Numerical Dynamic Programming in Economics

Handbook of Computational Economics

H. Amman, D. Kendrick and J. Rust *(eds.)*

1. Introduction
2. Review of MDP's and the Theory of Dynamic Programming
   - Definitions of MDP's, DDP's and CDP's
   - Bellman's Equation, Contractions, Blackwell's Theorem
   - Analytic Solutions to "Test Problems"
   - Euler Equations and Euler Operators
3. Computational Complexity and Optimal Algorithms
   - Discrete Computational Complexity
   - Continuous Computational Complexity
4. Numerical Methods for MDP's
   - Discrete Finite Horizon
   - Discrete Infinite Horizon
   - Continuous Finite Horizon
   - Continuous Infinite Horizon
5. Conclusions

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

- A *time index* $t \in \{0, 1, 2, \ldots, T\},\ T \leq \infty$
- A *state space* S
- An *action space* A
- A family of *constraint sets* $\{A(s) \subseteq A\}$
- A *transition probability* $p(ds'|s, a)$.
- A *discount factor* $\beta \in (0, 1)$.
- A *single period utility function* $u(s, a)$

$$V_0(s) = \max_{\alpha=(\alpha_0,\ldots,\alpha_T)} E_\alpha \left\{ \sum_{t=0}^{T} \beta^t u(s_t, a_t) | s_0 = s \right\}. \qquad (2.1)$$

Discrete MDP's vs. Continuous MDP's

Definition 2.2: *A Discrete Decision Process (DDP) is an MDP with the following property:*

- There is a finite set A such that $A(s) \subset A$ for each $s \in S$.

Definition 2.3: *A Continuous Decision Process (CDP) is an MDP with the following property:*

- For each $s \in S$, the action set $A(s)$ is a compact subset of R^{d_a} with non-empty interior.

Comment 1: Important distinction between CDP's and DDP's:

- Solutions to DDP's can be computed exactly (provided state space is finite), CDP's can only be approximated.
- CDP's are subject to an inherent *curse of dimensionality* but can break the curse of dimensionality for DDP's.
- Approximate solution methods might also be attractive for very large DDP problems even though exact solution methods exist.

Comment 2: The CDP/DDP dichotomy is mirrored by two fundamental approaches to approximation of solutions to MDP's:

- *Discrete Approximation Methods*
- *Continuous Approximation Methods*

Comment 3: There is substantial controversy in the economics literature over the "best" approximation method.

- The proliferation of different algorithms and controverseries over which approach is "best" is similar to controveries that have arise in statistics and econometrics about which estimations methods are "best". The computational problem is akin to nonparametric econometrics: which is "best" way to "estimate" (i.e. compute) unknown functions V, α?
- This chapter appeals to *computational complexity theory* to help shed new light on these controversies.
- Complexity theory is in principle capable of characterizing *optimal algorithms* for solving MDP's.

Computational Complexity and Optimal Algorithms

optimal strategies for finding optimal strategies.

- *Discrete Computational Complexity*

  Turing model of computation, exact solutions to discrete, finite-dimensional problems (matrix mult, LP, etc).

  $comp(n)$ minimal computer time to solve a problem of size n. Example, matrix multiplication $comp(n) = O(n^{2.376})$.

  Outstanding problems: $P = NP$ and $P = NC$ problems.

- *Continuous Computational Complexity*

  Real model of computation, ϵ-approximations to continuous infinite-dimensional problems (e.g. PDE's, integration, optimization, etc.)

  $comp(\epsilon, d)$ minimal time to compute an ϵ-approximation of a d-dimensional problem.

  Tight upper and lower bounds on complexity of many problems have been established.

