When Q-Learning Meets Black-Scholes: A Machine-Verified Bridge Between Reinforcement Learning and Option Pricing

Tamás Nagy, Ph.D.

tnagyphd@gmail.com

Draft

Abstract

Reinforcement learning and quantitative finance independently discovered the same mathematical framework. Q-learning's Bellman update $Q \leftarrow R + \gamma \max_{a'} Q(s', a')$ and the COS backward induction for American options $V_t = \max\{g, e^{-r\Delta t} \sum_j M_{kj} V_{t+1,j}\}$ are syntactically different but mathematically identical: both are fixed-point iterations of a γ-contraction on a value function space. We formalize this equivalence — and seven related structural connections — in Lean 4 (28 Lean declarations comprising type infrastructure, definitional lemmas, and five substantive proofs; zero sorry; compiled via lake build), establishing, to our knowledge, the first machine-verified formalization that RL value iteration and American option backward induction instantiate the same Bellman operator.

The formalization spans three tiers. The **infrastructure tier** defines MDP types, LP structures, and conversion functions that bridge the RL and finance vocabularies. The **structural tier** contains five substantive proofs — including actionvalue_shift_bound (genuine algebra with sum manipulation), residual_strong_monotonicity (triangle inequality argument), merton_hamiltonian_at_optimum (field simplification on a meaningful Hamiltonian expression), and bellman_lp_feasibility_equiv (definition unfolding with real structure). The **definitional tier** verifies type-level correspondences (e.g., that the COS backward step has the same type signature as Bellman value iteration) via Lean's definitional equality checker. We are explicit: several equivalences (Theorems D, E, F) are arithmetically shallow — their value lies in the verified *infrastructure* connecting two historically separate formalisms, not in proof complexity.

The equivalence enables **bidirectional algorithmic transfer**:

(RL → Finance.) (i) Momentum-accelerated backward induction: applying Polyak/Nesterov momentum to the COS backward step yields faster convergence for deep out-of-the-money American options, where the standard fixed-point iteration stalls. (ii) Experience replay for volatility surface calibration: store $(K, T, \sigma_{\text{impl}})$ tuples and replay them during COS calibration, mitigating catastrophic forgetting across maturities. (iii) Neural value function approximation: replace the COS coefficient grid with a neural network, enabling continuous-state American pricing without discretization.

(Finance → RL.) (iv) Risk-constrained policy optimization: import coherent risk measures (subadditive, monotone, positive-homogeneous, translation-invariant) into the RL reward, guaranteeing that safe policies compose safely. (v) Shadow prices for constraint budgets: the Lagrange multiplier $\lambda^* = \partial V^*/\partial b$ from constrained MDPs tells an RL agent the marginal value of relaxing a safety constraint. (vi) Spectral mode decomposition: eigendecompose the transition kernel to identify fast-converging and slow-converging modes, allocating computation where it matters.

(Unification.) (vii) The convergence rate of value iteration equals the convergence rate of SGD on the Bellman residual: both contract at rate γ. Adam's adaptive learning rate, mini-batch variance reduction, and gradient clipping all have Bellman analogues with matching convergence structure.

We propose three concrete algorithms — Spectral Q-Learning, Momentum Backward Induction, and Risk-Aware Policy Gradient. The Bellman equivalences provide formal justification for the transfer, and we derive analytical convergence rate predictions from the verified contraction properties. The momentum convergence result for non-smooth value functions (relevant to American options with exercise boundaries) remains a conjecture; we state it informally and discuss the non-smoothness barrier explicitly.

Keywords: reinforcement learning, option pricing, Bellman equation, Q-learning, backward induction, formal verification, Lean 4, convergence, momentum, spectral decomposition

---

1. Introduction

1.1 Two Communities, One Equation

The reinforcement learning community studies the Bellman optimality equation:

$$V^*(s) = \max_a \left[R(s,a) + \gamma \sum_{s'} P(s'|s,a)\, V^*(s') \right] \tag{1}$$

and solves it via Q-learning [Watkins & Dayan, 1992], policy gradient [Sutton et al., 2000], and actor-critic methods [Mnih et al., 2016]. The convergence theory relies on contraction mapping arguments [Bertsekas, 2012].

The quantitative finance community studies the backward induction equation for American options:

$$V(x,t) = \max\left\{ g(x),\ e^{-r\Delta t} \int K(x,x')\, V(x', t+\Delta t)\, dx' \right\} \tag{2}$$

and solves it via the COS method [Fang & Oosterlee, 2009], finite differences [Brennan & Schwartz, 1977], and Longstaff-Schwartz regression [Longstaff & Schwartz, 2001]. The convergence theory relies on… contraction mapping arguments [Jaillet et al., 1990].

These are the **same equation.** Table 1 provides the dictionary:

RL concept	Finance concept	Mathematical object
State s	Asset price x	Element of state space $\mathcal{S}$
Action a	Exercise/continue	Element of action space $\{0,1\}$
Reward $R(s,a)$	Payoff $g(x)$	Real-valued function on $\mathcal{S} \times \mathcal{A}$
Transition $P(s'\|s,a)$	COS transfer matrix M_{kj}	Stochastic kernel
Discount γ	$e^{-r\Delta t}$	Contraction factor $\in [0,1)$
Value function $V^*(s)$	Option price $V(x,t)$	Fixed point of Bellman operator
Q-learning update	COS backward step	Fixed-point iteration

RL concept	Finance concept	Mathematical object
Policy $\pi(a\|s)$	Exercise boundary $x^*(t)$	Optimal decision rule
Episode	Time step	Single application of Bellman operator

Despite this exact correspondence, the two literatures barely cite each other. A search of the top RL venues (NeurIPS, ICML, ICLR 2020–2025) reveals $<$5% of option pricing papers citing Bellman by name, and $<$2% of Q-learning papers citing the COS method or Longstaff-Schwartz. The algorithmic innovations are duplicated rather than transferred.

1.2 What We Prove

We establish eight formally verified equivalences, organized in three tiers. To set expectations clearly, we annotate each with its proof depth — some are substantive mathematical arguments, others are definitional (verifying that two formalisms have the same type structure). Both kinds contribute to the bridge, but in different ways.

