,1]^d} \mathcal{V}\left(V(\cdot)p(\cdot|s,a)\right), \tag{2.26}$$

and

$$K_2(p,a) = \sup_{s\in[0,1]^d} \mathcal{V}\left(p(\cdot|s,a)\right). \tag{2.27}$$

Proof: Define a linear operator $\hat{E}_{a,N}$ by

$$\hat{E}_{a,N}(V)(s) = \frac{1}{N}\sum_{i=1}^{N} V(s_i)p(s_i|s,a). \tag{2.28}$$

Then it follows that

$$\hat{E}_{a,N}(V)(s) - E_{a,N}(V)(s) = \frac{1}{\frac{1}{N}\sum_{i=1}^{N} p(s_i|s,a)} \left[1 - \frac{1}{N}\sum_{i=1}^{N} p(s_i|s,a)\right] \left[\frac{1}{N}\sum_{i=1}^{N} V(s_i)p(s_i|s,a)\right]. \tag{2.29}$$

Applying the Koksma-Hlwaka inequality we have:

$$\left|1 - \frac{1}{N}\sum_{i=1}^{N} p(s_i|s,a)\right| = \left|\int p(s'|s,a)ds' - \frac{1}{N}\sum_{i-1}^{N} p(s_i|s,a)\right| \leq \mathcal{V}\left(p(\cdot|s,a)\right) D_N^*(s_1, \ldots, s_N). \tag{2.30}$$

Using equation (2.29) and inequality (2.30) and taking suprema over $s \in [0,1]^d$ we have:

$$\|\hat{E}_{a,N}(V) - E_{a,N}(V)\| \leq K_2(p,a)\|V\| D_N^*(s_1, \ldots, s_N). \tag{2.31}$$

A similar argument yields

$$\|E_a(V) - E_{a,N}(V)\| \leq K_1(V,p,a) D_N^*(s_1, \ldots, s_N). \tag{2.32}$$

Finally, applying the triangle inequality we have:

$$\|E_a(V) - E_{a,N}\| = \|E_a(V) - \hat{E}_{a,N} + \hat{E}_{a,N} - E_{a,N}\| \leq \|E_a(V) - \hat{E}_{a,N}\| + \|\hat{E}_{a,N} - E_{a,N}\|. \tag{2.33}$$

Substituting inequalities (2.31) and (2.32) into the triangle inequality (2.33) yields the result.

Combining inequalities (2.15) and (2.25) we see that the discrepancy $D_N^*(s_1, \ldots, s_N)$ of the grid $\{s_1, \ldots, s_N\}$ provides an upper bound on the approximation error $\|V - V_N\|$.

Theorem 2.1: *Under the conditions of Lemma 2.1, it follows that the approximate value function V_N will be a consistent estimator of V (i.e. $\|V - V_N\| \to 0$ as $N \to \infty$), if the grid $\{s_1, \ldots, s_N\}$ is uniformly distributed. Furthermore we have:*

$$\|V - V_N\| = O\left(D_N^*(\{s_N\})\right). \tag{2.34}$$

Theorem 2.1 immediately implies that for given any deterministic, pseudo-random grid we have $\|V - V_N\| = O(1/\sqrt{N})$ since pseudo-random grids are designed to mimic the key properties of truly random grids (which are just *IID* draws in $[0,1]^d$), and one of these key properties is their rate of convergence, which is $O_p(1/\sqrt{N})$. However there are other deterministic sequences which are uniformly distributed and known to converge at rates faster than $1/\sqrt{N}$:

Definition 2.3: ***A Low Discrepancy Grid*** *$\{s_1, \ldots, s_N\}$ is any sequence of points in $[0,1]^d$ whose discrepancy satisfies:*

$$D_N^*(s_1, \ldots, s_N) = O\left(\frac{(\log N)^d}{N}\right). \tag{2.35}$$

A number of different sequences have discrepancies that satisfy the bound in (2.35), including the *Hammersley, Halton, Lapeyre-Pagés, Faure,* and *Sobol'* sequences. Many of these sequences have deep connections to number theory and it is beyond the scope of this paper to go into details about how these sequences are generated and why they satisfy the bound (2.35). I refer the reader to Bouleau and Lépingle (1994), Niederreiter (1992) and Tezuka (1995) for further details on this theory. Tezuka's book defines a class of sequences he terms *Generalized Neiderreiter sequences* and *Generalized Faure sequences.* Computer programs for generating a variety of low discrepancy sequences have been developed by Papageorgiou and Traub (1996), who have been very generous in allowing me to use their FINDER

software to generate the Sobol' and Generalized Faure sequences used in this paper. Following their terminology I refer to the latter sequences as *Tezuka sequences* in honor of Tezuka's work.[9]

Note that when $d = 1$ it is easy to show that a uniform grid, $s_i = \frac{1}{2N} + \frac{(i-1)}{N}$, $i = 1, \ldots, N$, has the smallest possible discrepancy and $D_N^*(s_1, \ldots, s_N) = \frac{1}{2N}$ in this case. However in higher dimensions, $d > 1$, uniform grids do not have the low discrepancy property. Indeed, it is easy to show that the discrepancy of a uniform grid with N points in d dimensions satisfies:

$$D_N^*(s_1, \ldots, s_N) \geq \frac{1}{2N^{\frac{1}{d}}}, \tag{2.36}$$

so uniform grids are obviously subject to a curse of dimensionality.

2.8 Deterministic Error Bounds for Uniform and Quadrature Grids via Polynomial Approximations

We need to use a slightly different approach to derive error bounds for $\|V - V_N\|$ when using uniform grids or any grid defining a triangulation of S and we assume we can do *exact integration* over the elements of the triangulation. Let V_N be a step function adapted to this triangulation: i.e. V_N is piecewise constant within each element of the triangulation of S. Then we can show that the approximate Markov operator $E_{a,N}$ can be represented by

$$E_{a,N}(V) = \int \hat{V}_N(s')p(s'|s,a)ds', \tag{2.37}$$

where $\hat{V}_N$ is a step function satisfying $\hat{V}_N(s) = V(s_i)$ whenever s is an element of the i^{th} cube of the triangulation of S induced by the uniform grid. Using equation (2.37) we have

$$\|E_a(V) - E_{a,N}(V)\| \leq \int \|V - \hat{V}_N\| p(s'|s,a)ds' \leq \|V - \hat{V}_N\|, \tag{2.38}$$

so the rate of convergence of $E_{a,N}$ to E_a is determined by the rate of convergence of a step function approximation (a 0-spline) to V. If V is continuously differentiable or Lipschitz continuous, it is easy to show that $\|V - \hat{V}_N\|$ is bounded by Lipschitz constant for V times the maximum diameter of the partition elements induced by the triangulation of S. For a uniform grid we have

$$\|V - \hat{V}_N\| \leq \frac{K\sqrt{d}}{N^{\frac{1}{d}}}. \tag{2.39}$$

It follows that the approximation error $\|V - V_N\|$ (where V_N is the step function approximation to V computed from policy iteration on the embedded N-state MDP) satisfies the following bound:

---

[9] Software for generating Sobol's sequences can also be found in Press *et. al.* 1992.

Theorem 2.2: *Let V_N be a 0-spline or step function approximation to V computed from an embedded N-state MDP defined on a uniform grid in $[0,1]^d$, i.e. $V_N(s) = V_N(s_i)$ if s is in the i^{th} partition element of $[0,1]^d$ centered at s_i. Then we have:*

$$\|V - V_N\| = O\left(\frac{1}{N^{\frac{1}{d}}}\right). \tag{2.40}$$

If V is continuously differentiable at higher orders, we can increase the rate of convergence by using higher order splines. For example if we use a 1-spline (piece-wise linear approximation) $\hat{V}_N$ to V in (2.39) then by doing a second order Taylor expansion of V within each triangulation element we can show that the bound corresponding to (2.39) becomes the sup norm of the hessian of V times the *square* of the maximum diameter of hte partition elements induced by the triangulation of S. In this case we get the following corollary to Theorem 2.2 due to Santos and Vigo (1996):

Corollary: *Let V_N be a 1-spline or piecewise linear function approximation to V, the fixed point $V_N = \Gamma_N(V_N)$ to the contraction operator that performs exact integration of any 1-spline function defined on the uniform triangulation with knot points located at the uniform grid points of $[0,1]^d$. Then we have:*

$$\|V - V_N\| = O\left(\frac{1}{N^{\frac{2}{d}}}\right). \tag{2.41}$$

As I noted in section 2.5, it is not computationally feasible to do policy iteration with exact integrations of m-spline approximations to V when $m > 0$. This is due to the fact the the approximate Markov operators $E_{a,N}$ can no longer be represented by an $N \times N$ matrix as in equation (2.11): while the inverse operators $(I - \beta E_{a,N})^{-1}$ that are key to the policy evaluation step of policy iteration exist "in theory", they generally can only be explicitly calculated when $E_{a,N}$ has a matrix representation. Thus, one must generally rely on successive approximations to approximate the fixed point $V_N = \Gamma_N(V_N)$ when using these higher order exact integration approaches. However note that while higher order methods do increase the rate of convergence, they do not break the curse of dimensionality. Similarly, it is not difficult to show that it is impossible to avoid the curse of dimensionality by using other non-uniform triangulations of the state space. Using the concept of *covering numbers* and *metric entropy* (see, e.g. Dudley, 1978), any triangulation of S which has maximum diameter ϵ requires a minimum of $N(\epsilon) = o(1/\epsilon^d)$ elements, or equivalently, any triangulation defined from a grid of N points will have a maximum diameter of order $O(1/N^{\frac{1}{d}})$. So one cannot avoid the curse of dimensionality by exact integration of higher order spline approximations to V using special triangulations of S.

I now briefly outline how one can derive error bounds for $\|V - V_N\|$ for the case where V_N is calculated using the quadrature grid approach described in section 2.5 and in Tauchen and Hussey (1991). The error bounds for $\|V - V_N\|$ in this case are the same as for uniform discretizations: $\|V - V_N\| = O(1/N^{\frac{1}{d}})$. In the interest of space I will only briefly outline the key ideas involved in deriving this error bound, referring the reader to Davis and Rabinowitz (1984) for details. Recall that in unidimensional quadrature one chooses N quadrature abscissa and N weights so that the approximate integral on the right hand side of equation (2.17) is exact for all polynomials of degree $2N - 1$ or less. Then an error bound for the approximate integral of a general function f can be derived using

Jackson's Theorem: *If f is Lipschitz continuous function on $[0,1]$ with Lipschitz bound K, then there exists a polynomial of degree N which is uniformly within $3K/N$ of f, so that with N point Gaussian quadrature we have:*

$$\left| \int_0^1 f(s)ds - \sum_{i=1}^{N} w_i f(s_i) \right| \leq \frac{3K}{2N-1}. \tag{2.42}$$

If we let K denote the maximum Lipschitz bound of the functions $V(\cdot)p(\cdot|s,a)$ as s ranges over the $[0,1]$ interval, then it follows that $\|E_a(V) - E_{a,N}(V)\| \leq 3K/(2N-1)$. We can extend this bound to the d-dimensional case using a bound for product rule quadrature due to Haber (see section 5.6 of Davis and Rabinowitz). Haber's bound for multivariate integration shows that if N point rule is used in each dimension of a product rule, then the absolute integration error will be $O(1/N)$. Thus, if N points are used, $N^{\frac{1}{d}}$ in each of d-dimensions, it follows that the error in using a quadrature grid to construct $E_{a,N}$ will be $O(1/N^{\frac{1}{d}})$. I summarize this result as:

Theorem 2.3 *Let V_N be an approximate value function computed from a product rule quadrature grid using weights which are products of the corresponding unidimensional quadrature weights and transition probability p_N given in (2.17). Then we have:*

$$\|V - V_N\| = O\left(\frac{1}{N^{\frac{1}{d}}}\right). \tag{2.43}$$

I conclude this section with figure 2.1, which summarizes the 5 grids used in the numerical comparisons in sections 4 and 5.

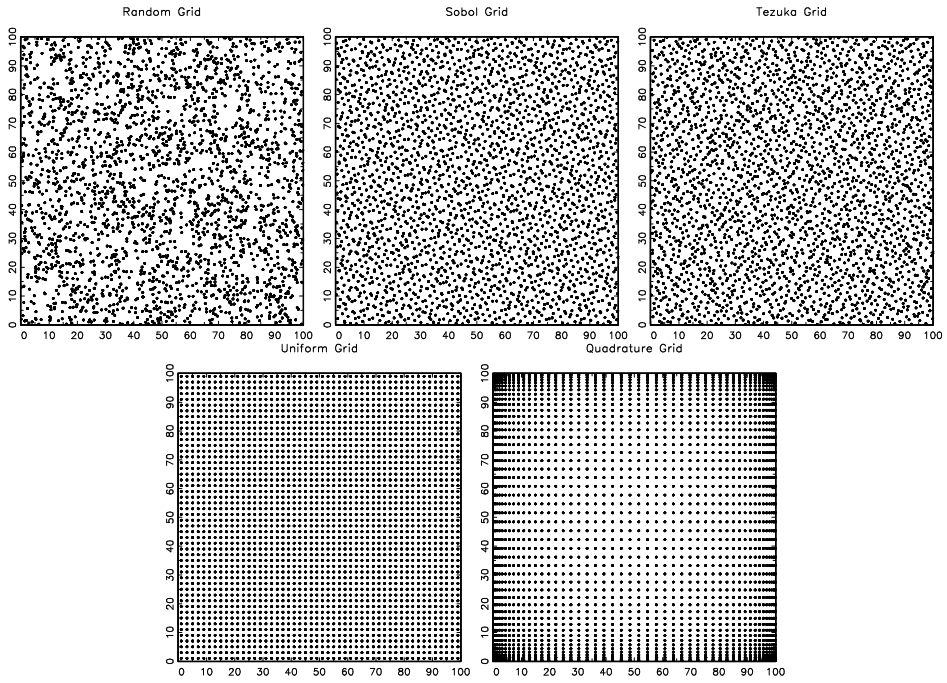

Figure 2.1 Alternative 2500 Point Grids for a Two-DimensionalState Space

3. Test Problems Used in the Numerical Comparisons

This section presents two test problems with analytic solutions that are used to evaluate the accuracy of the numerical solutions presented in sections 4 and 5. The test problems are a unidimensional replacement problem (regenerative optimal stopping problem) and a unidimensional search problem (non-regenerative optimal stopping problem). It turns out that both test problems are "irregular" in the sense that the transition densities are discontinuous (indeed, the transition density doesn't even exist in the search problem), and the state spaces are unbounded. Nevertheless in sections 4 and 5 I will show that by smoothing the transition density and truncating the state space it is possible to apply the methods described in section 2 to solve these irregular problems. I conclude this section by describing an approach due to Ken Judd (private communication) which allows me to construct analytic solutions to multidimensional MDP problems from solutions to unidimensional problems.