  Enables us to formalize the *curse of dimensionality:*

Definition: *A class of discrete MDP problems with d_s state variables and d_a control variables is subject to the curse of dimensionality if $comp(d_s, d_a) = \Omega(2^{d_s+d_a})$. A class of continuous MDP problems is subject to the curse of dimensionality if $comp(\epsilon, d_a, d_a) = \Omega(1/\epsilon^{(d_s+d_a)})$.*

Dynamic Programming Solution, Finite Horizon Problems

$$\alpha_t(s_t) = \underset{a_t \in A(s_t)}{argmax}[u(s_t, a_t) + \beta \int V_{t+1}(s_{t+1}) p(ds_{t+1}|s_t, a_t)], \quad (2.5)$$

$$V_t(s_t) = \underset{a_t \in A(s_t)}{max}[u(s_t, a_t) + \beta \int V_{t+1}(s_{t+1}) p(ds_{t+1}|s_t, a_t)]. \quad (2.6)$$

Dynamic Programming Solution, Infinite Horizon Problems

$$\alpha(s) = \underset{a \in A(s)}{argmax}[u(s, a) + \beta \int V(s') p(ds'|s, a)], \quad (2.8)$$

$$V(s) = \underset{a \in A(s)}{max}[u(s, a) + \beta \int V(s') p(ds'|s, a)]. \quad (2.9)$$

Bellman's equation in operator form:

$$V = \Gamma(V) \quad (2.11)$$

where $\Gamma : B \to B$ is the *Bellman operator* defined by

$$\Gamma(V)(s) = \underset{a \in A}{max}\left[u(s, a) + \beta \int V(s') p(s'|s) \lambda(ds')\right] \quad (2.10)$$

Contraction Mapping Theorem: Γ *is a contraction mapping with unique fixed point* $V \in B$.

$$\|\Gamma(V) - \Gamma(W)\| \leq \beta \|V - W\| \quad (2.12)$$

Lemma 2.1: *Suppose* $\{\Gamma_N\}$ *converge pointwise to* Γ*: i.e.* $\forall W \in B$ *we have*

$$\underset{N \to \infty}{lim} \Gamma_N(W) = \Gamma(W). \quad (2.19)$$

Then the approximate fixed point $V_N = \Gamma_N(V_N)$ *satisfies the error bound:*

$$\|V_N - V\| \leq \frac{\|\Gamma_N(V) - \Gamma(V)\|}{(1 - \beta)}. \quad (2.20)$$

Lemma 2.2: *Let Γ be a contraction mapping with fixed point $V = \Gamma(V)$. For each W the following inequalities hold:*

$$\|W - \Gamma(W)\| \leq (1+\beta)\|V - W\| \tag{2.22}$$

$$\|\Gamma^t(W) - V\| \leq \beta^t \|\Gamma(W) - W\|/(1-\beta) \tag{2.23}$$

$$\|\Gamma^{t+1}(W) - V\| \leq \beta \|\Gamma^t(W) - V\|. \tag{2.24}$$

Lemma 2.3: *Let Γ be a Bellman operator. Then we have:*

$$\Gamma : B\left(0, \frac{K}{(1-\beta)}\right) \rightarrow B\left(0, \frac{K}{(1-\beta)}\right), \tag{2.25}$$

where $B(0,r) = \{V \in B| \ \|V\| \leq r\}$ and K is given by:

$$K \equiv \underset{s \in S}{sup} \ \underset{a \in A(s)}{sup} \ |u(s,a)|. \tag{2.26}$$

Infinite Horizon MDP's as Generalized Geometric Series

$$\begin{aligned} V_\alpha &= [I - \beta M_\alpha]^{-1} u_\alpha \\ &= u_\alpha + \beta M_\alpha u_\alpha + \beta^2 M_\alpha^2 u_\alpha + \beta^3 M_\alpha^3 u_\alpha + \cdots \end{aligned} \tag{2.29}$$

where M_α is the *Markov operator* defined by:

$$M_\alpha(V)(s) = \int V(s')p(ds'|s, \alpha(s)) \tag{2.30}$$

Solution Algorithms for Contraction Mappings

- *Successive Approximations*

$$V_k = \Gamma(V_{k-1}) = \Gamma^k(V_0). \tag{4.3}$$

- *Newton-Kantorovich iterations*

$$V_{k+1} = V_k - [I - \Gamma'(V_k)]^{-1}(I - \Gamma)(V_k), \tag{4.5}$$

Fundamental Tradeoffs:

1. SA is linearly convergent, NK is quadratically convergent.
2. SA is cheap per iteration, NK is expensive per iteration.
3. SA is globally convergent, NK is locally convergent.