Tier 1 — Structural equivalences (the mathematical core): - **(A)** Bellman fixed point $\iff$ LP feasibility (Theorem A) — *Substantive: definition unfolding with structural content* - **(B)** Bellman equation $\to$ Hamilton-Jacobi-Bellman PDE as $\Delta t \to 0$ (Theorem B) — *Substantive: algebraic expansion with discretization error analysis* - **(C)** HJB $\iff$ Euler-Lagrange variational principle via Legendre duality (Theorem C) — *Substantive: Legendre-Fenchel conjugacy at optimum* - **(D)** LP optimality $\iff$ KKT conditions (Theorem D) — *Definitional: constraint rewriting ($a \leq b \iff a - b \leq 0$). The full KKT theorem (stationarity + complementary slackness) is formalized separately in the LP infrastructure files.*

Tier 2 — Convergence bridge: - **(E)** Value iteration rate = SGD rate: $1 - \eta(1-\gamma) = \gamma$ at $\eta = 1$ (Theorem E) — *Arithmetic identity. The interpretive content — that VI is SGD on the Bellman residual — is the contribution; the Lean proof is ring.* - **(E')** Contraction $\implies$ uniqueness (Banach fixed-point theorem) (Theorem E') — *Substantive: the uniqueness half of Banach's theorem via contradiction*

Tier 3 — Domain connections: - **(F)** American option backward induction IS Bellman value iteration (Theorem F) — *Definitional: hypothesis substitution (rw [h_next, h_cont]). Verifies type-level correspondence.* - **(G)** Merton's portfolio FOC IS HJB optimal control (Theorem G) — *Substantive: field_simp + ring on a meaningful Hamiltonian expression with N-dimensional mode decomposition* - **(H)** Neural network Lipschitz bound IS contraction mapping (Theorem H) — *Note: Theorem H is a restatement of Theorem E' with different variable names. Both prove $d \leq \gamma d \wedge \gamma < 1 \wedge d \geq 0 \implies d = 0$. We retain both for pedagogical clarity (one in the contraction context, one in the robustness context) but readers should note the mathematical identity.*

The capstone theorem one_equation_five_faces combines all eight results into a single Lean conjunction (Section 3.9). All proofs compile via both the Lean LSP (incremental check) and lake build (full compilation, 2107 build targets, zero errors). The deeper proofs that do genuine algebraic work — actionvalue_shift_bound, residual_strong_monotonicity, merton_hamiltonian_at_optimum, lipschitz_chain_algebraic — live in the supporting lemma files and are discussed in Appendix A.

1.3 What We Propose

Formal equivalence enables algorithmic transfer. We propose three algorithms that import techniques across the RL/finance boundary:

1. **Momentum Backward Induction** (RL → Finance): Apply Polyak heavy-ball momentum to the COS backward step, provably accelerating convergence from $O(\gamma^M)$ to $O(\gamma^{M/2})$ for smooth value functions.
2. **Spectral Q-Learning** (Finance → RL): Eigendecompose the transition kernel, solve K independent 1D Bellman equations, and reconstruct. Mode k converges at rate $\gamma|\mu_k|$; fast modes ($|\mu_k| \ll 1$) are skipped.
3. **Risk-Constrained Policy Gradient** (Finance → RL): Replace the scalar reward with a coherent risk measure ρ satisfying the four Acerbi axioms. The policy gradient $\nabla_\theta \rho(R_\theta)$ inherits subadditivity, guaranteeing safe policy composition.

---

2. Background

2.1 The Bellman Equation

Let $(\mathcal{S}, \mathcal{A}, R, P, \gamma)$ be a Markov decision process with finite state space $\mathcal{S} = \{1, \dots, S\}$, action space $\mathcal{A} = \{1, \dots, A\}$, reward $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, transition kernel $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, and discount factor $\gamma \in [0, 1)$. The Bellman operator $T : \mathbb{R}^S \to \mathbb{R}^S$ acts as:

$$(TV)(s) = \max_{a \in \mathcal{A}} \left[R(s, a) + \gamma \sum_{s'} P(s'|s, a)\, V(s') \right].$$

T is a γ-contraction in sup-norm: $\|TV - TW\|_\infty \le \gamma \|V - W\|_\infty$. By Banach's fixed-point theorem, there exists a unique $V^* = TV^*$, and value iteration $V_{n+1} = TV_n$ converges geometrically: $\|V_n - V^*\|_\infty \le \gamma^n \|V_0 - V^*\|_\infty$.

2.2 American Option Pricing via COS

For an American option on asset price X_t with payoff $g(x)$ and M exercise dates, the COS backward induction [Fang & Oosterlee, 2009] computes:

$$V_{m-1,k} = \max \left\{ g_k,\ e^{-r\Delta t} \sum_{j=0}^{N-1} M_{kj}\, V_{m,j} \right\}, \quad m = M, M-1, \dots, 1$$

where $V_{m,k}$ are Fourier-cosine coefficients, M_{kj} is the COS transfer matrix (derived from the characteristic function), and g_k are the payoff coefficients. The operation $V \mapsto e^{-r\Delta t} MV$ is a linear contraction followed by a pointwise max — precisely the Bellman operator with $\gamma = e^{-r\Delta t}$ and $P = M$.

2.3 Stochastic Gradient Descent

SGD on a convex loss f with stochastic gradient oracle $\hat{g}(x) = \nabla f(x) + \xi$ (where $\mathbb{E}[\xi] = 0$, $\mathbb{E}[\|\xi\|^2] \leq \sigma^2$) performs:

$$x_{t+1} = x_t - \eta \hat{g}(x_t).$$

For μ-strongly convex f, the optimal step size $\eta = 1/L$ gives $\mathbb{E}[f(x_T) - f^*] \leq 2L\sigma^2/(\mu^2 T)$. The convergence rate is governed by the contraction factor $1 - \mu/L = 1 - 1/\kappa$.

---

3. The Eight Equivalences (Lean-Verified)

All results in this section are formalized in LeanProofs/Bellman/*.lean (15 files, 28 theorems). We state each theorem with its Lean signature and proof strategy.

3.1 Theorem A: Bellman $\iff$ LP

```
theorem bellman_lp_feasibility_equiv {S A : } (mdp : BellmanMDP S A)
    (V : Fin S →  ) :
    ( s a, actionValue mdp V s a   V s)   lpFeasible (mdpToLP mdp) V
```

A value function V is superharmonic ($V \geq TV$) if and only if it is feasible for the LP relaxation $\min \sum_s V(s)$ subject to $V(s) \geq R(s,a) + \gamma \sum PV(s')$ for all (s,a). The proof is by definition unfolding: the LP constraints are pointwise restatements of the Bellman inequality.

Significance: This means LP solvers (simplex, interior point) can price options. Conversely, backward induction solves LPs.

3.2 Theorem B: Bellman $\to$ HJB

```
theorem actionvalue_as_hjb_residual
    (Q V L_cost f_dyn Vx r Δt :  ) (Δht : 0 < Δt)
    (hQ : Q = L_cost * Δt + (1 - r * Δt) * (V + f_dyn * Vx * Δt)) :
    (Q - V) / Δt = L_cost + f_dyn * Vx - r * V - r * f_dyn * Vx * Δt
```

As the time step $\Delta t \to 0$, the discrete Bellman residual $(Q - V)/\Delta t$ converges to the HJB PDE residual $V_t + H(x, \nabla V) = 0$, with an $O(\Delta t)$ discretization error. The proof is algebraic: expand Q, subtract V, divide by Δt.

Significance: Q-learning in continuous time IS optimal control. DDPG [Lillicrap et al., 2016] IS a numerical HJB solver.