3.1 Optimal Replacement Problem

This problem is described in Rust (1985, 1986, 1996), but for completeness I briefly describe the solution here. Consider a durable whose state $s_t \in R_+$ can be interpreted as a measure of the accumulated utilization (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 $A(s) = \{0, 1\}$ where $a_t = 1$ 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 utilization of the asset each period has an exogenous exponential distribution. This corresponds to a transition probability p given by:

$$p(ds_{t+1}|s_t, a_t) = \begin{cases} 1 - \exp\{-\lambda(ds_{t+1} - s_t)\} & \text{if } a_t = 0 \text{ and } s_{t+1} \geq s_t \\ 1 - \exp\{-\lambda(ds_{t+1} - 0)\} & \text{if } a_t = 1 \text{ and } s_{t+1} \geq 0 \\ 0 & \textit{otherwise.} \end{cases} \tag{3.1}$$

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:

$$u(s_t, a_t) = \begin{cases} -c(s_t) & \text{if } a_t = 0 \\ -[\overline{P} - \underline{P}] - c(0) & \text{if } a_t = 1. \end{cases} \tag{3.2}$$

Bellman's equation takes the form:

$$\begin{aligned} V(s) = max\Big[& -c(s) + \beta \int_s^\infty V(s')\lambda \exp\{-\lambda(s' - s)\}ds', \\ & -[\overline{P} - \underline{P}] - c(0) + \beta \int_0^\infty V(s')\lambda \exp\{-\lambda(s')\}ds'\Big]. \end{aligned} \tag{3.3}$$

Rust (1986) differentiated Bellman's equation to derive a differential equation for V. The differential equation is a *free boundary value problem* since the boundary condition:

$$V(\gamma) = [\overline{P} - \underline{P}] + V(0) = -c(\gamma) + \beta V(\gamma) = \frac{-c(\gamma)}{1-\beta}, \tag{3.4}$$

is determined endogenously. Rust showed this differential equation has the following closed-form solution:

$$V(s) = max\Big[\frac{-c(\gamma)}{1-\beta}, \frac{-c(\gamma)}{1-\beta} + \int_s^\gamma \frac{c'(y)}{1-\beta}[1 - \beta e^{-\lambda(1-\beta)(y-s)}]dy\Big], \tag{3.5}$$

where γ is the unique solution to:

$$[\overline{P} - \underline{P}] = \int_0^\gamma \frac{c'(y)}{1-\beta}[1 - \beta e^{-\lambda(1-\beta)y}]dy. \tag{3.6}$$

It follows that the optimal decision rule is given by:

$$\alpha(s) = \begin{cases} 0 & \text{if } s \in [0, \gamma] \\ 1 & \text{if } s > \gamma. \end{cases} \tag{3.7}$$

Notice that the transition density $p(s'|s,a)$ has a discontinuity at $s' = s$ when $s > 0$: this discontinuity arises from an "irreversibility" condition that except for replacement (which constitutes complete regeneration) the durable can only deteriorate with age.

3.2 Optimal Search Problem

Consider an unemployed worker who receives one job offer each period while searching, a random draw from a density $f(w)$. If the worker accepts a wage offer w, the search problem terminates and the worker is assumed to be employed forever after at wage w. If the worker reject a wage offer w the worker receives an unemployment benefit of u and continuous searching in the next period. Bellman's equation for this problem is give by:

$$V(w) = max\left[\frac{w}{(1-\beta)}, u + \beta\int_0^\infty V(w')f(w')dw'\right]. \tag{3.8}$$

As is well known, the optimal search strategy involves accepting any offer greater than the *reservation wage* $\overline{w}$ given by:

$$\overline{w} = \frac{(1-\beta)\left[u + \beta\int_{\overline{w}}^\infty w' f(w')dw'\right]}{1 - F(\overline{w})}. \tag{3.9}$$

Given $\overline{w}$, the value function is given by

$$V(w) = \begin{cases} \dfrac{w}{(1-\beta)} & \text{if } w \geq \overline{w} \\ \dfrac{\overline{w}}{(1-\beta)} & \text{if } w < \overline{w}. \end{cases} \tag{3.10}$$

Notice that the transition probability density $p(s'|s,a)$ for the search problem does not exist when the decision is made to stop searching: in this case $p(s'|s,a)$ is a unit probability mass (also known as a dirac delta function) on the accepted wage offer w.

3.3 Generating Multidimensional Test Problems

It is rare that we can find closed-form solutions to one-dimensional MDPs and rarer still that we can find closed-form solutions to multi-dimensional MDP problems. However Ken Judd suggested a way to generate closed-form solutions to a multii-dimensional family of problems from a closed-form solution to a one dimensional problem. Judd's idea is implict in Lemma 3.1 below. Consider a multidimensional MDP problem with the following additive-separable structure:

$$u(\mathbf{s}, \mathbf{a}) = \sum_{i=1}^{N} u_i(s_i, a_i) \tag{3.11}$$

$$p(\mathbf{s'}|\mathbf{s}, \mathbf{a}) = \prod_{i=1}^{N} p_i(s_i'|s_i, a_i) \tag{3.12}$$

$$A(\mathbf{s}) = A_1(s_1) \times A_2(s_2) \cdots A_N(s_N) \tag{3.13}$$

Note that each s_i represents the state of "task" i. Equation (3.12) states that task states evolve independently. For example, the task of "driving to work by automobile" may be separable from task of "debugging a computer program at work", and the task of "deciding where to go for lunch", and so forth. The value function for this MDP problem is given by:

$$V(\mathbf{s}) = \max_{\mathbf{a} \in A(\mathbf{s})} \left[u(\mathbf{s}, \mathbf{a}) + \beta \int_{\mathbf{s'}} V(\mathbf{s'}) p(\mathbf{ds'}|\mathbf{s}, \mathbf{a}) \right]. \tag{3.14}$$

Lemma 3.1: *If $u(\mathbf{s}, \mathbf{a})$ has the additive structure given in equation (3.11) and $p(\mathbf{s'}|\mathbf{s}, \mathbf{a})$ satisfies the independence decomposition in (3.12), and the action sets $A(\mathbf{s})$ have the product structure given in (3.13), then the value function $V(\mathbf{s})$ has an additive decomposition given by:*

$$V(\mathbf{s}) = \sum_{i=1}^{N} V_i(s_i). \tag{3.15}$$

where each V_i is given by:

$$V_i(s_i) = \max_{a_i \in A_i(s_i)} \left[u_i(s_i, a_i) + \beta \int_{s_i'} V_i(s_i') p_i(ds_i'|s_i, a_i) \right]. \tag{3.16}$$

Using Lemma 3.1 we can construct closed-form solutions to multidimensional MDPs by simiply adding closed-form expressions for the value functions for one-dimensional problems. Although the multidimensional problems formed this way are in some sense no more difficult to solve than the original one-dimensional problem (such as the test problems given in sections 3.1 and 3.2), the algorithms that we use to solve multidimensional problems formed in this way do not "know" that the problem has this simple structure. Thus, none of the methods we evaluate for the multidimensional test problems in section 5 exploit the fact that the value function can be written as a simple sum of unidimensional value functions.

4. Numerical Comparisons of Methods in One-Dimensional Test Problems

This section presents results of the numerical comparison of five different discrete approximation methods in the two one-dimensional test problems described in section 3. The approximation methods compared are 1) uniform grids with exact integration of 0-spline approximations to the value function, 2) quadrature grids, 3) Sobol' grids, 4) Tezuka grids, and 5) "random" grids. In this study I used the Uniform random number in the *Gauss* programming language (a linear congruential uniform random number generator), and I used algorithms in the *FINDER* software package of Papageorgiou and Traub (1996) to generate Sobol' and Tezuka grids. Quadrature grids (and associated weights) were generated by the Gauss-Legendre algorithm from *Numerical Recipes* (Press, *et. al.* 1992).

Figure 4.1 illustrates the convergence of the value function V_N to V for the replacement problem with $S = [0, 100]$, $\lambda = .8$, $\overline{P} - \underline{P} = 100000$, and $c(s) = 150s$ for $N = \{50, 100, 150, 500\}$.[10] For convenience I solved the replacement problem as a cost-minimization problem rather than a utility maximization problem so the value functions would be increasing and positive. The true value function V appears as the uppermost curve in all four panels of figure 4.1. Notice that in all cases the V_N approach V monotonically from below as N increases. This was also true for all of the other methods. This is an important finding, suggesting that there is a common downward bias to all of the methods. I have been unable to present a formal proof of the fact that all of the methods lead to downward biased estimates of V, but I suspect it has something to do with Jensen's inequality. The intuitive explanation of the finding goes as follows: since V_N is the minimum of two functions $u_1 + \beta E_{1,N} V_N$ and $u_2 + \beta E_{2,N} V_N$ (corresponding to the decisions to keep and replace, respectively), even if the latter functions were approximately "unbiased" estimates of their limiting values $u_i + \beta E_i V$, $i = 1, 2$, the fact that we are taking the minimum of the two functions means that V_N tends to be a downward biased estimate of V: by Jensen's inequality the expectation of the minimum is less than the minimum of the the expectations. As $N \to \infty$ the approximation error converges to 0 and the V_N increase monotonically towards V just as we would expect from Jensen's inequality as the variance of the random variables in the inequality converges to 0.

The fact that the bias is fairly uniform in s and monotonically decreasing in N suggests that there may be a simple "bias correction" that would allow us to accelerate the convergence of V_N to V. Note, however, that the presence of a truly uniform bias would not affect the implied decision rule α. We can see from figure 4.1 that the bias is not exactly uniform and that the implied optimal stopping threshold γ_N (the smallest value of s after which the value function V_N becomes flat) tends to increase monotonically in N from about $\gamma_N = 40$ when $N = 50$ to its limiting value of $\gamma = 54.386$. The fact that the $\{\alpha_N\}$ are a sequence of non-optimal policies would ordinarily

[10] The figure plots V_N at 5,000 uniformly spaced points in $[0, 100]$ and the *Gauss* plotting program automatically interpolates these points in producing the plots.

imply the $V_N \geq V$, since a non-optimal policy should have a higher cost than the optimal policy α. Thus, the "Jensen inequality bias" more than offsets the upward bias in $\{V_N\}$ caused by the fact that the $\{\alpha_N\}$ are non-optimal.

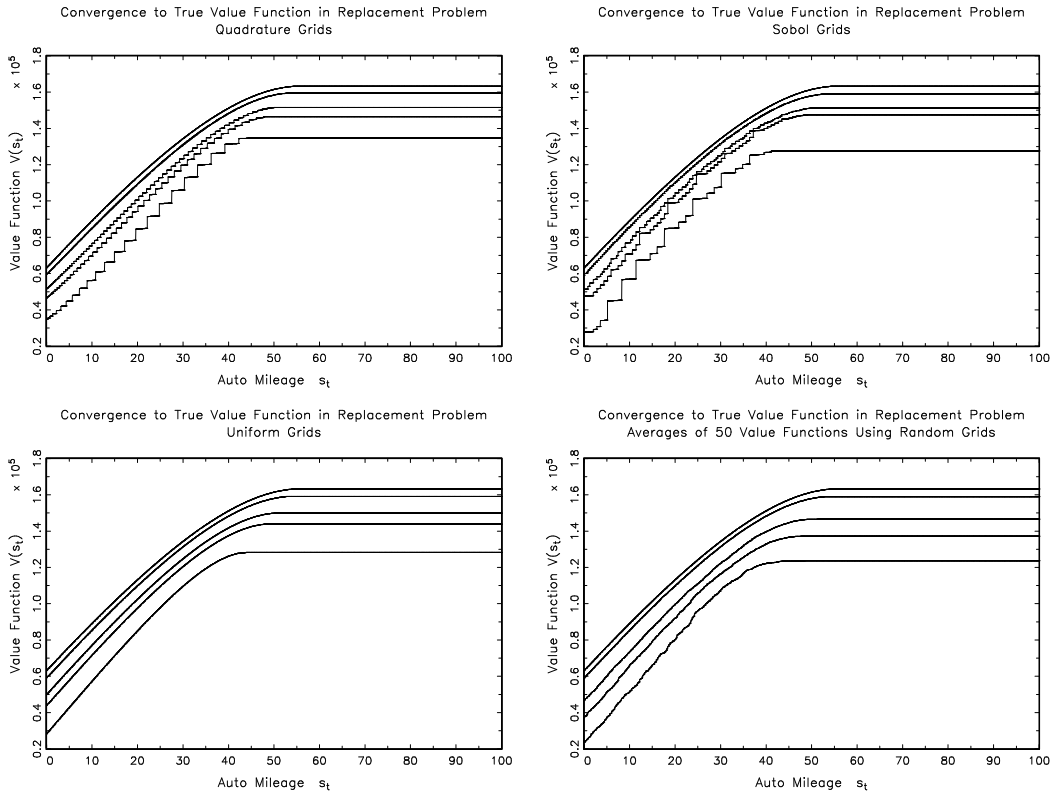

Figure 4.1 Convergence of Value Functions for 1-Dimensional Replacement Problem
(Panels plot V_N for N equal to $\{50, 100, 150, 500\}$. True V is uppermost curve in each panel)

The other thing to notice from figure 4.1 is the discontinuity of the calculated value functions V_N.[11] It is easy to see from section 3.1 that the true V is a C^∞ function for almost all $s \in [0, 100]$: the only discontinuity occurs in the *derivative* of V at the optimal stopping threshold γ. However the calculated V_N for the quadrature and Sobol' grids display notable discontinuities for small values of N. This is due to the fact that V_N is a convex combination of the $p(s'|s, a)$ functions, and each of these functions is discontinuous at $s' = s$ as noted in section 3.1. However as $N \to \infty$ the discontinuities become smaller and smaller and V_N does indeed appear to converge uniformly to its limiting value. Although when sufficiently magnified any V_N will have a finite number of discontinuities, it is evident from figure 4.1 that by the time $N = 500$ that the discontinuities are negligible and V_N can essentially be treated

[11] The lower right panel of figure 4.1 plots averages of 50 "random" realizations of the $\{V_N\}$ generated by the RPI algorithm. The averaging obviously smooths the realized value functions: individual realizations of V_N from the RPI algorithm are just as discontinuous as the V_N generated by quadrature and Sobol' grids.

as a continuously differentiable function. Note that for V_N's calculated for uniform grids with exact integration, I used a 1-spline to interpolate the V_N as a piecewise linear function and this is why the V_N's are smooth for all N. By doing this I have given a somewhat unfair advantage to the uniform grid approach since the V_N were actually calculated using a 0-spline (step function) approximation as noted in section 2.5. If I would have plotted the actual 0-spline approximation, then it too would have exhibited discontinuities similar to those for V_N calculated using the quadrature and Sobol' grids. The point of doing this is to show that despite the strong advantages of the uniform grid in this case (i.e. the fact that the uniform grid uses exact integration, that the uniform grid has minimal discrepancy in the 1-dimensional case, and the fact that I used linear interpolation to make the calculated V_N's appear as smooth as possible), the plots show that the V_{500}'s calculated by each of these the methods are virtually identical. Obviously, any of the V_N can be "smoothed" by some auxiliary interpolation or approximation procedure, however we see that when N is sufficiently large (such as $N \geq 500$ in this case), the solutions will be uniformly close to a smooth solution V and will therefore be smooth themselves. In particular for $N \geq 500$, individual realizations of V_N from the RPI algorithm are just about as smooth as the V_N calculated by any of the other methods.