Example: smoothed Bellman operator, Γ_σ

$$\Gamma_\sigma(V)(s) = \sigma\, log \left[\sum_{a \in A(s)} exp \left\{ \frac{1}{\sigma} \left[u(s,a) + \beta \int V(s')p(ds'|s,a) \right] \right\} \right]$$

$$\Gamma'_\sigma(V)(W)(s) = \beta \sum_{a \in A(s)} \int W(s')p(ds'|s,a)P(a|s),$$

where $P(a|s)$ is the *conditional choice probability*

$$P(a|s) = \frac{exp\left\{\frac{1}{\sigma}\left[u(s,a) + \beta \int V(s')p(ds'|s,a)\right]\right\}}{\sum_{a' \in A(s)} exp\left\{\frac{1}{\sigma}\left[u(s,a') + \beta \int V(s')p(ds'|s,a')\right]\right\}}$$

Solution Algorithms for Discrete MDP's

- *General iterative methods* Gauss-Seidel, Successive Overrelaxation, etc. from literature on nonlinear equations

- *Successive Approximations*

- *Accelerated Successive Approximations*

$$\Gamma^k(V) + \underline{b}_k e \leq V \leq \Gamma^k(V) + \overline{b}_k e, \tag{5.4}$$

$$\begin{aligned} \underline{b}_k &= \beta/(1-\beta)\, min[\Gamma^k(V) - \Gamma^{k-1}(V)] \\ \overline{b}_k &= \beta/(1-\beta)\, max[\Gamma^k(V) - \Gamma^{k-1}(V)]. \end{aligned} \tag{5.5}$$

- *Linear Programming (with Constrain Generation)*

$$\min_{V \in R^{|S|}} \sum_{s=1}^{|S|} V(s)$$

subject to:

$$V(s) \geq u(s,a) + \beta \sum_{s'=1}^{|S|} V(s')p(s'|s,a)$$

- *Policy iteration*

$$\alpha_t(s) = \underset{a \in A(s)}{argmax}[u(s,a) + \beta \sum_{s'=1}^{|S|} V_{\alpha_{t-1}}(s')p(s'|s,a)]. \tag{5.6}$$

$$V_{\alpha_t} = [I - \beta M_{\alpha_t}]^{-1} u_{\alpha_t} \tag{5.7}$$

- *Modified Policy Iteration* solve (5.7) approximately using successive approximations:

$$V_{\alpha_t} \simeq \left[\sum_{i=0}^{k} \beta^i M_{\alpha_t}^i\right] u_{\alpha_t}$$

- *Policy Iteration Using State Aggregation Methods*
- *Massively Parallel Policy Iteration* Pan/Reif idea: use Newton's method to invert $L = [I - \beta M_\alpha]$.

$$F(X) = [I - LX] = 0$$

$$L_k = (2I - L_{k-1}M)L_{k-1}.$$

Theorem *Suppose that the initial estimate L_0 of L^{-1} satisfies $\|I - L_0 L\| < 1$. Then a massive parrallel processor with $O(|S|^\omega)$ processors ($\omega \leq 2.376$) can find an approximate solution L_k satisfying the convergence criterion $\|L_k - L^{-1}\| \leq 2^{-n^c}\|L_0\|$ in $O(\,log(|S|)^2)$ time.*

Conjecture: *Let $\epsilon > 0$ be an arbitrary solution tolerance between an approximate solution $\hat{V}$ and the true solution $V = \Gamma(V)$ to an $|S|$-state infinite horizon MDP problem. Then the massively parallel policy iteration algorithm using the Pan-Reif algorithm to approximately solve each policy valuation step generates an approximate solution $\hat{V}$ satisfying $\|\hat{V} - V\| < \epsilon$ in at most in $O(\,log(log(1/\epsilon))\,log(|S|)^2)$ time steps using a massively parallel processor with $O(|S|^\omega)$ processors.*

- *Policy Iteration Using Minimum Residual Algorithms*

$$u = LV \quad L = [I - \beta M] \tag{5.32}$$

$$\hat{V}_k = \sum_{i=1}^{k} c_i^* L^{i-1} u$$

$$(c_1^*, \ldots, c_k^*) = \underset{(c_1,\ldots,c_k)\in R^k}{argmin} \|u - \sum_{i=1}^{k} c_i L^{i-1} u\|_2.$$