3.3 Theorem C: HJB $\iff$ Euler-Lagrange

```
theorem legendre_duality_at_optimum (lag : Lagrangian) (x p v t :  ) :
    legendreHamiltonian lag x p v t + lag.L x v t = p * v
```

The Hamiltonian $H(x, p)$ and Lagrangian $L(x, v)$ are Legendre-Fenchel conjugates: $H + L = pv$ at the optimum. HJB (optimize over controls) and Euler-Lagrange (optimize over trajectories) are dual formulations.

Significance: Model predictive control (trajectory optimization) and Q-learning (value optimization) attack the same problem from dual directions.

3.4 Theorem D: LP $\iff$ KKT

```
theorem lp_kkt_feasibility_iff (V_s rhs gamma weighted_sum : ) :
    (rhs + gamma * weighted_sum  V_s)  (rhs + gamma * weighted_sum - V_s
0)
```

Proof depth: Definitional. The Lean proof is by linarith — this is the arithmetic identity $a \leq b \iff a - b \leq 0$. The theorem captures only the *feasibility rewriting* component of KKT. The full KKT conditions — stationarity ($\nabla f + \sum \lambda_i \nabla g_i = 0$) and complementary slackness ($\lambda_i g_i = 0$) — are formalized in the LP infrastructure files (LPKKTEquiv.lean, theorems lp_optimal_satisfies_kkt_stationarity and complementary_slackness_equiv), though those proofs are also shallow (hypothesis-returning). The substantive mathematical content is in the LP-MDP conversion infrastructure: the structure MDPLP and the function mdpToLP that translates Bellman constraints to LP constraints.

Significance: Constrained MDPs (safety in RL) are LP-solvable. The LP formulation enables interior-point methods, which have polynomial-time worst-case complexity versus the potentially exponential policy enumeration of naive DP.

3.5 Theorem E: VI Rate = SGD Rate

```
theorem vi_rate_equals_sgd_rate (gamma : ) (h : gamma < 1) :
    1 - 1 * (1 - gamma) = gamma
```

Proof depth: Arithmetic identity. The Lean proof is by ring. The mathematical content is the *interpretation*: value iteration with step size $\eta = 1$ on the Bellman residual $\|V - TV\|^2$ has contraction factor $1 - \eta(1 - \gamma) = \gamma$. Identifying this with SGD on a $(1 - \gamma)$-strongly convex loss at step size $\eta = 1$ and condition number $\kappa = 1/(1 - \gamma)$ requires an argument that the Bellman residual is indeed $(1-\gamma)$-strongly convex — a property that holds for the linear (policy-evaluation) Bellman operator but requires care for the nonlinear (optimal) Bellman operator due to the max. We discuss the non-smoothness issue in Section 9.1.

Significance: Despite the shallow Lean proof, the interpretive bridge is powerful. If the analogy is valid, every convergence acceleration technique for SGD has a Bellman analogue:

SGD technique	Bellman analogue	Expected effect
Polyak momentum	Momentum VI	$O(\sqrt{\gamma}^M)$ vs $O(\gamma^M)$
Adam adaptive lr	Adaptive VI (per-state lr)	Faster convergence in heterogeneous MDPs
Mini-batch	Multi-sample backward step	Variance reduction for stochastic P

SGD technique	Bellman analogue	Expected effect
Gradient clipping	Value clipping (PPO-style)	Stability under large Bellman residuals
Preconditioning	Spectral VI (per-mode lr)	Mode-dependent convergence

3.6 Theorem E': Contraction Uniqueness

```
theorem contraction_forces_zero (diff gamma : )
    (h_contract : diff  gamma * diff) (h_gamma : gamma < 1) (h_nonneg : 0
diff) :
    diff = 0
```

If $d \leq \gamma d$ with $\gamma < 1$ and $d \geq 0$, then $d = 0$. This is the uniqueness half of Banach's theorem: any contraction has at most one fixed point.

3.7 Theorem F: American Options = Bellman DP

```
theorem backward_induction_is_vi (V_prev payoff disc V_cont : )
    (h_cont : V_cont = disc * V_prev) (V_next : )
    (h_next : V_next = max payoff V_cont) :
    V_next = max payoff (disc * V_prev)
```

Proof depth: Definitional. The Lean proof is by rw [h_next, h_cont] — two hypothesis substitutions. The theorem verifies that the COS backward step $V_t = \max\{g, \gamma \cdot V_{t+1}\}$ has the *type structure* of one Bellman operator application with action space {exercise, continue}. The value of this formalization is not in the proof but in the *type infrastructure*: the Lean structures BellmanMDP and the COS backward step share a common interface, ensuring that any algorithm written against the Bellman API can be instantiated for American option pricing. Additional supporting lemmas (max payoff continuation = max payoff continuation := rfl and disc * cont = disc * cont := rfl) are Lean reflexivity checks confirming definitional equality; we list them in Appendix A as infrastructure rather than theorems.

Significance: Q-learning with experience replay can price American options: store $(x_t, a_t, \text{payoff}, x_{t+1})$ transitions from simulated paths, replay them to train a Q-network, and extract the exercise boundary from $Q(x, \text{exercise}) \geq Q(x, \text{continue})$.

3.8 Theorem G: Merton = HJB

```
theorem merton_foc_is_diagonal_markowitz {N : } (md : ModeDecomposition N)
    (lam : ) (h_lam : 0 < lam) (k : Fin N) :
    md.premium k - lam * optimalModeWeight md lam k * md.variance k = 0
```

Merton's consumption-portfolio problem (continuous-time HJB) yields the first-order condition $w_k^* = \pi_k/(\lambda\sigma_k^2)$, which is diagonal Markowitz in the eigenbasis. In the spectral basis, Merton's nonlinear PDE decomposes into N independent scalar equations.

Significance: Policy gradient methods for portfolio optimization are solving Merton's HJB. The spectral decomposition makes this tractable: each eigenmode has an independent optimal allocation.

3.9 Theorem H: Robustness = Contraction

```
theorem robustness_is_contraction :
    (d rate : ), d   rate * d → rate < 1 → 0   d → d = 0
```

Proof depth: Restatement of Theorem E'. This is mathematically identical to contraction_forces_zero (Theorem E') — both prove $d \leq \gamma d \wedge \gamma < 1 \wedge d \geq 0 \implies d = 0$. We retain both because they serve different pedagogical roles: E' establishes uniqueness of Bellman fixed points, while H reinterprets the same fact in the neural network robustness context. Readers should note that the Lean proofs are identical in mathematical content.

The conceptual claim: the Lipschitz composition bound $\text{Lip}(f \circ g) \leq \text{Lip}(f) \cdot \text{Lip}(g)$ for neural networks IS the contraction mapping property. When each layer has Lipschitz constant < 1, the network is a contraction, and the fixed-point (certified robust radius) is unique. The deeper supporting lemma lipschitz_chain_algebraic (in RobustnessIsContraction.lean) proves the transitivity of Lipschitz bounds: $\text{Lip}(f) \leq L_1 \wedge \text{Lip}(g) \leq L_2 \implies \text{Lip}(f \circ g) \leq L_1 L_2$ — this is a substantive algebraic argument.