Note that the discontinuity in $p(s'|s,a)$ violates the Lipschitz continuity conditions in Rust (1997) necessary to prove uniform convergence of the RPI algorithm. However as well known, the expected error in monte carlo integration converges to 0 at rate $1/\sqrt{N}$ despite the presence of discontinuities in the integrand. This is also true for methods based on deterministic grids provided the grids are uniformly distributed as defined in section 2.5. However while discontinuities to not affect *pointwise* convergence for *fixed* integrands, it can affect *uniform* convergence over a *class* of integrands. This can be easily seen in the replacement problem for points $s > max[s_1, \ldots, s_N]$. In that case, the transition probability p_N given in equation (2.16) is undefined since the denominator of (2.16) is 0 due to the fact that $p(s'|s,a) = 0$ for $s' < s$ when $a = 1$ (do not replace). For such points we need an auxiliary definition to be able to construct $V_N(s)$ for $s > max[s_1, \ldots, s_N]$. In the present case, due to the monotonicity of the value function, a natural extension is to assume that $V_N(s) = max_{i=1,\ldots,N}[V_N(s_i)]$ for $s > max_{i=1,\ldots,N}\, s_N$. Note that the fact that all of the grids we are considering are uniformly distributed implies that the set of s satisfying $s > max[s_1, \ldots, s_N]$ is "negligible" in the sense that the Lebesgue measure of this set converges to zero at rate $1/N$ as $N \to \infty$. In practice we simply ignore the approximation error on this negligible part of the state space in the absence of some special reason for computing V_N there. In many cases it is sufficient that V_N be a good approximation to V over the same grid of points $\{s_1, \ldots, s_N\}$ that we used to calculate V_N.

Of course in any problem where $p(s'|s,a)$ is a continuous or differentiable function of s for each $s' \in S$ and $a \in A$, then V_N will be a correspondingly continuous or differentiable function of s (for almost all $s \in S$) since it is the maximum of a finite number of convex combinations of the $p(s'|s,a)$ functions. It is possible to guarantee that individual realizations of V_N are smooth by solving a "smoothed" MDP problem that eliminates the discontinuities in $p(s'|s,a)$. There are a variety of ways to smooth the density in different problems: in the replacement problem one

convenient way is to smooth $p(s'|s,a)$ by replacing it with $p_\sigma(s'|s,a)$, an asymmetric truncated double exponential distribution where $1/\sigma$ denotes the parameter of a "reflected" exponential distribution, where the reflection occurs at the point of discontinuity $s' = s$ in the original unsmoothed exponential transition density given in equation (3.1). It is easy to see that as $\sigma \to 0$ we have $p_\sigma \to p$ both pointwise and uniformly. The right panel of figure 4.2 illustrates the use of the double exponential approximation to smooth the discontinuity in the original exponential transition probability $p(s'|s,a)$.

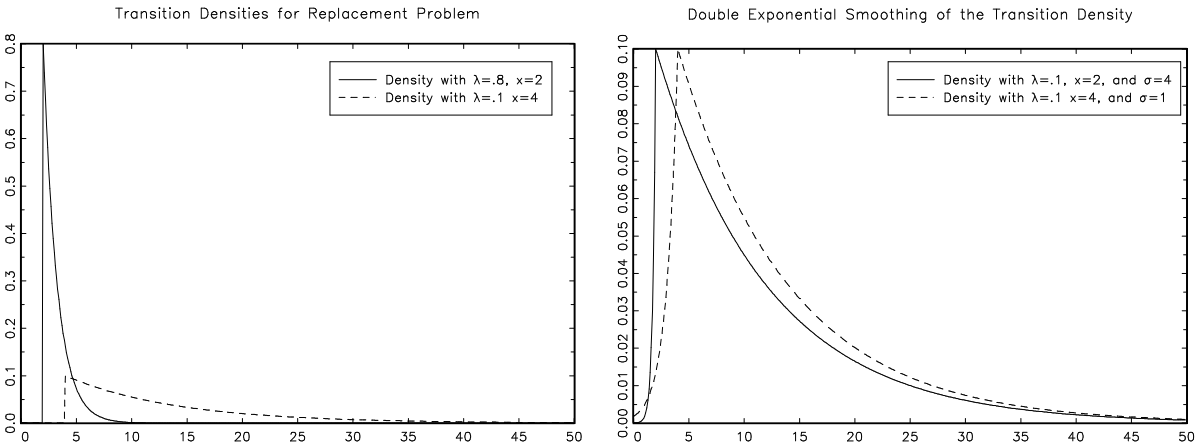

Figure 4.2 Smoothing the Transition Density

Using inequality (2.4) it is easy to show that if V_σ if the value function for an MDP problem with a smoothed transition density p_σ then if p_σ converges uniformly to p as $\sigma \to 0$, then $\|V_\sigma - V\| \to 0$ as $\sigma \to 0$. While the use of smoothing to eliminate discontinuities in p and V_N might seem like a good idea, it creates an additional complication because now there are two sources of approximation error: 1) the integration error, and 2) the "smoothing error". If $V_{\sigma,N}$ denotes the approximate solution to the smoothed problem, then to evaluate $\|V_{\sigma,N} - V\|$ we can invoke the triangle inequality and show that this error is bounded by $\|V_{N,\sigma} - V_\sigma\| + \|V_\sigma - V\|$. There will generally be a "bias vs. variance" tradeoff in the use of smoothing: larger values of σ tend to smooth the problem reducing the proportionality constants in the error bounds to that $V_{\sigma,N}$ is a better approximation to V_σ for any value of N (the "variance reduction effect"), yet at the same time the larger the value of σ the greater the distortion in V_σ from V (the "bias effect"). In general, it is difficult to determine the optimal tradeoff between the bias and variance effects, and I leave it to a future study to investigate this issue in more detail. I simply conclude this discussion with figure 4.3 which shows 10 realizations of V_N from the RPI algorithms with smoothing parameter equal to $\sigma = 0$ (i.e. no smoothing) and $\sigma = 1$ when $N = 50$. The figure clearly illustrates the "bias vs. variance tradeoff": not only are the realizations of the V_{50} functions much smoother, but there is far less variation about their mean. Indeed, the sample average of $\|\tilde{V}_{\sigma,50} - V_\sigma\|$ for these 10 realizations is $17,000$, far less than the average maximum error of $50,000$ when $\sigma = 0$.

On the other hand there is far more bias: without smoothing the V_{50} functions underestimate V by approximately 50000 whereas with smoothing the $V_{\sigma,50}$ functions underestimate V by about 61200. The bias is partly due to the fact that V_σ underestimates V by about 40000 as you can see from figure 4.4. [12] It appears, therefore, that in this case smoothing has not achieved a significant increase in accuracy: the increased bias due to smoothing outweighs the "variance reduction effect". I investigated the effects of smoothing for deterministic grids such as Sobol' and Tezuka grids and for a range of smoothing parameters and come to a similar conclusion: *smoothing does not appear to help a great deal in the replacement problem.*

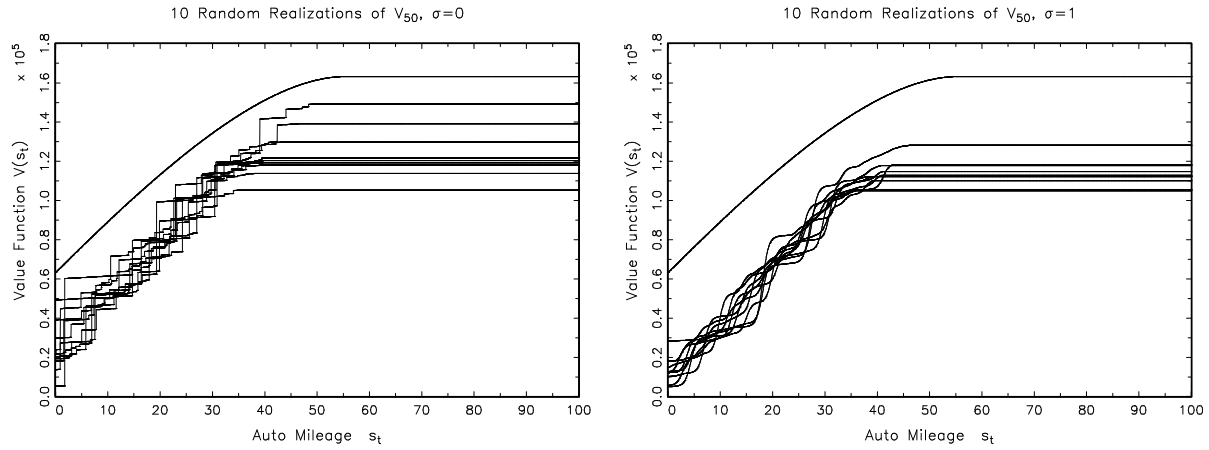

Figure 4.3 Effect of Smoothing on Realizations of the RPI Algorithm

In the absence of simple "bias corrections" it is not evident from figure 4.3 and other numerical experiments (not shown) that the reduction in variance due to smoothing would be sufficiently great to justify use of smoothing techniques. In the remainder of this paper I will not use smoothing, except for a brief discussion of the use of smoothing in the search problem below. I think the most striking aspect of figure 4.3 is that every realization of V_N is significantly below V. This underscores the "Jensen inequality bias" phenomenon noted earlier, but the bias does not seem to occur on average (as would be expected from Jensen's inequality), but *every realization* of V_N appears to be below V using any of the random or deterministic discretization methods. This suggests that there might be something more than Jensen's inequality going on here. It also suggests that there could be large payoffs to finding simple bias correction formulas in order to adjust for the "smoothing bias" $V - V_\sigma$ and the "Jensen inequality bias" $V_\sigma - V_{\sigma,N}$ that seems to lead to systematic downward biases for when the grid size N is relatively small.

[12] V_σ was computed using a uniform grid with $N = 1000$ points.

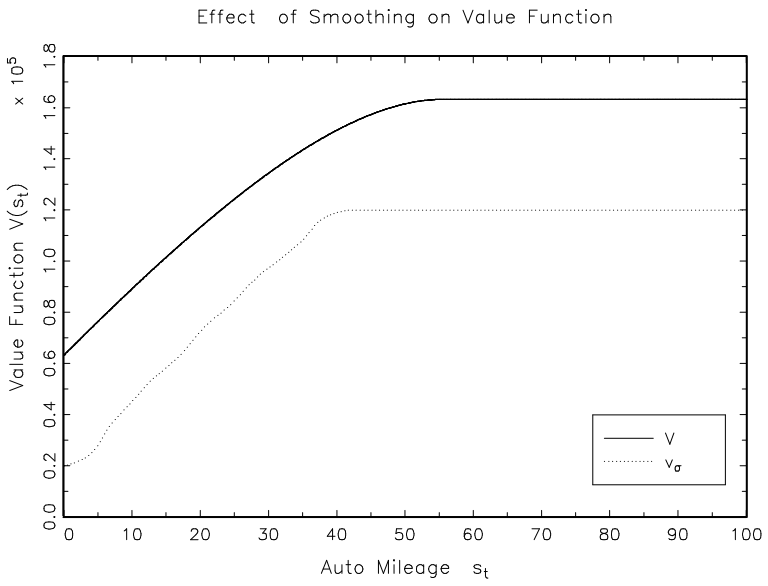

Figure 4.4 Effect of Smoothing on the Value Function

The next set of figures illustrate the convergence of V_N to V in the one-dimensional search problem, with $u = -.2$ (implying that the unemployment benefit net of search costs is negative), $\beta = .95$, and a exponential wage offer distribution with parameter $\lambda = .02$ (implying an expected offered wage of $E\{w\} = 50$). The reservation wage for this problem (see equation (3.9)) is $\overline{w} = 93.6$, so the value of search is $\overline{w}/(1-\beta) = 1872$. I solved the search problem on a truncated state space $S = [0, 200]$. Given that the probability of a wage offer greater than 200 is less than 2%, the truncation of the state space has negligible effect on the solution of the problem.

The numerical results indicate that the search problem is considerably "easier" than the replacement problem in the sense that I was able to obtain very accurate approximations V_N for relatively small values of N. Figure 4.5 plots V_N computed using uniform grids and random grids when $N = 30$. The V_{30} function computed from a uniform grid on S is virtually indistinguishable from the true V: indeed in figure 4.5 V and V_{30} are virtually the same line (V_{30} is the long-dashed curve, but it is virtually impossible to see the long dashes when they are directly superimposed over the solid curve that represents the true V). Approximation error in the random $\tilde{V}_{30}$ functions from the RPI algorithm are more evident, although the small dotted line that plots the average of 50 such realizations is very close to the true V. One reason the job search problem may be "easier" than the replacement problem is the assumption that continued search yields *IID* draws from a fixed wage offer distribution $f(w)$. This means that there is essentially only one integral to be approximated: the expected value of continuing search. The fact that the search problem involves essentially only one integral may be part of the reason why there appears to be much less of a "Jensen's inequality bias" in the search problem: as we can see from figure 4.5 there is no evidence of upward bias in the V_N's that we would expect if Jensen's inequality was operative.