Note that MR algorithm uses *Krylov information:*

$$I_k(L, u) = (u, Lu, L^2u, , \ldots, L^k u).$$

Theorem: *The MR algorithm using the Krylov information is almost optimal for solving system (5.32) for the class* $F = \{L|L = I - \beta M, \|M\| \leq 1\}$ *in sense that the value of* k_{MR} *is nearly equal the minimal level needed to get an* ϵ*-approxmation to solution* V*.*

$$k_{MR}(\epsilon, F) = min\left(|S|, \left\lfloor \frac{log(\epsilon)}{log(\beta)}(1-\delta)\right\rfloor + 1\right)$$

Observation: If we don't need extremely high precision, then if $k_{MR}(\epsilon, F) \leq |S|$, and if L is sparse with $O(|S|)$ nonzeros, then MR algorithm will find approximate solution $\hat{V}$ in $O(|S|)$ vs. $O(|S|^3)$ steps used by standard Gaussian elimination to solve linear system (5.32).

Fundamental trade-off: Use tailor-made algorithms that exploit special structure of a given problem or general purpose algorithms that don't take advantage of special structure of problem?

Approximate Solution Methods for Continuous MDP's

- *Discrete Approximation Methods* Approximate V via an expanding sequence of approximate finite-dimensional fixed points $V \simeq V_N = \Gamma_N(V_N)$ where $\Gamma_N : R^N \to R^N$ is defined over a finite grid $\{s_1, \ldots, s_N\}$ of points in S.

$$\Gamma_N(V)(s_j) = \max_{a \in A(s_j)} [u(s_j, a) + \frac{\beta}{N} \sum_{i=1}^{N} V(s_i) \hat{p}(s_i | s_j, a)]$$

- *Smooth Approximation Methods* Approximate V via an expanding sequence of finite dimensional manifolds in B, i.e. $V \simeq V_\theta = g_k(\theta)$ for some smooth mapping $g_k : R^k \to B$. Choose $\hat{\theta}$ so that $V_{\hat{\theta}}$ is "as close as possible" to the true fixed point V.

$$V_\theta(s) = \sum_{i=1}^{k} \theta_i p_i(s) \tag{1.3}$$

$$\sigma(\theta) \equiv \|V_\theta - \Gamma(V_\theta)\|_2 \equiv \sqrt{\int |V_\theta(s) - \Gamma(V_\theta)(s)|^2 \mu(ds)} \tag{1.4}$$

$$\sigma_N(\theta) \equiv \sqrt{\sum_{i=1}^{N} |V_\theta(s_i) - \hat{\Gamma}(V_\theta)(s_i)|^2}, \tag{1.5}$$

How to Choose Grid Points for DA methods?

- *Uniform Grid Points* (Chow and Tsitsiklis, 1991) Assume $S = [0,1]^{d_s}$ and $A(s) = [0,1]^{d_a} \equiv A$. Partition S and A into equal subcubes of length h on each side. Then have $N_s = (1/h)^{d_s}$ discrete states and $N_a = (1/h)^{d_a}$ discrete actions. Define discretized utility and transition probability by:

$$u_h(s, a) = u(s_{k(s)}, a)$$
$$p_h(s'|s, a) = \frac{p(s_{k(s')}|s_{k(s)}, a)}{\int p(s_{k(s')}|s_{k(s)}, a) ds'}$$

where $s_{k(s)}$ is an arbitrary "representative" from subcube $k(s)$ containing point $s \in S$. Define approximate Bellman operator Γ_h by:

$$\begin{aligned}\hat{\Gamma}_h(V)(s) &= \max_{a_k}[u_h(s,a_k) + \beta \int V(s')p_h(s'|s,a_k)ds'] \\ &= \max_{a_k}[u_h(s,a_k) + \frac{\beta}{N}\sum_{k=1}^{N} V(s_k)p_h(s_k|s_{k(s)},a_k)],\end{aligned}$$

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

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

Note this method is subject to "curse of dimensionality": number of operations to find ϵ-approximation to V is $O(T/((1-\beta)^2\epsilon)^{(2d_s+d_a)})$.