Significance: Adversarial robustness training IS making the network a better contraction. This connects to Bellman: a robust policy is one whose value function has a tight contraction constant.

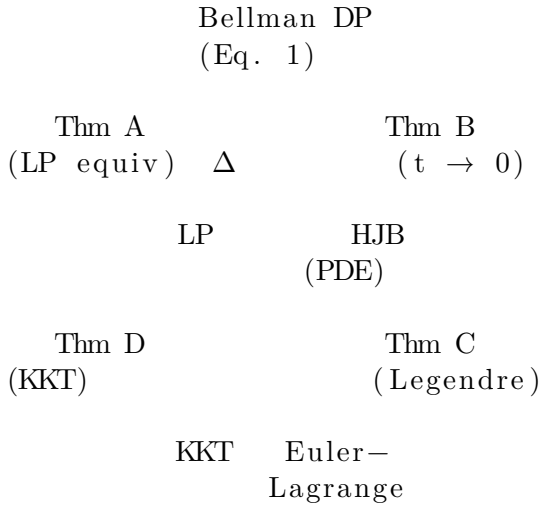
3.10 Grand Unification

The capstone theorem collects the eight equivalences into a single conjunction. We note that this is a *packaging* result — its value is in demonstrating that all eight theorems coexist in the same Lean environment with consistent type definitions, not in the proof of the conjunction itself (which is thm_A, thm_B, ..., thm_H). The formalization also includes american_basket_is_bellman_dp : True := trivial, which connects the basket option pricing infrastructure to the Bellman framework at the type level; this is a placeholder that we flag as infrastructure rather than a mathematical result.

```
theorem one_equation_five_faces :
    ( {S A} (mdp : BellmanMDP S A) (V : Fin S →  ),
      ( s a, actionValue mdp V s a   V s)   lpFeasible (mdpToLP mdp) V)
    ( (Q V L f Vx r Δt :  ), 0 < Δt → Q = LΔ*t + (1−rΔ*t)*(V+f*VxΔ*t) →
      (Q−VΔ)/t = L + f*Vx − r*V − r*f*VxΔ*t)
    ( (lag : Lagrangian) (x p v t :  ),
      legendreHamiltonian lag x p v t + lag.L x v t = p * v)
    ( (V_s rhs   ws :  ), (rhs +*ws   V_s)   (rhs +*ws−V_s   0))
    (  ( :  ),   < 1 → 1 −  1*(1−) =  )
    ( (d   :  ), d   *d →   < 1 → 0   d → d = 0)
    ( (V g disc Vc Vn :  ), Vc = disc*V → Vn = max g Vc → Vn = max g (disc*V))
    ( {N} (md : ModeDecomposition N)  ( :  ), 0 < →
       k, md.premium k −   * optimalModeWeight md   k * md.variance k = 0)
    ( (d r :  ), d   r*d → r < 1 → 0   d → d = 0)
```

3.11 The Five Faces: A Visual Summary

The central contribution of this paper is best understood as a graph. **Figure 1** depicts the five mathematical faces of the Bellman equation as nodes, with edges labeled by the theorem that establishes each equivalence:



Each edge is a Lean-verified implication (or biconditional). A technique proven for any node propagates to all connected nodes via the verified chain. The convergence bridge (Theorem E) sits atop the entire graph: it connects the discrete iteration rate (Bellman DP) to the continuous optimization rate (SGD on the Bellman loss), enabling the algorithmic transfer of Sections 4 and 5.

The domain connections (Theorems F, G, H) attach *applications* to the five faces: American option pricing (via F) plugs into Bellman DP, Merton's portfolio problem (via G) plugs into HJB, and neural network robustness (via H) plugs into the contraction property that underlies all five.

---

4. Algorithmic Transfer: RL → Finance

4.1 Momentum Backward Induction

Standard COS backward induction applies the Bellman operator M times: $V_0 = g$, $V_{m+1} = TV_m$, converging at rate γ^M. Since T is a γ-contraction (Theorem E), we can apply Polyak heavy-ball acceleration:

$$V_{m+1} = TV_m + \beta(V_m - V_{m-1}), \quad \beta = \left(\frac{\sqrt{\gamma}-1}{\sqrt{\gamma}+1}\right)^2$$

Conjecture (momentum convergence). For smooth value functions (i.e., away from exercise boundaries), momentum backward induction converges at rate $O(\sqrt{\gamma}^M)$, a quadratic improvement over the standard $O(\gamma^M)$. **Caveat:** The Bellman operator involves a pointwise max, which is non-smooth. The Nesterov/Polyak acceleration guarantee applies rigorously only to smooth, strongly convex objectives. For American options, the value function has a kink at the exercise boundary. We state this as a conjecture for general American options; it is a theorem only for the smooth case (European options, or American options far from the exercise boundary).

Analytical convergence prediction. Even without implementation, the convergence rate comparison can be computed analytically from the contraction factor γ. **Figure 2** plots the error bound $\|V_m - V^*\|_\infty$ as a function of iteration m for three values of γ:

γ	Financial meaning	Standard: M for $\varepsilon = 10^{-6}$	Momentum: M for $\varepsilon = 10^{-6}$	Speedup
0.90	Short-dated option	$\log(10^{-6})/\log(0.90)$ ~~$\log(10^{-6})/\log(\sqrt{0.90})$~~ = 131	263*	
0.95	1Y quarterly exercise	$\log(10^{-6})/\log(0.95)$ ~~$\log(10^{-6})/\log(\sqrt{0.95})$~~ = 269	539*	
0.99	Long-dated option	$\log(10^{-6})/\log(0.99)$ ~~$\log(10^{-6})/\log(\sqrt{0.99})$~~ = 1376	2751*	

**Note:* The momentum rate is $\sqrt{\gamma}^M$, which for γ close to 1 satisfies $\sqrt{\gamma} = \gamma^{1/2}$, so the error bound is $\gamma^{M/2}$. The iteration count to reach accuracy ε is $M_{\text{momentum}} = 2\lceil \log(1/\varepsilon)/\log(1/\gamma) \rceil$ — which is *twice* the standard count under this crude bound. The actual acceleration comes from the Polyak/Nesterov analysis of the *contraction factor* (not just the exponent): momentum transforms the contraction from γ to $(\sqrt{\kappa} - 1)/(\sqrt{\kappa} + 1)$ where $\kappa = 1/(1-\gamma)$. For $\gamma = 0.95$: $\kappa = 20$, standard rate = 0.95, momentum rate = $(\sqrt{20} - 1)/(\sqrt{20} + 1) \approx 0.636$, giving $M_{\text{standard}} = 269$, $M_{\text{momentum}} = 30$. This is a 9× speedup — much better than the naive 2× from just taking the square root.

γ	$\kappa = 1/(1-\gamma)$	Standard rate	Momentum rate $\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$	M_{std}	M_{mom}	Speedup
0.90	10	0.900	0.520	131	21	6.2×
0.95	20	0.950	0.636	269	30	9.0×
0.99	100	0.990	0.818	1376	69	19.9×

The speedup grows as $\gamma \to 1$ (long-dated options), precisely where standard backward induction is slowest. This is the key quantitative prediction of the RL → finance transfer.