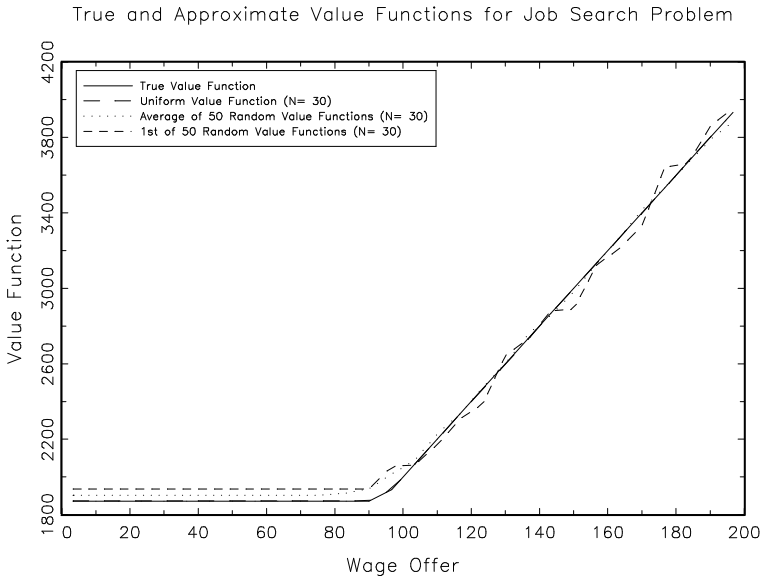

Figure 4.5 True and Approximate Value Functions in the 1-Dimensional Search Problem

Unlike the replacement problem, some form of smoothing is essential in order to compute "sensible" solutions V_N in the search problem. The reason is that, as noted in section 3.2, the transition density is not defined in the search problem when $a = 1$ (accept a job offer). The transition probability is a unit mass on the accepted wage, corresponding to a "Dirac" density function with infinite height on the accepted wage w. It is not difficult to modify the algorithms presented in section 2 to account for this: we simply set $p_N(s_i|s_j, a) = 1$ if $s_i = s_j$ and $p_N(s_i|s_j, a) = 0$ when $s_i \neq s_j$ and $a = 1$. Using this (identity) transition probability, it is straightforward to calculate a corresponding V_N by policy iteration. However the difficult comes when we want to evaluate V_N at points off of the grid. In this case $p_N(s_i|s, a) = 0$ for all $s \notin \{s_1, \ldots, s_N\}$, so that p_N in (2.16) is undefined for points off of the grid when $a = 1$. In this case we need to specify some auxiliary interpolation/extrapolation algorithm so that V_N is defined for all N. A simple approach is to use linear interpolation as described in equation (2.3). However an alternative approach is to use smoothing, using $p_\sigma(s'|s, a) = N(s, \sigma)$ (a normal density with mean s and standard deviation σ) as an approximation to a Dirac delta function when $a = 1$. It is easy to see that the smoothed RPI algorithm automatically interpolates and extrapolates $V_{\sigma,N}(s)$ at points s off the grid. For example if s lies between two grid points s_i and s_j, the smoothed value function $V_{\sigma,N}(s)$ will approximately equal a weighted average of $V_{\sigma,N}(s_i)$ and $V_{\sigma,N}(s_j)$ where the ration of these weights equals the ratio of normal densities $N(s, \sigma)$ evaluated at $s = s_i$ and $s = s_j$. As $\sigma \to 0$ the calculated V_N become increasingly jagged, falling to 0 when $s \notin \{s_1, \ldots, s_N\}$ and $a = 1$. For this reason smoothing was used in the search problem, and in figure 4.5 I used $\sigma = 1$ as the value for the smoothing parameter.

There are two main conclusions that can be drawn from figure 4.5: 1) all of the approximation algorithms seem to have "discovered" the fact that the search problem is significantly easier than the replacement problem: the

percentage approximation errors are far smaller for any given value of N than in the replacement problem, 2) there is no evidence that any of the approximation methods considered suffers from "Jensen inequality bias" in the search problem. This second finding suggests the need for a great deal of caution when searching for general "bias correction" formulae: in particular if Jensen's inequality is really the key to understanding this bias, we need to understand why the bias is so strong in the replacement problem but is virtually non-existent in the search problem.

The remainder of this section investigates convergence rates of the algorithms. The first question is whether the observed convergence rates satisfy the theoretical upper bounds derived in section 2. The answer is "yes" as illustrated in figure 4.6 which plots theoretical and predicted convergence rates for the expected error $E\{\|V - \tilde{V}_N\|\}$ of the RPI algorithm in the job search problem. Recall that Rust's (1997) bound predicts that the expected error in $\|V - \tilde{V}_N\|$ should decrease at rate $1/\sqrt{N}$. Figure 4.6 shows that the empirical convergence rate of $E\{\|V - \tilde{V}_N\|\}$ closely conforms to this theoretically predicted upper bound: in fact the empirical convergence rate (determined from a linear regression fit to the observed rate of convergence) is virtually identical to $K/\sqrt{N}$ predicted by Rust's bound in equation (2.20), and the empirically observed bounding constant $K = \exp\{7.269\} = 1435.1146$ is only 10% lower than the theoretically predicted bounding constant from (2.20) for the corresponding smoothed MDP and value function V_σ in the case $\sigma = 1$.

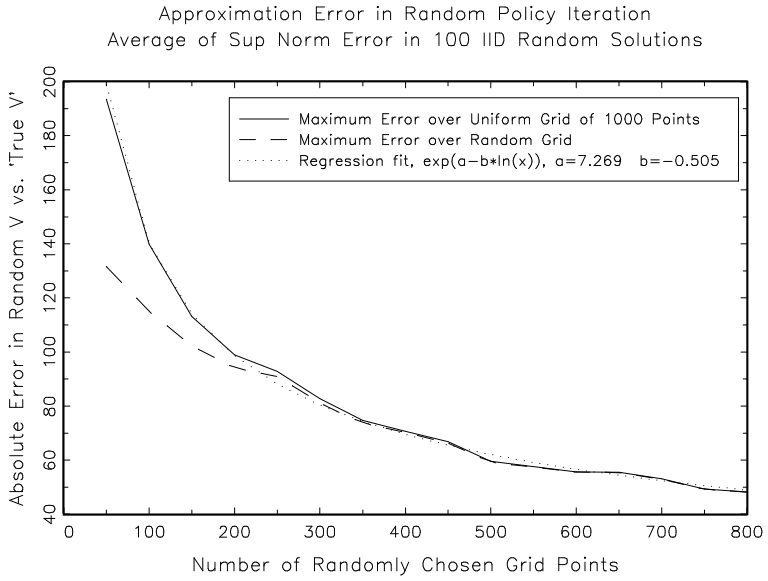

Figure 4.6 Predicted vs. Actual Convergence Rates of $E\{\|V - \tilde{V}_N\|\}$ in the 1-Dimensional Search Problem

Figure 4.7 compares theoretical vs. predicted convergence rates of $\|V - V_N\|$ in the replacement problem with $\lambda = .1$, $\overline{P} - \underline{P} = 50000$ and $c(s) = 200s$. The four panels plot the observed convergence rates for uniform grids, quadrature grids, Tezuka grids and random grids. The observed convergence rate for uniform grids, $N^{-.99}$, is very close to the theoretically predicted value of $1/N$ given in Theorem 2.2. This is also the convergence rate predicted for $\|V - V_N\|$ using quadrature grids, but we see the that actual rate of convergence is slightly slower, $N^{-.931}$. Theorem 2.1 predicts that the convergence rate for low discrepancy grids should be between the lower bound of $1/2N$ in the unidimensional case and the upper bound of $\log(N)^d/N$. Thus, if I regress the observed errors $\|V - V_N\|$ less $\log(N)$ on a constant term and slope term equal to $\log(\log(N))$, Theorem 2.1 predicts that the coefficient b on the slope term should be in the interval $(0, 1]$ for V_N computed from any low discrepancy grid. We see from the lower left panel of figure 4.7 that the observed $\hat{b} = .71$ does indeed satisfy this restriction. Finally the lower right panel of figure 4.7 plots the expected errors from the RPI algorithm. In this case we see that the observed rate of convergence $N^{-.612}$ is faster than the theoretically predicted upper bound of $1/\sqrt{N}$.

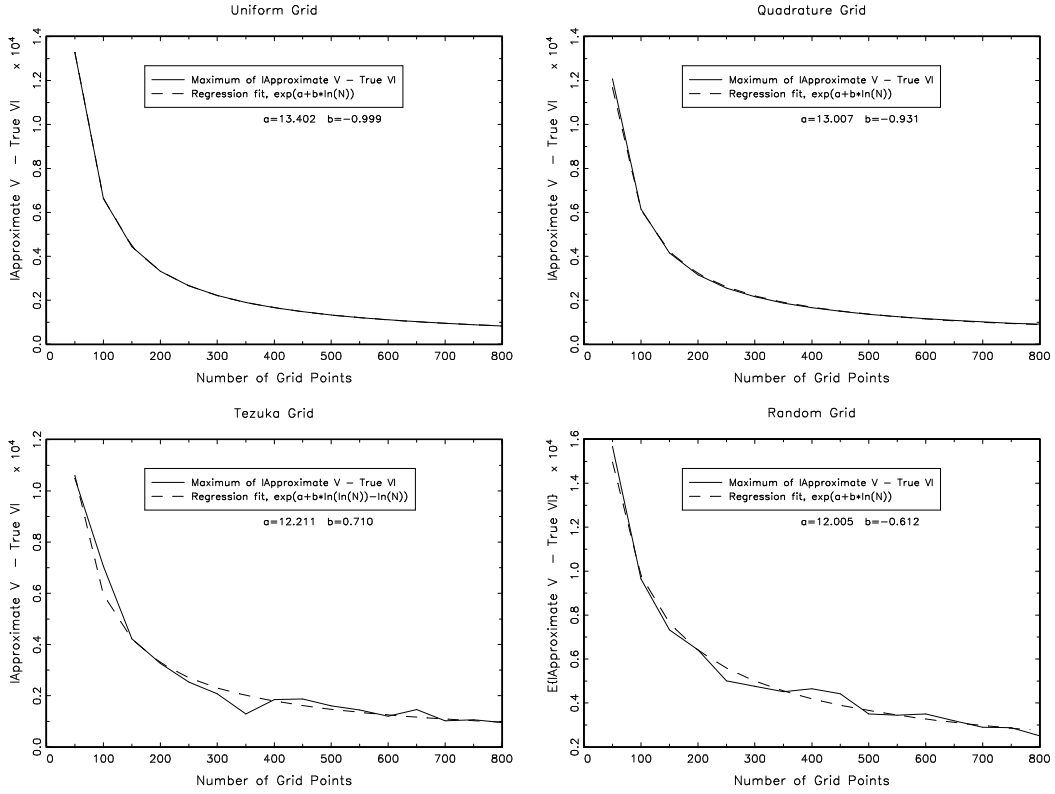

Figure 4.7 Predicted vs. Actual Convergence Rate for $\|V - V_N\|$ in the 1-Dimensional Replacement Problem

I conclude this section with figure 4.8 summarizes the relative efficiencies of the various algorithms in two different cases, $\lambda = .1$ and $\lambda = .8$, with $\overline{P} - \underline{P}$ and $c(s)$ set equal to the values described previously when $\lambda = .1$ and $\overline{P} - \underline{P} = 100000$ and $c(s) = 150s$ when $\lambda = .8$.[13] Since the transition density is discontinuous at $p(s|s,1) = \lambda$, there is a substantially larger discontinuity in the transition density when $\lambda = .8$ than when $\lambda = .1$ as you can see from the left hand panel of figure 4.2. We see that all of the methods performed comparably in the "high discontinuity" case $\lambda = .8$, however in the "low discontinuity" case $\lambda = .1$ policy iteration using Sobol' and Tezuka grids generated smaller errors than the other methods. The reason is clear from figure 4.7: although the convergence rate of $\|V - V_N\|$ is slightly lower for low discrepancy sequences, the bounding constant is lower for low discrepancy grids than for methods based on uniform or quadrature grids ($\exp(12)$ vs. $\exp(13)$). However all of the deterministic methods significantly outperformed the RPI algorithm in this case, just as we would have predicted from the analysis of their relative convergence rates in section 2. However it is somewhat surprising that policy iteration using exact integration and uniform grids did not do better than the other methods. *A priori* I would have expected this method to significantly outperform the other methods for several reasons: 1) the uniform grid has the smallest discrepancy in one-dimension, 2) the use of exact integration eliminates most of the error in the "integration subproblem", and 3) due to the discontinuity of the transition density $p(s'|s,a)$ when $a = 1$, the use of linear interpolation as a procedure for extending the discretized solution to the entire state space yields a much smoother, well behaved function V_N than for the other methods that adopt the self-approximating or Nyström approach to constructing p_N. The fact that these latter methods performed much better than expected suggests that they will also be superior in higher dimensional problems since the former methods are not subject to the curse of dimensionality while the convergence rates of uniform and quadrature grids ought to decrease at rate $1/d$ as predicted by Theorems 2.2 and 2.3 of section 2.

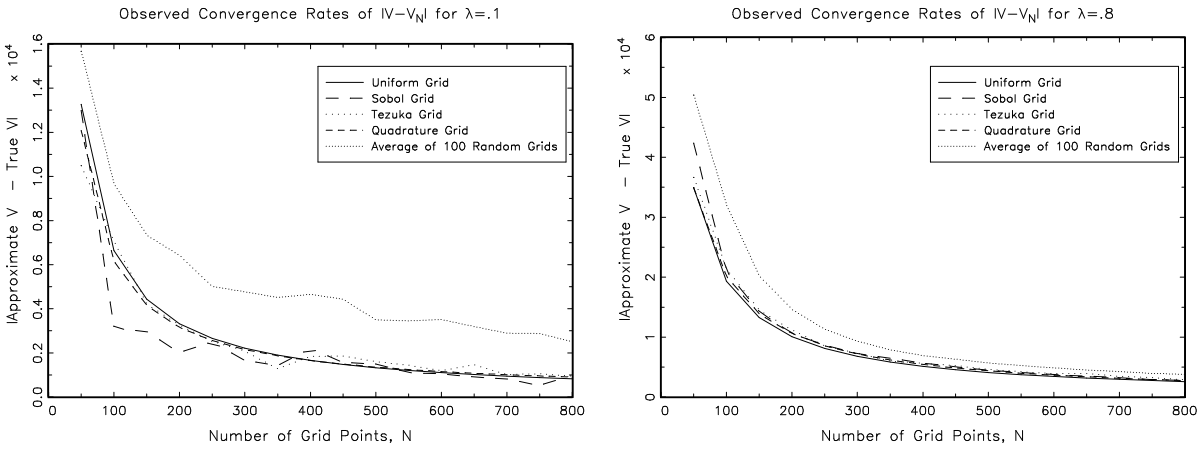

Figure 4.8 Summary of Observed Convergence of $\|V - V_N\|$ for Different Grids in Replacement Problem

[13] I made these adjustments so the optimal replacement threshold γ was roughly the same in the two cases: $\gamma = 65.698$ when $\lambda = .1$ and $\gamma = 54.386$ when $\lambda = .8$.