- *Quadrature Grids* (Tauchen and Hussey, 1991)

$$\begin{aligned}\int V(s')p(s'|s,a)ds' &= \int V(s')\frac{p(s'|s,a)}{p(s')}p(s')ds' \\ &\simeq \sum_{k=1}^{N} V(s_k)\frac{p(s_k|s,a)}{p(s_k)}p_k\end{aligned}.$$

The quadrature abscissa $\{s_1,\ldots,s_N\}$ then yield our grid, and the weights $\{p_1,\ldots,p_N\}$ can be used to define an N-state Markov chain with transition probability p_N given by:

$$p_N(s_k|s_j,a) = \frac{p(s_k|s_j,a)p_k/p(s_k)}{\Sigma_{i=1}^{N} p(s_i|s_j,a)p_i/p(s_i)}, \quad j,k = 1,\ldots,N.$$

Then the approximate Bellman operator is given by:

$$\hat{\Gamma}_N(V)(s) = \max_{a \in A_N(s)}[u(s,a) + \beta \sum_{k=1}^{N} V(s_k)p_N(s_k|s,a)]$$

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

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

Comment: Quadrature grids are also subject to the curse of dimensionality.

- *Random Grids* (Rust, 1994) Choose a random grid $\{\tilde{s}_1, \ldots, \tilde{s}_N\}$ via *IID* draws from $S = [0,1]^{d_s}$. Form *random Bellman operator* $\tilde{\Gamma}_N$ defined by:

$$\tilde{\Gamma}_N(V)(s) = \underset{a \in A(s)}{max}[u(s,a) + \frac{\beta}{N} \sum_{i=1}^{N} V(\tilde{s}_i) p(\tilde{s}_i|s,a)],$$

Theorem *Suppose $S = [0,1]^d$, A is a finite set, and (u,p) satisfy the Lipschitz conditions. Then for each $N \geq 1$ the expected error in the random Bellman operator satisfies the following uniform bound:*

$$\underset{p}{sup} \underset{\|V\| \leq K/(1-\beta)}{sup} E\left\{\|\tilde{\Gamma}_N(V) - \Gamma(V)\|\right\} \leq \frac{\gamma(d)|A|K_p K}{(1-\beta)\sqrt{N}}, \qquad (0.1)$$

where K_p is the Lipschitz bound on p and $\gamma(d)$ is a bounding constant given by:

$$\gamma(d) = \sqrt{\frac{\pi}{2}} \beta [1 + d\sqrt{\pi} C] \qquad (0.2)$$

and C is an absolute constant independent of p, V, β or d.

Comment: This result says that randomization succeeds in breaking the curse of dimensionality for DDP's. However randomization cannot break the curse of dimensionality of CDP's:

Theorem *Randomization cannot succeed in breaking the curse of dimensionality of the class of all continuous MDP problems, i.e. a lower bound on the computational complexity is given by:*

$$comp^{\textit{wor-ran}}(\epsilon, \beta, d_a, d_s) = \Omega\left(\frac{1}{[(1-\beta)^2\epsilon]^{d_a}}\right). \tag{0.3}$$

- *"Low Discrepancy" Grid Points* (Traub and Paskov, 1994, Judd, 1994). Use analog of random Bellman operator, but use deterministically chosen Hammersley, Halton or Sobol' points instead of random grid points from S.

- *Hybrid methods* (Keane and Wolpin, 1994). Based on alternative-specific value functions:

$$V_t(s,a) = u(s,a) + \beta \int \max_{a' \in A(s')} [V_{t+1}(s',a')] p(ds'|s,a),$$

$$V_t(s) = \max_{a \in A(s)} [V_t(s,a)].$$

$$\hat{V}_t(s_j,a) = u(s_j,a) + \frac{\beta}{N} \sum_{i=1}^{N} \max_{a' \in A(\tilde{s}_{ija})} [\hat{V}_{t+1}(\tilde{s}_{ija},a)],$$

Comment: Note that unlike previous methods, Keane-Wolpin method is not *self-interpolating,* i.e. $\hat{V}_t$ is only defined at the pre-assigned grid points $(s_1,\ldots,s_N)$ and not at other points $s \in S$. Since the realizations $(\tilde{s}_{1ja},\ldots,\tilde{s}_{Nja})$ will generally not fall on the pre-assigned grid points $(s_1,\ldots,s_N)$, Keane and Wolpin fit an interpolating regression using the N simulated values $(\hat{V}_t(s_1),\ldots,\hat{V}_t(s_N))$ as dependent variables in order to evaluate $\hat{V}_t(s)$ at arbitrary points $s \in S$. In principle many other interpolation procedures could be used such as simple linear interpolation or any of the multivariate interpolation procedures. Our formula assumes that this procedure has already been carried out on $\hat{V}_{t+1}$ so that it can be evaluated at all the random points $\tilde{s}_{ija}$, $i = 1,\ldots,N$.