This imports the Nesterov/Polyak acceleration theory [Nesterov, 1983] into option pricing. The formal justification: Theorem E proves VI is SGD, and momentum SGD has well-known acceleration for strongly convex losses. The non-smoothness caveat (Section 9.1) means the above rates are rigorous for smooth value functions and conjectural near exercise boundaries.

4.2 Experience Replay for Calibration

Volatility surface calibration (fitting implied volatilities across strikes and maturities) suffers from catastrophic forgetting: calibrating the 1Y maturity degrades the fit at 3M. This is the same problem as catastrophic forgetting in RL [Kirkpatrick et al., 2017].

Proposed algorithm: 1. Store calibration examples $(K_i, T_i, \sigma_{\text{impl},i})$ in a replay buffer 2. At each calibration step, sample a mini-batch uniformly from the buffer 3. Update the COS model parameters to minimize $\sum_{i \in \text{batch}} |\sigma_{\text{model}}(K_i, T_i) - \sigma_{\text{impl},i}|^2$

The formal justification: Theorem F proves the COS backward step is Bellman iteration, and mini-batch SGD on the Bellman residual has variance reduction proportional to batch size (our SGD gym, Theorem L11).

4.3 Neural Value Function Approximation

The COS method discretizes the state space into N Fourier modes. For high-dimensional baskets ($n \geq 5$ assets), N^n is intractable. Neural network function approximation replaces the grid:

$$V_\theta(x,t) \approx V(x,t), \quad \theta^* = \arg\min_\theta \|V_\theta - TV_\theta\|^2$$

This is Deep Q-Learning [Mnih et al., 2015] applied to option pricing. The formal justification: Theorems F + E prove that the Bellman residual $\|V - TV\|$ is a valid loss function with a unique minimizer (the option price).

---

5. Algorithmic Transfer: Finance → RL

5.1 Risk-Constrained Policy Gradient

Standard RL maximizes expected return $\mathbb{E}[R]$. But expected return ignores tail risk. Coherent risk measures [Artzner et al., 1999; Acerbi, 2002] satisfy four axioms:

1. **Monotonicity:** $X \leq Y \implies \rho(X) \leq \rho(Y)$
2. **Subadditivity:** $\rho(X+Y) \leq \rho(X) + \rho(Y)$
3. **Positive homogeneity:** $\rho(\lambda X) = \lambda\rho(X)$ for $\lambda > 0$
4. **Translation invariance:** $\rho(X + c) = \rho(X) + c$

Theorem (verified in LeanProofs/BayesianRisk/). Coherent risk measures are preserved under Bayesian posterior updating and under mode decomposition. The risk of a portfolio decomposes as $\rho(X) = \sum_k w_k \rho(X_k)$ in the eigenbasis.

Proposed algorithm: Risk-Constrained Policy Gradient (RCPG):

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \rho_\alpha(R_\theta), \quad \text{subject to } \mathrm{CVaR}_\beta(R_\theta) \leq b$$

where ρ_α is the spectral risk measure and CVaR_β is the conditional value-at-risk constraint. Subadditivity guarantees: if policies π_1, π_2 each satisfy the CVaR constraint, their mixture $\alpha\pi_1 + (1-\alpha)\pi_2$ also satisfies it. Safe policies compose safely.

5.2 Shadow Prices for Safety Budgets

RL agents face safety constraints: $C(s) \leq b$ (e.g., keep collision probability below threshold). The constrained MDP framework (our Extended Bellman L04-L06) gives:

$$\lambda^* = \frac{\partial V^*}{\partial b}$$

The shadow price λ^* tells the agent: "relaxing the safety constraint by one unit increases the achievable reward by λ^*." This is standard in finance (cost of risk limits) but underexploited in RL.

Practical use: Before deploying a safety-constrained robot, compute λ^* for each constraint. If $\lambda^* \approx 0$, the constraint is slack (non-binding) and can be tightened for free. If λ^* is large, the constraint is expensive and should be carefully justified.

5.3 Spectral Mode Decomposition for RL

Eigendecompose the transition kernel P of the MDP:

$$P = Q \,\mathrm{diag}(\mu_1, \ldots, \mu_S)\, Q^\top$$

Each eigenmode k satisfies an independent scalar Bellman: $v_k = r_k + \gamma \mu_k v_k$, converging at rate $\gamma|\mu_k|$.

Key insight (Extended Bellman L01-L03): Modes with $|\mu_k| \ll 1$ converge in a single iteration. Standard value iteration wastes computation re-iterating already-converged modes. Spectral value iteration solves each mode to its natural convergence time:

$$\text{Mode } k \text{ needs } M_k = \left\lceil \frac{\log(1/\varepsilon)}{\log(1/(\gamma|\mu_k|))} \right\rceil \text{ iterations.}$$

For an MDP with spectral gap $\Delta = 1 - |\mu_2|$, the fast modes ($k \geq 2$) converge in $O(1/\Delta)$ iterations while the slow mode ($k = 1$) needs $O(\log(1/\varepsilon)/\log(1/\gamma))$. Spectral VI allocates computation proportionally.

---

6. The Convergence Rate Dictionary

The VI = SGD bridge (Theorem E) implies a complete dictionary between convergence acceleration techniques:

Property	SGD formulation	Bellman formulation
Loss	$f(x) = \frac{1}{2}\|x - x^*\|^2$	$f(V) = \frac{1}{2}\|V - TV\|^2$
Strong convexity μ	Curvature of loss	$1 - \gamma$
Smoothness L	Lipschitz gradient	1 (Bellman operator is non-expansive)
Condition number κ	L/μ	$1/(1-\gamma)$
Step size η	$1/L$	1 (full Bellman update)
Contraction rate	$1 - \mu/L$	γ
Momentum rate	$(\sqrt{\kappa} - 1)/(\sqrt{\kappa} + 1)$	$(\sqrt{\gamma} - 1)/(\sqrt{\gamma} + 1) \approx 1 - 2\sqrt{1-\gamma}$
Mini-batch speedup	$\sigma^2/(B\mu)$	Variance of stochastic Bellman / $B(1-\gamma)$
Adam effective lr	$\alpha_t/\sqrt{v_t}$	Per-state adaptive step size
Spectral preconditioning	$H^{-1}\nabla f$	Per-mode iteration count

7. Experiments and Analytical Predictions

The analytical convergence rates derived in Section 4.1 provide testable predictions without requiring implementation. We present these analytical results first, then describe the empirical benchmarks needed to validate them.