5. Numerical Comparisons of Methods in Two-Dimensional Test Problems

In this section I compare the performance of the five different approaches to discretization using artificial two-dimensional test problems created from the "convolution" of the one-dimensional search and replacement problems as described in Lemma 3.1. This lemma implies that an approximate solution to the multidimensional value function can be constructed by simply summing a one-dimensional approximation to the value function with itself. If I did this, the two dimensional problem would be essentially no more difficult than the one-dimensional problem: the only extra work involved would be simply adding the one-dimensional value functions together. None of the algorithms that I have evaluated in this section have exploited this special structure: instead, all algorithms treated the problem as a generic two-dimensional MDP problem. All of the solution methods that I evaluate in this section directly approximate the multidimensional value function using a value function for an embedded finite state MDP. Since the embedded MDP has the same general structure regardless of the dimension or structure of the original DDP problem, there doesn't appear to be any way for any of the methods to "discover" and exploit the special additive structure of the MDP problem. For this reason I believe the results in this section will be representative of the performance we can expect in more general two-dimensional DDP problems. Indeed, I will show that the test problems in this section are probably much harder for the algorithms than other problems that might be encountered in practice due to the fact that the transition density for the two-dimensional problem is product of univariate transition densities for the one-dimensional problem, and I will show that this can create a curse of dimensionality even when randomization is used. The problem is exacerbated by discontinuities in the univariate transition density: the discontinuities in the implied multivariate transition density becomes increasingly problematic the higher the dimension of the problem. However despite the irregular nature of the test problems, I show that the random and low-discrepancy policy iteration algorithms perform remarkably well, and systematically outperform policy iteration based on uniform or quadrature grids provided the discontinuity in the transition density is not "too large."

I begin by presenting plots of the true and approximate value functions for the two-dimensional job search problem. The state space for this problem is $S = [0, 200] \times [0, 200]$, and the other parameters of the search problem are the same as for the one-dimensional search problem discussed in section 4. Recall that in the job search problem the transition density corresponding to a decision to accept a wage offer is a dirac delta function, so some form of smoothing is necessary to compute estimates of the value function off of the grid for reasons already discussed in section 4. Just as in section 4 I smoothed the dirac function by assuming that there is slight amount of uncertainty in an accepted wage offer: that is, if the decision maker accepts a wage offer, instead of receiving a certain wage of w, they receive an uncertain draw from a $N(w, \sigma)$ distribution. I assumed a smoothing parameter of $\sigma = 2$ for this case, twice as large as in the one-dimensional case analyzed in section 4. In a two-dimensional problem, $\sigma = 2$ does not imply a great deal of smoothing as we can see from the plot of the smoothed transition density in figure 5.1. The smoothed transition density is still very close to a "spike" and attempting to compute the integral of this "spike" function by

taking a sample average of points is an obviously error-prone procedure unless the grid in the state space is quite fine: if there are no grid points underneath the spike, the approximate integral will greatly underestimate the true integral. Henceforth I will refer to this as the *needle in the haystack problem.*

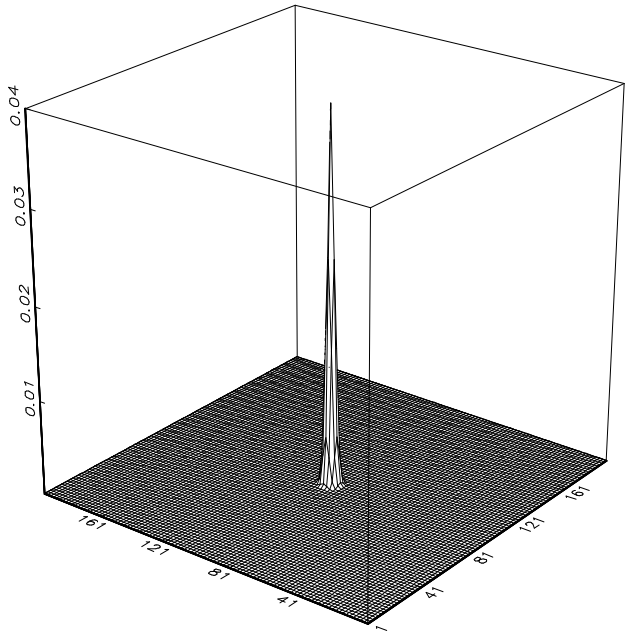

Figure 5.1 Smoothed Transition Density for the Two-Dimensional Job Search Problem (stop-searching)

Figure 5.2 plots the true V followed by V_N's for each of the five different grids (uniform, quadrature, random, Sobol' and Tezuka grids) using $N = 1600$ points. Remarkably, all of the methods produced quite accurate approximations to V in spite of the needle in the haystack problem. This is visually apparent in figure 5.1, which shows that all of the V_N's are about equally close to the true solution V. Of course, a solution based on a uniform grid with exact integration completely avoids the needle in the haystack problem, since the spike must fall in some element of the partition induced by the grid, the discretized transition probability p_N will capture this spike since it is calculated from the exact integral of the underlying continuous transition density $p(s'|s, a)$ over each square in the partition. Thus, the assumption that we can do exact integration of the transition density gives the uniform grid approach a substantial advantage. As expected, the V_N calculated using the uniform grid is visually the "smoothest" and most like the true solution V. However part of this is due to the fact that I have given the uniform grid an additional "unfair" advantage by using multilinear interpolation of the original solution, which is a step function on the 1600 squares in the partition of S induced by the uniform grid. If I had chosen to plot the original original solution (a 0-spline), it would have appeared substantially more jagged than the other V_N's in figure 5.1. Amazingly, despite the substantial advantages given to the uniform grid, it did not produce the most accurate estimate V_N of V: the maximum difference $\|V - V_N\|$ was 300 for the V_N calculated using the uniform grid, followed by 245 for the quadrature grid, 212 for the random

grid, 163 for the Tezuka grid and 152 for the Sobol' grid. The V_N's were approximately unbiased estimates of V just as in the one-dimensional search problem. Indeed the average errors in $V - V_N$ (computed over a uniform grid, with equal weighting at each grid point) were quite close to zero for all of the problems. Most of the approximation error was probably introduced by the smoothing: when I set $\sigma = 0$ and computed V_N using a uniform grid the maximum error was only $\|V - V_N\| = 50$. Thus, while the V_N's calculated using the low discrepancy grids had lower error than V_N's calculated from uniform or quadrature grids when $\sigma = 2$, the uniform grid is superior in the job search problem since we can calculate V_N with no smoothing, $\sigma = 0$, and get a lower maximum error.

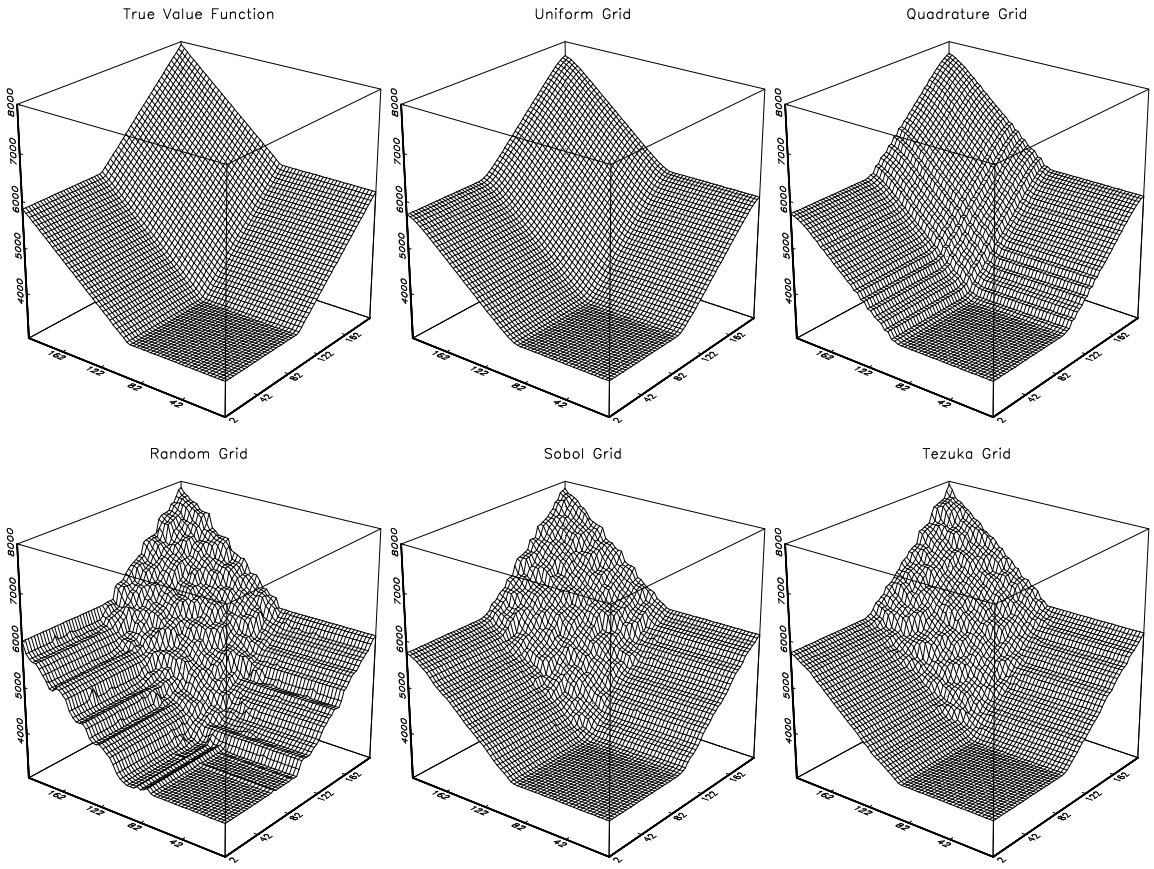

Figure 5.2 Value Functions for 2-Dimensional Search Problem

Figure 5.3 plots the implied decision rules α_N for the job search problem. The true decision rule α is defined by a partition of the state space into 4 regions: 1) a region where both "agents" continue to search, 2) a region where one agent stops searching since $w_1 > \overline{w}$ and the other agent continues to search since $w_2 < \overline{w}$, 3) a region where agent 1 continues to search ($w_1 < \overline{w}$) and agent 2 accepts a wage offer ($w_2 > \overline{w}$), and 4) a region where both agents stop searching and accept the wage offer pair (w_1, w_2). In figure 5.3 we see that the α_N's computed from the uniform and quadrature grids appear to be significantly more accurate than the α_N's calculated from the random and low discrepancy grids. The inaccuracies in the latter methods are concentrated mostly near the boundaries of the regions defining the different values of the decision rule. Along these boundaries the agent is indifferent between accepting an offer or continuing to search, so it is natural that most errors would occur here since slight errors in the expected values for each alternative can lead to nonoptimal actions being taken. To see this, note that the expected value for action a in state s is denoted by $V_N(s,a)$ and given by:

$$V_N(s,a) = u(s,a) + \beta \sum_{i=1}^{N} V_N(s_i) p_N(s_i|s,a). \tag{5.1}$$

The needle in the haystack problem can lead $V_N(s,a)$ to be a poor estimate of the true expected value function $V(s,a) = u(s,a) + \beta \int V(s') p(s'|s,a) ds'$ at points s which are not on the grid. Since $EV(s,a) = EV(s,a')$ for at least two actions $a = \alpha(s) \neq a'$ along these boundaries, it is not surprising that misclassifications are most likely to occur here. Further away from the boundary there is a strict gap between the expected utility of the best and second best alternatives, so it takes larger errors in the EV_N functions to lead to misclassifications away from the boundaries. Since the agent is close to being indifferent between various actions near these boundaries, the loss in discounted utility from these errors is not of great consequence. Indeed, since we are calculating the exact value functions via policy iteration, we can see that there is little consequence to these misclassifications from the fact that the value functions for each of the methods are approximately equal over all parts of the state space. Further analysis reveals that most of the misclassifications shown in figure 5.3 occur "off the grid", i.e. they occur when trying to predict the values of the $EV_N(s,a)$ at points s which are not in the grid $\{s_1, \ldots, s_N\}$ using to calculate V_N. The percentage of misclassifications at the grid points $\{s_1, \ldots, s_N\}$ is 0.0% for α_N calculated from the uniform grid with no smoothing ($\sigma = 0$), 6.55% for α_N calculated from the uniform grid with $\sigma = 2$, 3.3% for the quadrature grid, 4.6% for the Tezuka grid, 4.3% for the Sobol' grid, and 11.2% for the random grid. Thus, except for the random grid, the decision rules calculated from the low discrepancy grids are nearly as accurate as the decision rule calculated for the uniform and quadrature grids, despite the visual impression created in figure 5.3. However I do note that the decision rules calculated from the quadrature and uniform grids are remarkably accurate approximations of the true decision rule α.