Numerical solution of the auto replacement problem

- Consider the problem of optimal replacement of durable assets analyzed in Howard 1960 (discrete state) and Rust (1985,1986) (continuous state).
- State space $S = R_+$, where s_t is interpreted as a measure of the accumulated utilization of the durable (such as the odometer reading on a car). Thus $s_t = 0$ denotes a brand new durable good.
- 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 $\overline{P}$.
- Utilization of the asset each period has an exogenous exponential distribution. This corresponds to a transition probability p is given by:

$$p(ds_{t+1}|s_t, a_t) = \begin{cases} \lambda\, exp\{-\lambda(s_{t+1} - s_t)\} & \text{if } a_t = 0 \text{ and } s_{t+1} \geq s_t \\ \lambda\, exp\{-\lambda(s_{t+1} - 0)\} & \text{if } a_t = 1 \text{ and } s_{t+1} \geq 0 \\ 0 & \textit{otherwise.} \end{cases} \tag{0.4}$$

- The per-period cost of operating the asset in state s is given by a function $c(s)$. This implies the following per period utility function:

$$u(s_t, a_t) = \begin{cases} -\, c(s_t) & \text{if } a_t = 0 \\ -\, [\,\overline{P} - \underline{P}\,] - c(0) & \text{if } a_t = 1. \end{cases} \tag{0.5}$$

- Bellman's equation is given by:

$$V(s) = max[-c(s) + \beta \int_s^\infty V(s')\lambda\, exp\{-\lambda(s'-s)\}ds', \\ -[\overline{P} - \underline{P}] - c(0) + \beta \int_0^\infty V(s')\lambda\, exp\{-\lambda(s')\}ds'] \tag{0.6}$$

- Let γ be the smallest value of s such that the agent is indifferent between keeping and replacing. Differentiating Bellman's equation (0.6), it follows that on the *continuation region,* $[0, \gamma)$, $\alpha(s) = 0$ (i.e. keep the current durable) and V satisfies the differential equation:

$$V'(s) = -c'(s) + \lambda c(s) + \lambda(1-\beta)V(s). \tag{0.7}$$

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

$$V(\gamma) = [\,\overline{P} - \underline{P}\,] + V(0) = -c(\gamma) + \beta V(\gamma) = \frac{-c(\gamma)}{1-\beta}, \tag{0.8}$$

- Integrating the ODE, we obtain:

$$V(s) = max[\frac{-c(\gamma)}{1-\beta}, \frac{-c(\gamma)}{1-\beta} + \int_s^\gamma \frac{c'(y)}{1-\beta}[1 - \beta e^{-\lambda(1-\beta)(y-s)}]dy], \tag{0.9}$$

where γ is the unique solution to:

$$[\,\overline{P} - \underline{P}\,] = \int_0^\gamma \frac{c'(y)}{1-\beta}[1 - \beta e^{-\lambda(1-\beta)y}]dy. \tag{0.10}$$

Solving the auto replacment problem by discrete approximation.

- Formulate discrete-state version of the replacement problem using $|S| = 100$ states and the same discount factor, cost function, and replacement costs.
- Approximate the exponential transition density $p(\cdot|s_t, d_t)$ by a 12-point probability distribution using a simple continuity correction so that each of the 12 mass points equal the probability that an exponential random variable falls within plus or minus $.5$ of the integer values j that s_t assumes in the discrete case:

$$p(s_{t+1} = s_t + j|s_t, d_t) = \begin{cases} \int_0^{.5} \lambda\, exp^{-\lambda y}\, dy & j = 0 \\ \int_{j-.5}^{j+.5} \lambda\, exp^{-\lambda y}\, dy & j = 1, \ldots, 10 \\ \int_{10.5}^{\infty} \lambda\, exp^{-\lambda y}\, dy & j = 11. \end{cases} \tag{0.11}$$