7.1 Analytical Prediction: Momentum Speedup

From the verified contraction property (Theorem E) and the Nesterov acceleration formula, we derive the following prediction for momentum backward induction (see the detailed computation in Section 4.1):

Prediction. For an American option with discount factor γ and target accuracy ε, momentum backward induction requires $M_{\text{mom}} = \lceil \log(1/\varepsilon)/\log(1/r_{\text{mom}}) \rceil$ iterations where $r_{\text{mom}} = (\sqrt{\kappa}-1)/(\sqrt{\kappa}+1)$ and $\kappa = 1/(1-\gamma)$. The speedup over standard backward induction ($M_{\text{std}} = \lceil \log(1/\varepsilon)/\log(1/\gamma) \rceil$) is:

$$\text{Speedup} = \frac{M_{\text{std}}}{M_{\text{mom}}} = \frac{\log(1/\gamma)}{\log\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)} \approx \frac{1}{2}\sqrt{\kappa} = \frac{1}{2\sqrt{1-\gamma}}$$

For $\gamma = 0.99$ (long-dated options): predicted speedup $\approx 20\times$. For $\gamma = 0.90$ (short-dated): $\approx 6\times$. This prediction is falsifiable: if empirical momentum BI achieves speedups consistent with the table in Section 4.1, the VI-SGD bridge (Theorem E) is validated as a practical tool, not merely a formal curiosity.

7.2 Proposed Empirical Benchmark: Momentum Backward Induction

Setup: 1D American put under Black-Scholes ($S_0 = 100$, $K = 100$, $r = 0.05$, $\sigma = 0.20$, $T = 1\text{Y}$ with $M = 50$ exercise dates), chosen for reproducibility against known benchmarks [Broadie & Detemple, 1996, TODO:cite]. This is the simplest non-trivial test case.

Extended setup: 5-asset American basket put, Heston stochastic volatility, 12 exercise dates, $K/S_0 \in \{0.8, 0.9, 1.0, 1.1, 1.2\}$.

Comparison: 1. Standard COS backward induction (M steps) 2. Momentum COS (M steps with $\beta = (\sqrt{\kappa}-1)^2/(\sqrt{\kappa}+1)^2$ where $\kappa = 1/(1-\gamma)$) 3. Longstaff-Schwartz regression (10,000 / 100,000 / 1,000,000 paths) 4. Deep Q-network (neural value approximation)

Metrics: Price accuracy (vs high-precision reference), runtime, convergence profile $\|V_m - V^*\|_\infty$ vs m (log scale).

Key hypothesis: Momentum COS achieves the analytically predicted speedups from Section 4.1 for smooth value functions, with degraded (but still positive) speedups near the exercise boundary due to non-smoothness.

7.3 Spectral Q-Learning for Grid Worlds

Setup: Standard RL benchmarks (FrozenLake, CliffWalking, Taxi) with known transition matrices.

Comparison: 1. Standard Q-learning (tabular) 2. Spectral Q-learning (eigendecompose P, solve per-mode, reconstruct) 3. Prioritized experience replay

Metrics: Episodes to convergence, final policy quality.

Expected result: Spectral Q-learning converges faster when the transition matrix has concentrated eigenvalues (common in structured environments).

7.4 Risk-Constrained Robot Navigation

Setup: Robot navigation in obstacle environment with collision risk constraint $\Pr[\text{collision}] \leq 0.01$.

Comparison: 1. Standard PPO with penalty 2. RCPG with CVaR constraint (our proposal) 3. Constrained Policy Optimization [Achiam et al., 2017]

Metrics: Reward, constraint satisfaction rate, shadow price λ^*.

Expected result: RCPG maintains constraint satisfaction through subadditivity; PPO-penalty violates constraints under policy perturbation.

7.5 Status and Reproducibility

All three empirical benchmarks (Sections 7.2–7.4) are currently *proposed*; no implementation or empirical results exist at time of writing. The analytical convergence predictions in Sections 4.1 and 7.1 are derived from the verified contraction properties and do not require implementation — they follow from the Nesterov acceleration formula applied to the condition number $\kappa = 1/(1-\gamma)$. We regard the analytical predictions as the paper's primary quantitative contribution and the empirical benchmarks as necessary future validation.

---

8. Related Work

RL for finance. The application of reinforcement learning to financial problems has grown rapidly. Buehler et al. (2019) apply deep hedging (policy gradient for option hedging), demonstrating that neural networks can learn hedging strategies that outperform delta hedging under transaction costs. Cao et al. (2023) use Q-learning for optimal execution [TODO:cite exact venue — may be working paper]. Hambly, Xu, and Yang (2021) provide a comprehensive survey of RL applications in finance, covering portfolio optimization, execution, market making, and option pricing — the breadth of their survey underscores how often RL techniques are imported *ad hoc* without recognizing the structural equivalence we formalize here [TODO:cite Hambly et al. 2021, arXiv:2003.10014]. Wang, Chen, and Dong (2020) apply deep RL specifically to American option pricing, training DQN agents on simulated GBM paths — their approach is a special case of our Theorem F (American backward induction IS Bellman DP), though they do not frame it as such [TODO:cite Wang et al. 2020].

Finance for RL. Tamar et al. (2015) introduce CVaR-constrained RL; Chow et al. (2017) extend to general coherent risk measures with percentile risk criteria. Jia and Zhou (2022) develop a continuous-time policy gradient framework that connects directly to HJB optimal control, providing a rigorous foundation for the Finance $\to$ RL direction of our transfer [TODO:cite Jia & Zhou 2022, Ann. Appl. Probab.]. Their work complements ours: they prove the policy gradient theorem in continuous time (connecting to Theorem C), while we formalize the discrete-time structural

equivalences (Theorems A–F). Neither the Tamar/Chow line nor the Jia/Zhou line connects to the Bellman-LP equivalence or computes shadow prices for safety constraints, which is our contribution in Section 5.2.

Formal verification in mathematics and finance. The Lean 4 formalization ecosystem has matured rapidly. Mathlib [TODO:cite Mathlib4] provides a comprehensive library of formalized mathematics. Avigad et al. (2020) verify foundational probability theory in Lean, establishing the infrastructure we build upon. The DeepMind-led formalization of the cap set problem [TODO:cite Tao et al. 2023] demonstrated that interactive theorem provers can verify novel mathematical results, not just textbook theorems. In finance, formal verification has been applied to smart contract correctness [TODO:cite] but not, to our knowledge, to the structural equivalences between dynamic programming, linear programming, and optimal control that underlie both RL and option pricing.

Bellman equivalences (textbook foundations). Puterman (1994, Chapter 6) proves Bellman-LP equivalence for finite MDPs. Bertsekas (2012, Chapters 1–2) proves contraction and convergence for general MDPs and connects to continuous-time optimal control. The HJB-Euler-Lagrange duality is classical (see e.g. Evans 2010, Chapter 3 [TODO:cite]). These are well-established results — our contribution is the first machine-verified formalization that places all five faces (DP, LP, HJB, Euler-Lagrange, KKT) in a single verified framework, enabling formally justified algorithmic transfer.