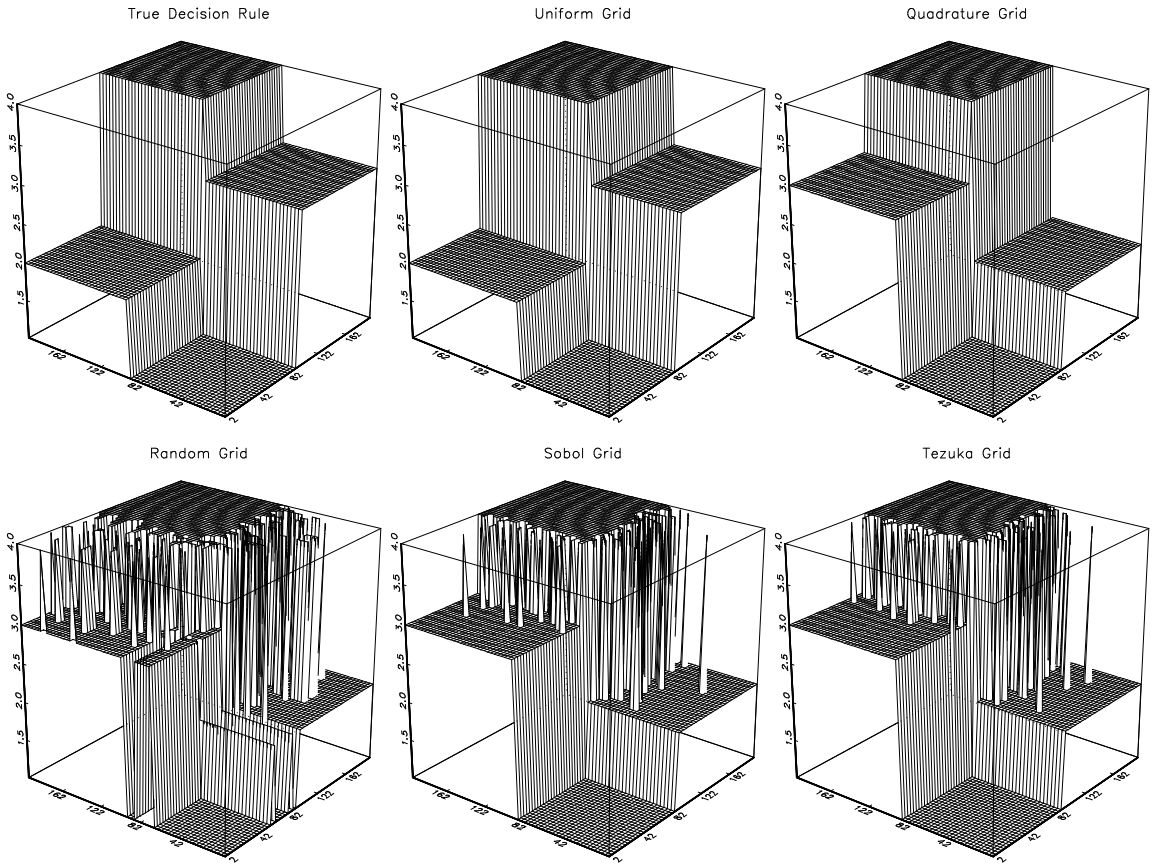

Figure 5.3 Decision Rules for 2-Dimensional Search Problem

Figure 5.4 plots the approximation errors $V - V_N$ for the the replacement problem with "small discontinuities", $\lambda = .1$ and no smoothing of the transition density, $\sigma = 0$. All of the approximate value functions were calculated using grids with $N = 2500$ points. We see that similar to the one-dimensional case, the V_N functions for each choice of grid systematically underestimates V. It is visually apparent that the V_N functions calculated from the uniform and quadrature underestimate V by a much larger amount than for the V_N functions calculated from the random and low discrepancy grids: the average value of $|V - V_N|$ over a grid of 3600 uniformly spaced points is 25,400 for the uniform grid, 21,700 for the quadrature grid, 9,600 for the random grid, 5500 for the Sobol grid, and 5600 for the Tezuka grid. The maximum absolute errors, (an approximation of $\|V - V_N\|$) were of similar magnitudes: 26500 for the uniform grid, 23600 for the quadrature grid, 12,200 for the random grid, 25400 for the Sobol' grid, and 7100 for the Tezuka grid. The maximum error for the Sobol' grid was much higher than the average error due to a puzzling tendency for the value function to take a sharp drop along the outer boundary of the state space as you can see in the middle panel of figure 5.4. These sorts of errors can occur due to the failure to smooth the transition density: whenever I evaluate V_N at a point s very close to the outer boundary of the state space, if there are no points s_i in the grid satisfying $s_i \geq s$, then it is easy to see that the denominator of the transition probability in equation (2.18) will be zero, and V_N will not

be defined for these points. However I verified that the steep drop in the value function also occurs for points s that are on the grid near the outer boundary. The asymmetric nature of the errors in this case is rather puzzling, i.e. the fact that value function falls precipitously only along one boundary of the state space and not the other. In any event I am confident that this is not a result of a computer programming error since the same code was used to generate the other two value functions and no such asymmetric decline on the boundaries are observed for random grids or Tezuka grids, and the drop off problem disappears for the Sobol case when smoothing is introduced.

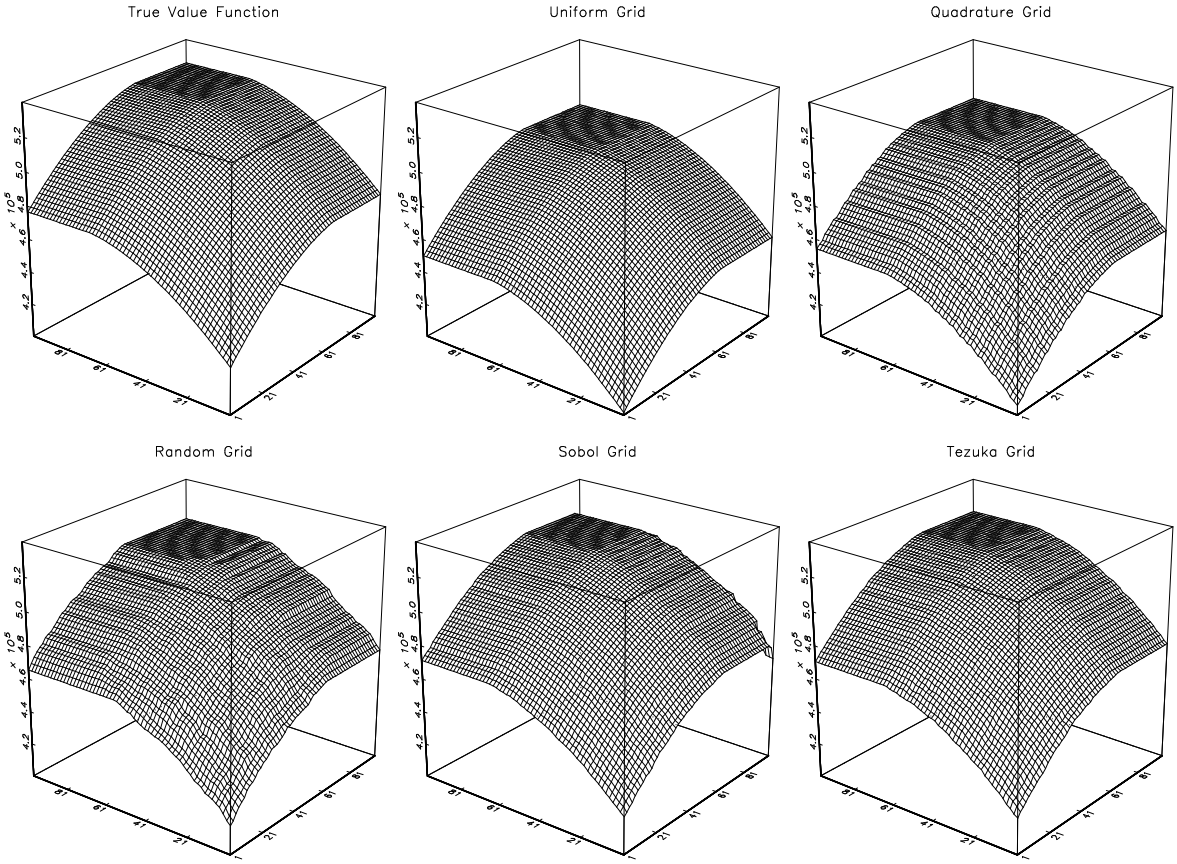

Figure 5.4 Value Functions for 2-Dimensional Replacement Problem with $\lambda = .1$

Figure 5.5 plots the corresponding decision rules. Similar to the search problem the true into 4 regions: 1) a region where both durables are kept, 2) a region where a durable 1 in state $s_1 > \gamma$ is replaced and durable 2 in state $s_2 < \gamma$ is kept, 3) a region where durable 1 in state $s_1 < \gamma$ is kept and durable 2 in state $s_2 > \gamma$ is replaced, and 4) a region where both durables states exceed the replacement threshold γ and are replaced. Just as in the search problem, the calculated decision rule α_N is most inaccurate for the random grid. The α_N's for the low discrepancy grids are virtually identical to the α_N's calculated for the quadrature and uniform grids except for slight "noise" around the boundaries of the different regions defining the decision rule. There is some additional errors at the back boundary of

the state space for the α_N corresponding to the Sobol' grid which corresponds to the underestimation of V_N in this area noted in figure 5.4 above.

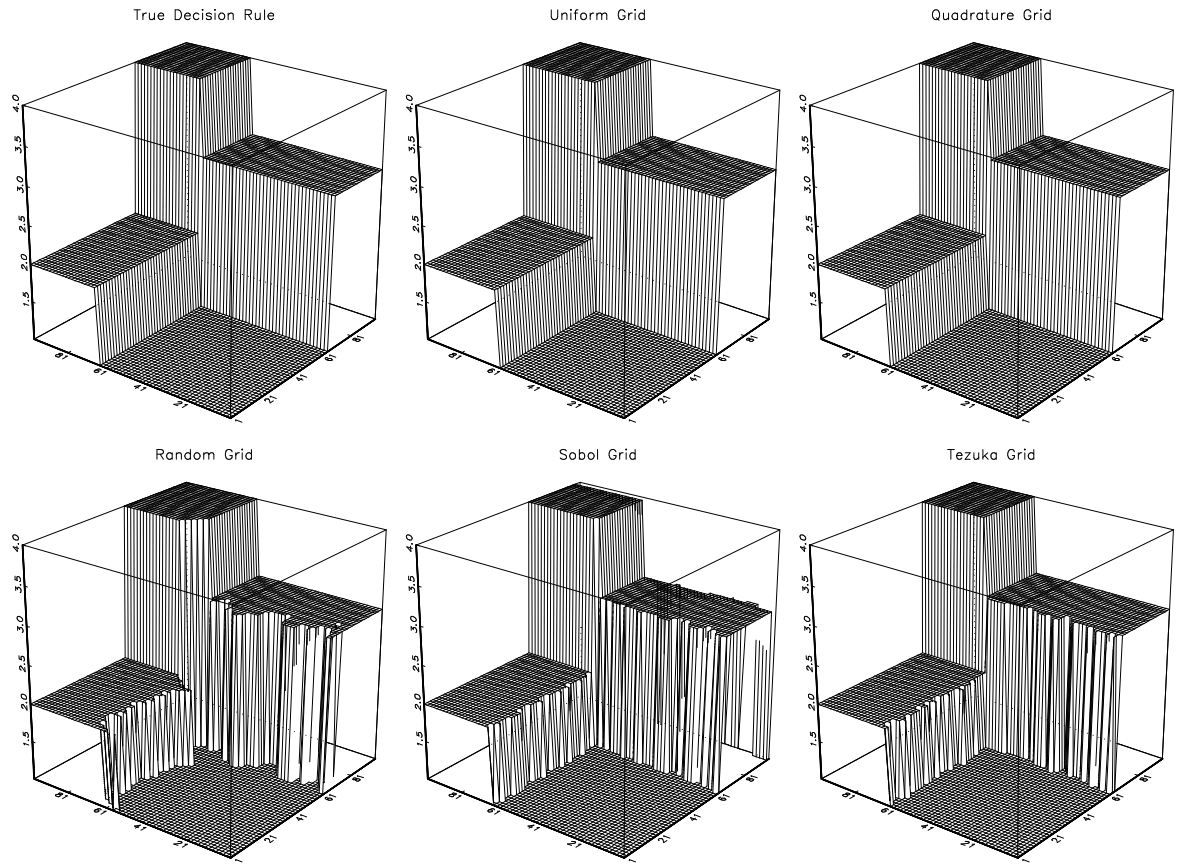

Figure 5.5 Decision Rules for 2-Dimensional Replacement Problem with $\lambda = .1$

Figure 5.6 presents value functions calculated for the 2-dimensional replacement problem in the case where there is a large discontinuity in the univariate transition density as described in section 4 ($\lambda = .8$). All solutions were also calculated for grids with $N = 2500$ points. Since I didn't do any smoothing of the discontinuous transition density, it is not surprising that these value functions are all rather more bumpy than the smoothed value function computed from the uniform grid. Other than their more bumpy appearance, these other value functions have the same general shape and magnitude. In particular all of the approximate solutions underestimate the true value function by about 60000, reflecting the same sort of "Jensen inequality bias" that we observed for the 1-dimensional approximations in section 4.

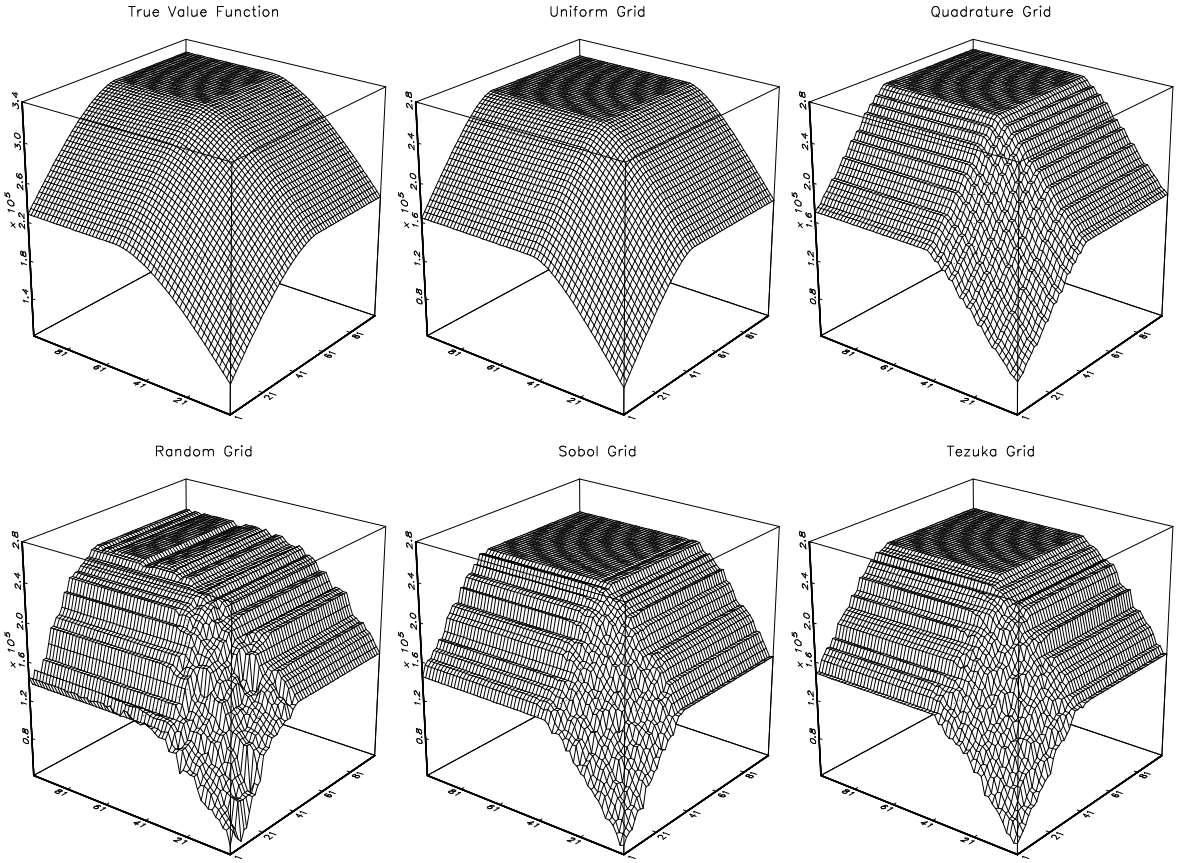

Figure 5.6 Value Functions for 2-Dimensional Replacement Problem with $\lambda = .8$

Figure 5.7 plots the corresponding decision rules for each of the value functions plotted in figure 5.6. The true decision rule is plotted in the upper left hand corner of figure 5.7. We see that the decision rules for the uniform and quadrature grids are remarkably similar in this case: in both cases the durable is replaced too soon so that the "keep" region of the state space is too small and the "replace" region is too large. The three bottom panels of figure 5.7 show that the decision rules computed from the random grids and low discrepancy grids are more irregular. In the case of the decision rules for the low discrepancy grids there are rough areas at the boundary between the different regions of the state space: these rough areas result from misclassification of the optimal decision rule due to errors in calculating the alternative specific value functions, $V_N(s, a)$ defined above. The fact that the random and low discrepancy grids lead to more jagged boundaries is another indication of the fact that there is more jaggedness in the alternative specific value functions for these methods. While each of the 4 alternative specific value functions is approximated with more error for random and low discrepancy grids than for uniform or quadrature grids, the overall value function — the minimum of the 4 alternative specific value function — is approximately the same for all 5 methods, although noticeably more bumpy for the RPI algorithm.