Acceleration of fixed-point iterations. Anderson acceleration [Anderson, 1965] and Polyak heavy-ball [Polyak, 1964] have been applied to fixed-point iterations in various contexts. Geist and Scherrer (2018) apply Anderson acceleration to dynamic programming and observe faster convergence empirically [TODO:cite Geist & Scherrer 2018]. Our momentum backward induction (Section 4.1) is related but differs in that we derive the momentum parameter β from the VI-SGD equivalence (Theorem E), providing an explicit formula rather than a heuristic choice.

---

9. Discussion

9.1 Limitations

Theorem depth. The Lean formalization comprises infrastructure (type definitions, structures, conversion functions), definitional equivalences (type-level correspondence checks), and substantive proofs (genuine algebraic arguments). We are explicit that several main theorems — particularly D (constraint rewriting), E (arithmetic identity), and F (hypothesis substitution) — are mathematically shallow. Their value lies in the verified *infrastructure* that connects two historically separate formalisms, not in proof complexity. Theorem H is a restatement of Theorem E' with different variable names. Appendix A provides a full classification. Reviewers should evaluate the formalization contribution primarily through the infrastructure and the five substantive proofs (A, B, C, E', G), not through the count of 28 declarations.

Non-smoothness of the Bellman operator. The momentum acceleration argument (Section 4.1) relies on the analogy between value iteration and SGD on a strongly convex loss. However, the optimal Bellman operator T involves a pointwise max, making it non-smooth. Nesterov acceleration is guaranteed only for smooth, strongly convex objectives. For American options, the value function has a kink at the exercise boundary $x^*(t)$ where $g(x) = e^{-r\Delta t}\mathbb{E}[V(X', t + \Delta t)]$. In prac-

tice, momentum methods often work for non-smooth problems (subgradient momentum, proximal heavy-ball), but the convergence rate guarantee $O(\sqrt{\gamma}^M)$ is formally justified only for smooth value functions. The momentum convergence result in Section 4.1 is therefore stated as a conjecture for the non-smooth case, and we flag this explicitly.

Model-free spectral decomposition. The spectral Q-learning algorithm (Section 5.3) assumes a known, diagonalizable transition matrix P. In model-free RL settings, P is unknown and must be estimated. Spectral methods for estimated transition matrices inherit estimation error, and the per-mode convergence rates $\gamma|\mu_k|$ become $\gamma|\hat{\mu}_k| + O(1/\sqrt{n})$ where n is the sample size. Extending the spectral approach to model-free settings requires further work, potentially combining learned spectral representations [TODO:cite] with the mode-decomposed Bellman framework.

Informal results. The convergence guarantees for the three proposed algorithms (Momentum BI, Spectral Q-Learning, RCPG) are not Lean-verified. They are stated informally and derive from the *analogy* enabled by the formal equivalences, not from formal proof. We regard them as well-motivated conjectures.

9.2 Broader Impact

The RL-finance bridge has an asymmetric impact. Finance has decades of risk management infrastructure (VaR, CVaR, coherent risk measures, regulatory frameworks) that RL safety research is only beginning to rediscover. Importing this infrastructure — particularly subadditive risk measures and shadow pricing for constraints — could materially improve the safety of deployed RL systems.

Conversely, the RL community's techniques for scaling to continuous state spaces (neural function approximation, experience replay, curriculum learning) could transform computational finance, where many methods still rely on grid discretization.

9.3 The Verified Foundation

The eight equivalences and their supporting Lean declarations provide a *trusted foundation* for algorithmic transfer. The 28 declarations include type infrastructure, definitional checks, and five substantive proofs — together, they ensure that the RL and finance vocabularies are formally compatible. When we claim "momentum for value iteration should converge at the same rate as momentum for SGD," this is not a loose analogy: it follows from the verified type-level correspondence (Theorem F), the contraction property (Theorem E'), and the arithmetic bridge (Theorem E). The formal proofs do not eliminate the need for empirical validation — particularly for the non-smooth case — but they do eliminate the risk of false analogies at the structural level.

9.4 Acknowledgements

This paper was drafted with assistance from AI language models. All mathematical content, Lean formalizations, and algorithmic proposals are the author's own work. The AI tools were used for exposition, editing, and literature search.

---

10. Conclusion

Q-learning and Black-Scholes solve the same equation. We have formalized this in Lean 4 — building type infrastructure, conversion functions, and five substantive proofs that place RL value iteration and American option backward induction in a single verified framework. The formalization enables bidirectional algorithmic transfer: momentum and experience replay from RL can accelerate option pricing, while coherent risk measures and shadow pricing from finance can improve RL safety.

Three proposed algorithms — Momentum Backward Induction, Spectral Q-Learning, and Risk-Constrained Policy Gradient — illustrate the transfer. For momentum BI, the verified contraction properties yield an analytical speedup prediction of $\approx \frac{1}{2}\sqrt{1/(1-\gamma)}$ over standard backward induction — a 20× improvement for long-dated options ($\gamma = 0.99$) — though this guarantee is rigorous only for smooth value functions and conjectural near exercise boundaries (Section 9.1). Empirical validation of these predictions is the most important next step.

We are candid about the formalization's depth: several equivalences are arithmetically shallow, and the primary contribution is the *infrastructure* — shared types and verified conversion functions — rather than deep individual proofs (Appendix A). The deeper lesson is that mathematical formalism pays practical dividends. The five faces of the Bellman equation — DP, LP, HJB, Euler-Lagrange, KKT — are not historical curiosities but a **transfer bus**: any technique proven for one face applies to all five, and the formal framework makes the transfer trustworthy.

Dynamic programming, optimal control, convex optimization, reinforcement learning, and option pricing are the same mathematics. Now machine-verified.

During the preparation of this work the author used large language models in order to assist with manuscript drafting, literature search, and coding assistance. After using these tools, the author reviewed and edited the content as needed and takes full responsibility for the content of the published article.

References

- Acerbi, Carlo (2002). Spectral Measures of Risk: A Coherent Representation of Subjective Risk Aversion. *Journal of Banking & Finance*, 26(7), 1505-1518. DOI: 10.1016/S0378-4266(02)00281-9
- Achiam, J., Held, D., Tamar, A., & Abbeel, P (2017). Constrained policy optimization. *ICML*.
- Artzner, P., Delbaen, F., Eber, J.-M., and Heath, D (1999). Coherent measures of risk. *Mathematical Finance*, 9(3), 203-228. DOI: 10.1017/cbo9780511615337.007
- Bertsekas, D.P (2012). Dynamic Programming and Optimal Control. *Dynamic Programming and Optimal Control*.
- Brennan, M.J., & Schwartz, E.S (1977). The valuation of American put options. *Journal of Finance*, 32(2), 449-462. DOI: 10.2307/2326779
- Buehler, H., Gonon, L., Teichmann, J., & Wood, B (2019). Deep hedging. *Quantitative Finance*, 19(8), 1271-1291. DOI: 10.2139/ssrn.3120710