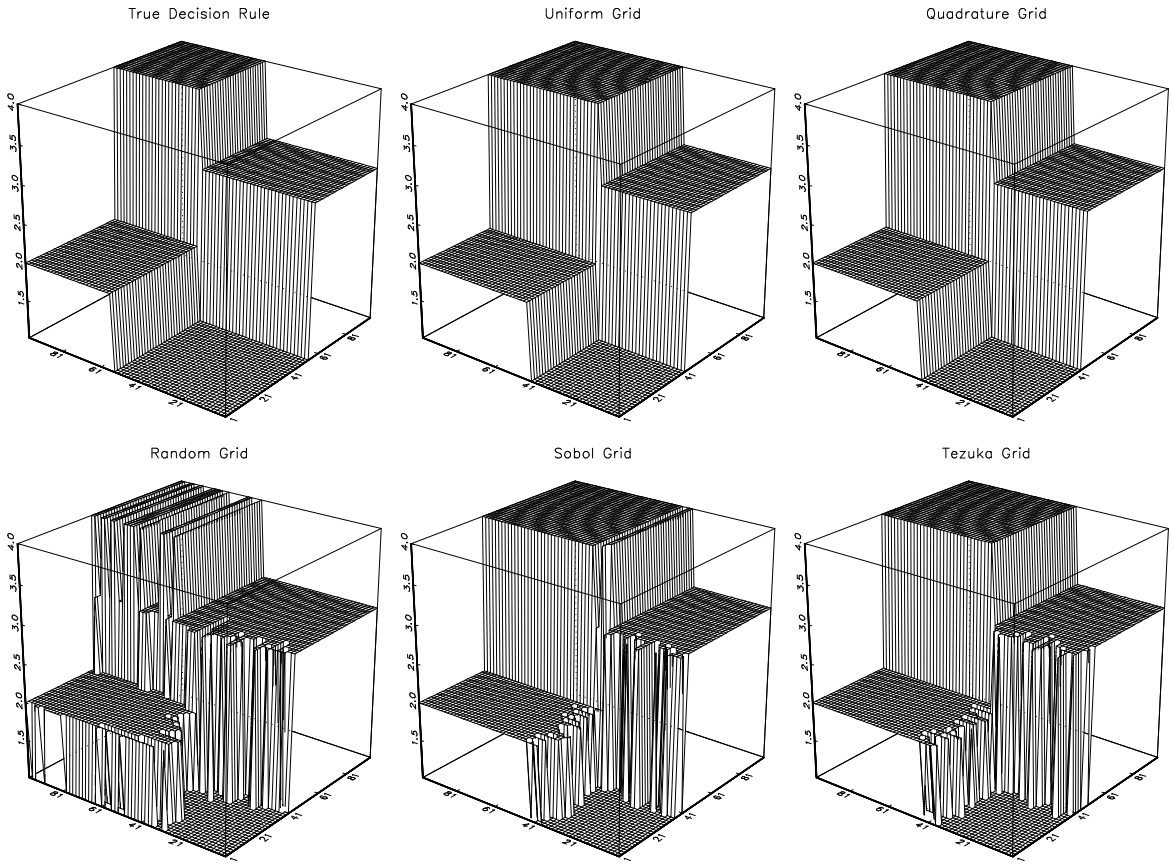

Figure 5.7 Decision Rules for 2-Dimensional Replacement Problem with $\lambda = .8$

Figure 5.8 plots the actual approximation error $\|V - V_N\|$ as a function of the number of grid points N for a sequence of uniform grids, quadrature grids, Sobol' grids, and random grids. The results for the random grids in the lower right panel of figure 5.8 are an average of $\|V - V_N\|$ for 50 independent draws of $\{s_1, \ldots, s_N\}$. The dashed line in this panel is the fitted regression line, $\exp(\hat{a} + \hat{b}\log(N))$ from a log-linear regression of the log of the actual average errors on $\log(N)$. We see that that the average errors appear to be converging at approximately $1/N^{.66}$ which is faster than the theoretically predicted rate of $1/N^{.5}$. In the case of the uniform and random grids, we see that the rate of convergence is approximately $1/N^{.5}$ just as predicted from the theoretical bounds derived in Theorem 2.2. The final panel of figure 5.8 presents the rate of convergence of $\|V - V_N\|$ for the Sobol' grid. The regression results show that the rate of convergence conforms to the predictions of Theorem 2.1: in particular, the rate of convergence is approximately $\log(N)^{1.6}/N$ which is slightly faster than the predicted upper bound of $\log(N)^2/N$ given in Theorem 2.1.

Figure 5.8 also presents the bounding constants for each of the methods. In each case the bounding constant in the two dimensional case, $\exp(14) = 2.6 \times 10^6$ is significantly higher than the bounding constant in the one-dimensional case, $\exp(12) = 1.6 \times 10^5$. The fact that this bounding constant appears to be growing exponentially

with the dimension of the problem shows that there is a curse of dimensionality for this problem despite the fact that randomization is being used. The curse of dimensionality arises from a failure of a key regularity condition in Rust (1997): that the transition density $p(s'|s,a)$ is Lipschitz continuous with a Lipschitz bound K_p that does not grow with the dimension of the problem. In this case the transition density is discontinuous, and the Lipschitz bound $K_p = \infty$. We have seen that the fact that the Lipschitz bound for $p(s'|s,a)$ is infinite does not prevent us from being able to approximate V and α using random grids. However the curse of dimensionality arises from the discontinuity in the transition density combined with the fact that the multidimensional transition density is a product of one-dimensional transition densities: this implies that even if smoothing is used to guarantee that K_p is finite, K_p will increase exponentially fast in the problem dimension d for this artificial test problem.

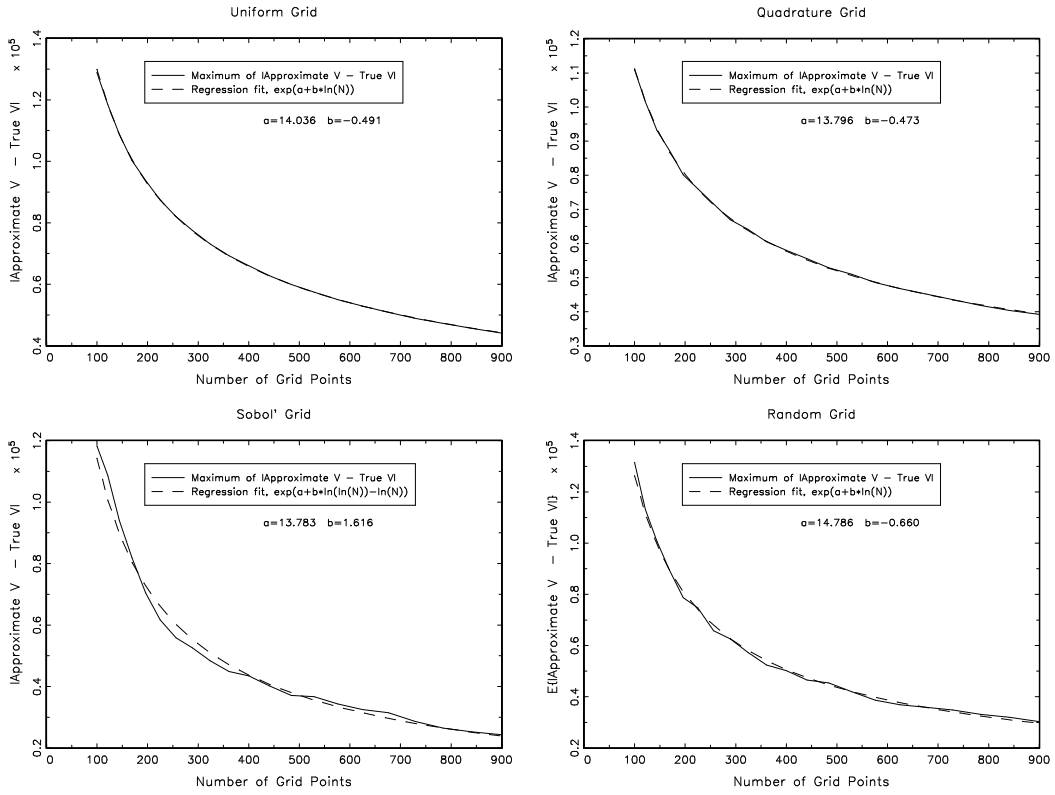

Figure 5.8 Predicted vs. Actual Convergence Rate for $\|V - V_N\|$ in the 2-Dimensional Replacement Problem

Figure 5.9 presents a summary of the convergence rates for $\|V - V_N\|$ and $\|\alpha - \alpha_N\|$ (where the latter is actually measured as a percentage deviation from α over a grid of points in S) for the replacement problem with low discontinuities ($\lambda = .1$) and high discontinuity ($\lambda = .8$). The upper two panels of figure 5.9 show the results for the low discontinuity case. Here we see that the fact that all of all the methods have similar bounding constants but that methods based on low discrepancy grids have a faster rate of convergence implies that when N sufficiently large the V_N's and α_N's computed from the low discrepancy grids will be more accurate than those calculated from the random, uniform or quadrature grids. Indeed, we see from the top left panel of figure 5.9 that for $N \geq 300$ the Tezuka grid yields the most accurate estimates of V and α. S The V_N's calculated from uniform grids were the least accurate, underestimating V by more than $20,000$ more than the V_N calculated using Tezuka grids. It is quite surprising that the ability to do exact integration conferred so little advantage to the uniform grid. However the increased smoothness resulting from exact integration did allow it to outperform the accuracy of decision rules calculated from random grids as we can see in the top right hand panel of figure 5.9.

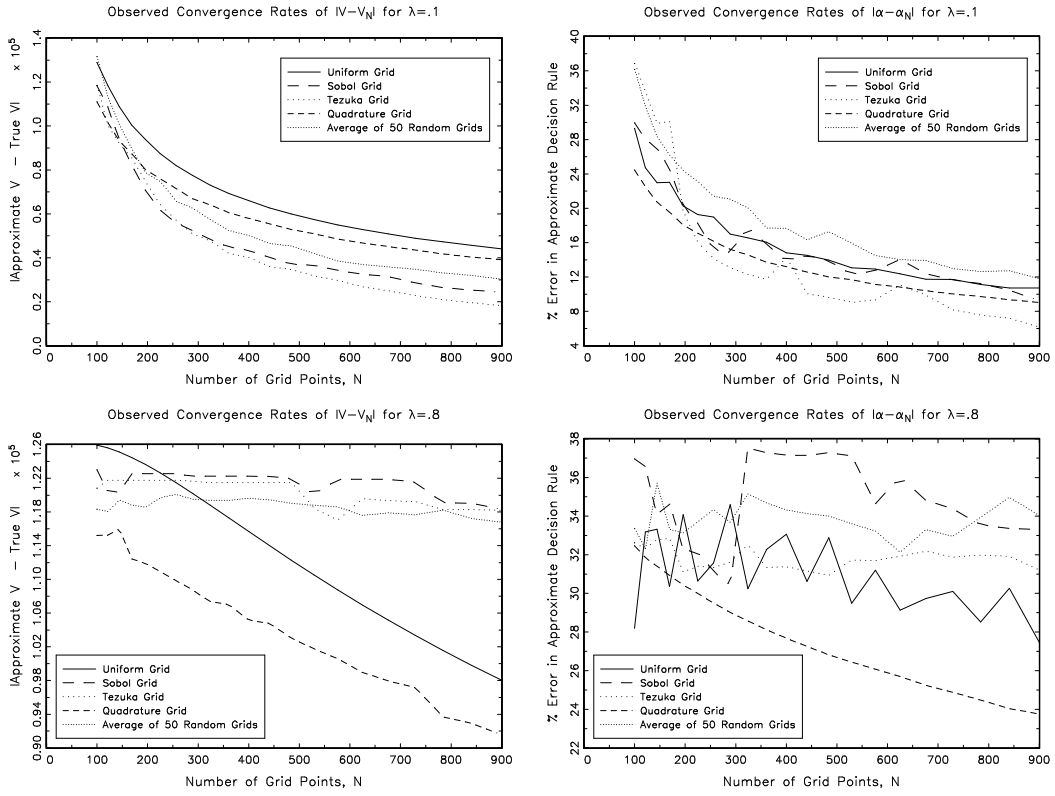

Figure 5.9 Convergence of Value Functions and Decision Rules in Replacement Problem with $\lambda = .1$ and $\lambda = .8$

The two bottom panels of figure 5.9 serve as a warning that these results can break down when the level of discontinuity in $p(s'|s,a)$ is sufficiently great. These panels plot the rate of convergence of $\|V - V_N\|$ and $\|\alpha - \alpha_N\|$ in the high discontinuity case when $\lambda = .8$. We see that in this case, the error bounds derived in section 2 no longer appear to correctly predict the rates of convergence of $\|V - V_N\|$ and $\|\alpha - \alpha_N\|$. The actual rates of convergence are much slower when $\lambda = .8$, even for the uniform and quadrature grids, where the error in $\|V - V_N\|$ is going to zero at approximately $1/N^{.1}$ instead of the predicted rate of $1/N^{.5}$ given in Theorem 2.2.

I conclude this section with figure 5.10 which gives some insight into the reasons behind the "breakdown" in convergence rates observed in the lower two panels of figure 5.9. Figure 5.10 shows that the transition density in the high discontinuity case is very close to a spike, which leads to the "needle in the haystack" problem described earlier. In particular, if the grid is not sufficiently dense, it will always be possible fo find a spike centered at s which is in a "gap in the grid". Since most of the probability mass in $p(s'|s,a)$ is located under this spike, an approximate value function at such a point will be very poorly estimated. We will therefore have very large errors in V_N until the grid gets sufficiently dense that it is impossible to find an $s \in S$ where a spike located at s will not cover at least a few points in the grid. Once this "density threshold" or critical value of N is attained, the rates of convergence will start to conform to those predicted in section 2. Figures 5.6 and 5.7 show that the random and low discrepancy methods are comparable with the uniform and quadrature grids when $N = 2500$, so that for $N \geq 2500$, the rates of convergence of these methods should conform to the bounds derived in section 2, although I have not verified this experimentally yet.