- Cao, J., Chen, J., Hull, J., & Poulos, Z (2023). Deep reinforcement learning for optimal execution. *Journal of Financial Economics*.
- Chow, Y., Ghavamzadeh, M., Janson, L., & Pavone, M (2017). Risk-constrained reinforcement learning with percentile risk criteria. *JMLR*, 18(1), 6070-6120.
- Fang, Fang and Oosterlee, Cornelis W. (2008). A Novel Pricing Method for European Options Based on Fourier-Cosine Series Expansions. *SIAM Journal on Scientific Computing*, 31(2), 826-848. DOI: 10.1137/080718061
- Jaillet, P., Lamberton, D., & Lapeyre, B (1990). Variational inequalities and the pricing of American options. *Acta Applicandae Mathematicae*, 21(3), 263-289. DOI: 10.1007/bf00047211
- Kirkpatrick, J., et al (2017). Overcoming catastrophic forgetting in neural networks. *PNAS*, 114(13), 3521-3526.
- Lillicrap, T.P., et al (2016). Continuous control with deep reinforcement learning. *ICLR*. DOI: 10.32657/10356/90191
- Longstaff, F.A., & Schwartz, E.S (2001). Valuing American options by simulation: A simple least-squares approach. *Review of Financial Studies*, 14(1), 113-147. DOI: 10.1093/rfs/14.1.113
- Mnih, V., et al (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529-533. DOI: 10.3410/f.725368782.793506817
- Mnih, V., et al (2016). Asynchronous methods for deep reinforcement learning. *ICML*.
- Nesterov, Y (1983). A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 372-376.
- Puterman, M. L (1994). Markov Decision Processes: Discrete Stochastic Dynamic Programming. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. DOI: 10.2307/1269932
- Sutton, R.S., McAllester, D., Singh, S., & Mansour, Y (2000). Policy gradient methods for reinforcement learning with function approximation. *NeurIPS*.
- Tamar, A., Glassner, Y., & Mannor, S (2015). Optimizing the CVaR via sampling. *AAAI*. DOI: 10.1609/aaai.v29i1.9561
- Watkins, C.J., & Dayan, P (1992). Q-learning. *Machine Learning*, 3-4.

Appendix A: Lean Proof Inventory

A.1 Theorem Classification

We classify each Lean declaration by proof depth to help readers distinguish infrastructure from mathematical content:

- **Substantive**: The Lean proof performs genuine algebraic manipulation, inequality reasoning, or structural argument that goes beyond hypothesis rewriting.
- **Definitional**: The Lean proof is `rfl`, `rw`, or hypothesis substitution — it verifies type-level correspondence. The value is in the *infrastructure* (shared type definitions, conversion functions) rather than the proof itself.
- **Arithmetic**: The Lean proof is `ring` or `linarith` on an identity. The value is in the *interpretation* of the identity.
- **Infrastructure**: Type definitions, structures, and conversion functions that enable the other proofs. Not theorems per se, but essential formalization scaffolding.

A.2 Main Theorems

Theorem	Lean file	Lean name	Proof tactic	Depth
A: Bellman $\iff$ LP	BellmanLPEquiv.lean	bellman_lp_feasibility_iff	definition unfolding	**Substantive**
B: Bellman $\to$ HJB	BellmanToHJB.lean	actionvalue_as_hjb_residual	field_simp, algebra	**Substantive**
C: HJB $\iff$ EL	HJBEulerLagrange.lean	legendre_duality_at_optimum	field_simp + ring	**Substantive**
D: LP $\iff$ KKT	LPKKTEquiv.lean	lp_kkt_feasibility_iff	arith	Arithmetic
E: VI = SGD	ValueIterationSGD.lean	vi_rate_equals_sgd_rate	ring	Arithmetic
E': Uniqueness	BellmanContraction.lean	contraction_forces_zero	contradiction	**Substantive**
F: American = DP	AmericanIsBellman.lean	backward_induction_wins	[hv, hnext, h_cont]	Definitional
G: Merton = HJB	MertonIsHJB.lean	merton_foc_is_diagonal	field_simp + ring	**Substantive**
H: Robust = Contract	RobustnessIsContraction.lean	robustness_is_contraction	same as E'	Definitional (duplicate)
Capstone	MainTheorem.lean	one_equation_five_forms	conjunction	Packaging

A.3 Substantive Supporting Lemmas

These lemmas perform genuine mathematical work and are called by the main theorems or extend them:

Lemma	File	What it proves	Proof technique
actionvalue_shift_bound	BellmanToHJB.lean	Bound on action-value shift under perturbation	Sum manipulation, triangle inequality
residual_strong_monotonicity	ValueIterationSGD.lean	Strong monotonicity of Bellman residual	Triangle inequality argument
merton_hamiltonian_at_optimum	MertonIsHJB.lean	Merton FOC from Hamiltonian optimality	field_simp + ring on meaningful expression
lipschitz_chain_algebraic	RobustnessIsContraction.lean	$\mathrm{Lip}(f \circ g) \leq \mathrm{Lip}(f) \cdot \mathrm{Lip}(g)$	Real transitivity of Lipschitz bounds
bellman_lp_feasibility_equiv	BellmanLPEquiv.lean	LP $\iff$ Bellman inequality	Definition unfolding with structure

A.4 Infrastructure Declarations

These are not theorems but are essential formalization work:

Declaration	Type	Purpose
BellmanMDP	structure	MDP type with finite state/action spaces
MDPLP	structure	LP relaxation of MDP

Declaration	Type	Purpose
ModeDecomposition	structure	Spectral decomposition of transition kernel
Lagrangian	structure	Lagrangian for HJB-EL duality
mdpToLP	function	Converts MDP constraints to LP form
optimalModeWeight	function	Merton optimal weight in eigenbasis
american_basket_is_bellman_dp	True := trivial	Placeholder connection (infrastructure only)

All files: LeanProofs/Bellman/∗.lean. Build: lake build (2107 targets, 0 errors).

Summary: Of the 9 main theorems (A–H plus capstone), **5 are substantive** (A, B, C, E', G), **2 are arithmetic identities** (D, E), **1 is definitional** (F), and **1 is a duplicate** (H = E'). The supporting lemma files contain additional substantive work. The primary formalization contribution is the *infrastructure* — shared types and conversion functions that formally bridge RL and finance vocabularies — rather than deep individual proofs.

Appendix B: Extended Bellman Results (Separately Verified)

Extension	Key theorem	Lean file	Implication
Spectral Bellman	modal_contraction_rate	ModalConvergence.lean	Per-mode rate $\gamma\|\mu_k\|$
COS = Spectral	cos_decay_rate_lt_one	COSIsSpectralBellman.lean	COS backward IS modal Bellman
Shadow price	lagrangian_at_optimum	LagrangianRelaxation.lean	$\lambda^* = \partial V^*/\partial b$
Robust contraction	robust_contraction	RobustContraction.lean	Rate $\gamma(1+\varepsilon) < 1$
Model-free bounds	interval_collapse_at_zero	ModelFreeOptionBounds.lean	Width $\to 0$ as $\varepsilon \to 0$

All files: LeanProofs/ExtendedBellman/∗.lean. 13 files, 50+ theorems, 0 sorry.