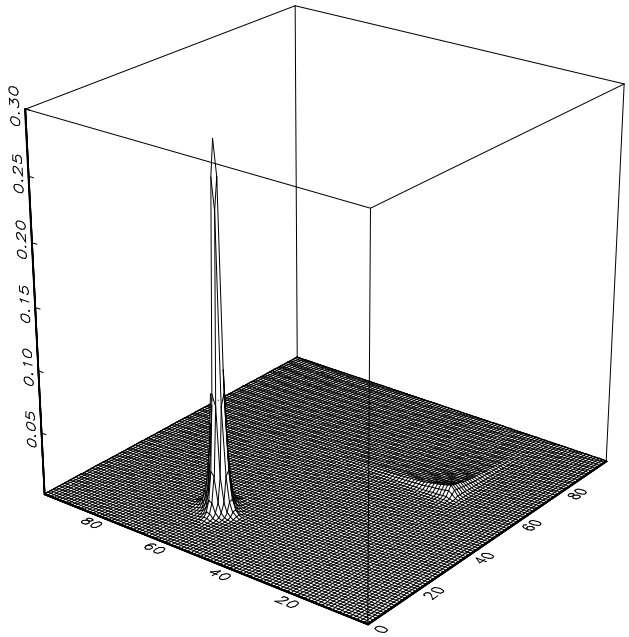

Figure 5.10 Transition Densities Associated with $(a_1 = 1, a_2 = 1)$ (keep both assets) for $\lambda = .1$ and $\lambda = .8$

Figure 5.10 shows that the transition density can be well approximated by a continuous transition density in the low discontinuity case, $\lambda = .1$, so it is not surprising that the convergence bounds in section 2 apply when $\lambda = .1$ but not in the high discontinuity case when $\lambda = .8$. Note that the results in figure 5.9 were computed using no smoothing, $\sigma = 0$. I have verified that when I use sufficient smoothing, the convergence rates conform to the bound presented in section 2, although I do not present these results here to keep this already long paper within bounds. However while increasing the amount of smoothing "solves" the convergence problem, as noted in section 4 smoothing also creates a bias problem. At this point I have no good advice to offer as to how to set the level of smoothing in an MDP problem where there are discontinuities in the transition density.

It may seem a bit ironic that computation becomes so difficult when $\lambda \to 0$, since in the limit where $\lambda = 0$ we have a problem where the durable good never ages and always remains in its current condition s. In that case it is trivial to verify that V is given by

$$V(s) = \begin{cases} \dfrac{c(s)}{(1-\beta)} & \text{if} s < \gamma \\ \overline{P} - \underline{P} + \dfrac{c(0)}{(1-\beta)} & \text{if} s \geq \gamma \end{cases} \tag{5.2}$$

where γ is the solution to

$$\frac{c(\gamma)}{(1-\beta)} = \overline{P} - \underline{P} + \frac{c(0)}{(1-\beta)}. \tag{5.3}$$

As $\lambda \to 0$ the MDP problem is converging to a deterministic problem which should be "easier" to solve than the stochastic problem. Indded, in this case the limiting deterministic problem is in fact trivial to solve. Further thought needs to be devoted to resolving this paradox.

6. Conclusion

This paper has compared policy iteration algorithms for solving continuous-state, infinite-horizon MDP problems using alternative discretizations of the state space. Specifically, I compared the performance of 4 different strategies for constructing an embedded finite state MDP that approximates the original continuous state MDP problem. The four strategies differ primarily in the way they discretize the state space, yielding four different types of *grids:* 1) uniform grids, 2) quadrature grids, 3) random grids, and 4) low discrepancy grids. I evaluated these methods using two test problems with a finite number of actions in order eliminate the "optimization error" that would be associated with continuous action MDP problem. I used policy iteration to eliminate the "fixed point error" associated with solving the Bellman equation for the embedded finite state MDP problem. This allowed me to focus on the magnitude of the approximation error due to "integration error" and show how it decreases as a function of the number of grid points N. I found that integration error can be substantial, so that in practical work I would not recommend use of policy iteration since cpu time for this method increases at rate $O(N^3)$ which makes this method impractical when N is greater than several thousand. Instead I recommend use of multigrid methods described in Rust (1996) since their cpu time increases at rate $O(N^2)$, and so will be practical with N is as large as several hundred thousand.

In section 2 I derived bounds on the rate of convergence of each of these methods for a fixed MDP problem. I showed that for uniform grids and quadrature grids the approximation error $\|V - V_N\|$ (where V is the true value function and V_N is an approximate value function computed from a grid containing N points) is bounded by $K_u(d)/N^{1/d}$. The numerical results show that actual rates of convergence are well described by this bound, so methods based on uniform and quadrature grids are subject to an inherent curse of dimensionality. I showed that random grids have rates of convergence that decrease at rate $K_r(d)/\sqrt{N}$, and low discrepancy grids have rates of convergence that decrease at rate $K_l(d)\log(N)^d/N$, where $K_r(d)$ and $K_l(d)$ are bounding constants that may depend on the dimension of the problem. It follows that random and low discrepancy grids break the curse of dimensionality *for fixed MDP problems provided the bounding constants $K_r(d)$ and $K_l(d)$ do not increase exponentially fast in d.* Unfortunately, it is difficult to evaluate the bounding constants K_u and K_l in practice, and K_l is especially difficult to evaluate since it is based on the concept of *variation* of a function, $\mathcal{V}(f)$, and it is typically difficult to derive analytical expressions or even tight upper bounds for this quantity. Fortunately, in the cases considered in this paper the bounding constants K_l and K_u were close to the bounding constant K_r for the random policy iteration algorithm which is relatively easy to evaluate (see inequality (2.20) in section 2.6).

In sections 4 and 5 I compared the actual performance of the four algorithms listed above in two test problems that have analytic solutions: a job search problem and a replacement problem. I found that the methods based on low discrepancy grids substantially outperformed the random policy iteration (RPI) algorithm using "random grids". The RPI algorithm in turned outperformed the deterministic policy iteration methods based on uniform and quadrature grids

in even the one-dimensional test problems provided the transition density for the problem was sufficiently "smooth". However the RPI algorithm can be inferior to the latter methods in problems where the transition density has large discontinuities or "spikes" due to the "needle in the haystack problem" that I described in section 5.

Overall, I found it remarkable that the methods performed as well as they did given that both of the test problems are "irregular" in the sense that the transition density for the job search problem does not exist, and the transition density in the replacement problem is discontinuous. I was quite surprised that the uniform grid did not yield substantially better results than the other methods given that it had a substantial advantage in exploiting the special features of the test problems to exactly integrate each of the partition elements induced by the uniform grid. The results for the uniform grid were quite close to the results for the quadrature grid which did not exploit the (unrealistic) special advantage of exact integration. The overall performance of the quadrature grid approach was quite impressive: it yielded very smooth and well-behaved approximate value functions V_N and decision rules α_N. However the curse of dimensionality implies that low discrepancy grids will generally outperform quadrature grids in problems where $d \geq 2$. This conclusion is supported by the numerical results in the test problems provided the discontinuities in the transition density were not too large. However when $p(s'|s,a)$ has large spikes, quadrature grids outperformed the low discrepancy grids for sufficiently small values of N, but eventually once N was sufficiently large to guarantee a "critical density" of points in the grid to avoid the needle in the haystack problem, the low discrepancy grids seemed to outperform quadrature grids since they have a faster rate of convergence when $d \geq 2$.

My results demonstrate the irregularities in the transition density are not "fatal" in the sense that that we can still use the algorithms in this paper to solve irregular problems by introducing a small amount of smoothing to make the problems regular. However any amount of smoothing distorts the true solution to the problem, creating an analog of the "bias vs. variance" tradeoff in nonparametric statistics. Unfortunately at this point I cannot offer any general guidance on how to choose the appropriate level of smoothing to optimally balance the bias and variance. I have shown that smoothing is essential in the job search problem where the transition density is a Dirac delta function. However smoothing did not appear to help a great deal in the replacement problem: the increased bias due to the smoothing appeared to outweigh the reduction in variance. However I admit that so far I have considered relatively naive forms of smoothing that didn't attempt to adjust for distortions in the smoothed law of motion: there could be substantial gains to using more sophisticated forms of smoothing that preserve the "first order properties" of the law of motion (e.g. the conditional expectation for the smoothed law of motion could be restricted to equal the conditional expectation of the original law of motion, $p(s'|s,a)$). There are also likely to be big gains to finding tractable expressions for a bias correction term, not only to correct for bias introduced by smoothing, but also to correct the "Jensen inequality bias" that seems to be responsible for the fact that all of the methods systematically underestimated the true value function in the replacement problem. However one must approach this latter area with some caution: my numerical results

show there was no similar "Jensen's inequality bias" in the job search problem, so before one tries to derive general bias correction formulas we need to understand why the bias is present in one problem but not the other.

The computational results in section 5 do suggest that lack of smoothness in the transition density can be a much more serious problem in higher dimensional problems, where it can cause the curse of dimensionality to reappear for all of the methods considered in this paper. This was well illustrated in my artificial sequence of multidimensional test problems which were created via the convolution procedure suggested by Ken Judd. The fact that the multidimensional transition density is a product of unidimensional densities is the underlying cause of the curse of dimensionality: this product structure implies that the bounding constants $K_u(d)$, $K_r(d)$, and $K_l(d)$ all appear to increase exponentially fast in d. However the bounding constants for all the methods were very close in magnitude for the test problems I considered. This implies that even though there is an inherent curse of dimensionality for this artificial suite of test problems, the low discrepancy methods still have an advantage over the other methods since their rates of convergence exceed the rates of convergence of the other problems in problems where $d \geq 2$. My results show that low discrepancy methods can even be superior in one dimensional problems, despite the fact that uniform grids are theoretically optimal in this case with a faster rate of convergence of $1/N$.

I conclude that the numerical results in this paper suggest that solving continuous MDPs using low discrepancy discretizations suggested by Rust (1996) are indeed an effective and promising method, even in relatively low dimensional problems and for some highly irregular test problems where I would not have expected these methods to work especially well *a priori.* However the results in section 5 do suggest the need for caution in applying these methods in multidimensional MDP problems where there may be irregularities in the transition density particularly in the form of discontinuities or "spikes". The results in this paper show that the "needle in the haystack" problem may necessitate a relatively high density discretization, i.e. one may need a relatively large number N of grid points to guarantee acceptable accuracy. However it is relatively reassuring that even in the highly irregular problems studied in this paper, results of acceptable accuracy could be obtained using thousands rather than hundreds of thousands or even millions of grid points.

7. References

Anderson, E. Hansen, L. McGratten, E. and T. Sargent (1996) "Mechanics for Forming and Estimating Dynamic Linear Economies" in H. Amman, D. Kendrick and J. Rust (eds.) *Handbook of Computational Economics* Amsterdam, Elsevier, Chapter 4, 171–252.

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

Bellman, R. and S. Dreyfus (1962) *Applied Dynamic Programming* Princeton University Press.

Bellman, R. Kalaba, R. and B. Kotkin (1963) "Polynomial Approximation: A New Technique in Dynamic Programming Allocation Processes" *Mathematics of Computation* **17** 155–161.

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

Bertsekas, D.P. (1995) *Dynamic Programming and Optimal Control* (volume 2) Athena Scientific, Belmont, MA.

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

Bouleau, N. and D. Lépingle (1994) *Numerical Methods for Stochastic Processes* Wiley, New York.

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.

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

Davis, P. and P. Rabinowitz (1984) *Methods of Numerical Integration* Academic Press, Orlando.

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

Fox, B.L. (1973) "Discretizing Dynamic Programming" *J. Opt. Theor. Appl.* **11** 228-234.

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

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

Judd, K. (1996) "Approximation, Perturbation, and Projection Methods in Economic Analysis" in *Handbook of Computational Economics* ed. by H. Amman, D. Kendrick and J. Rust, Amsterdam, Elsevier-North Holland, Chapter 12, 511–585.

Keane, M. P. Wolpin, K. I. (1994) "The Solution and Estimation of Discrete Choice Dynamic Programming Models by Simulation and Interpolation: Monte Carlo Evidence" *Review of Economics and Statistics* **76-4** 648–672.

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.

Niederreiter, H. (1992) *Random Number Generation and Quasi-Monte Carlo Methods* **63** SIAM CBMS-NSF Conference Series in Applied Mathematics, Philadelphia.

Pakes, A. and P. McGuire (1996) "Stochastic Algorithms for Dynamic Models: Markov Perfect Equilibrium and the 'Curse of Dimensionality"' manuscript, Department of Economics, Yale University.

Papgeorgiou, A. and J. Traub (1996) "Beating Monte Carlo" *Risk*

Paskov, S. (1996) "New Methodologies for Valuing Securities" S. Pliska and M. Dempster (eds.) Isaac Newton Institute, Cambridge, England.

Paskov, S. and J.F. Traub (1996) "Faster Evaluation of Financial Derivatives" forthcoming, *Journal of Portfolio Management.*

Pollard, D. (1989) "Asymptotics via Empirical Processes" *Statistical Science* **4-4** 341–386.

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.

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

Puterman, M. (1994) *Markov Decision Processes* Wiley, New York.

Puterman, M.L. and Brumelle, S. (1979) "On the Convergence of Policy Iteration in Stationary Dynamic Programming" *Mathematics of Operations Research* **4** 60–69.

Press, W.H., Teukolsky, S.A., Vetterling W.T., and B.P. Flannery (1992) *Numerical Recipes* Cambridge University Press.

Roth, K.F. (1954) "On Irregularities of Distribution" *Mathematika* **1** 73–79.

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. (1994) "Structural Estimation of Markov Decision Processes" chapter 51 in D. McFadden and R. Engle (eds.) *Handbook of Econometrics* **4** North Holland, 3082–3139.

Rust, J. (1996) "Numerical Dynamic Programming in Economics" in H. Amman, D. Kendrick, and J. Rust (eds.) *Handbook of Computational Economics* Elsevier-North Holland, Amsterdam, Chapter 14, 619–729.

Rust, J. (1997) "Using Randomization to Break the Curse of Dimensionality" forthcoming, *Econometrica.*

Santos, M. and J. Vigo (1996) "Analysis of Error for a Numerical Programming Algorithm Applied to Economic Models" manuscript, ITAM, Mexico.

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.

Tezuka, S. (1995) *Uniform Random Numbers: Theory and Practice* Kluwer, Boston.

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

Traub, J.F. and Woźniakowski, H. (1992) "The Monte Carlo Algorithm with a Psuedorandom Generator" *Mathematics of Computation* **58-197** 323–339.

Tsitsiklis, J.N. (1994) “Asynchronous Stochastic Approximation and Q-Learning” *Machine Learning* **16** 185–202.

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

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