Weierstrass Institute for
Applied Analysis and Stochastics

Leibniz Association

Pricing American Options by Exercise Rate Optimization

Christian Bayer

Joint work with Raúl Tempone and Sören Wolfers

ETH Zürich

Mohrenstrasse 39 · 10117 Berlin · Germany · Tel. +49 30 20372 0 · www.wias-berlin.de

January 9, 2020

Outline

$$v(s_0) := v(0, s_0) = \sup_{\tau \in \mathcal{S}} E\left[Y_{\tau \wedge T} \mid S_0 = s_0\right]$$

- $S_t \in \mathbb{R}^d$ denotes the underlying asset price process, $d \geq 1$
- Y_t denotes the discounted cash-flow process, e.g., $Y_t = e^{-rt} g(S_t)$

$$g(s) = \left(K - \sum_{i=1}^{d} s_i\right)^+ \quad \text{or} \quad g(s) = \max_{i=1,\ldots,d} (s_i - K)^+$$

- E is the expectation w.r.t. a pricing measure P
- $\mathcal{S}$ denotes the set of $\mathcal{F}_t$-stopping times

Let $v(t, s)$ be time and asset dependent value function.

Dynamic programming principle

Value $v(t, s)$ equals expected value at future time, or value of exercising right now, whichever is larger:

$$v(t, s) \approx \max\{E[v(t + \Delta t, S_{t+\Delta t}) \mid S_t = s], g(s)\}$$

Making this rigorous leads to two state of the art algorithms that determine $v(t, s)$ backwards in time, starting with $t = T$ where $v(T, \cdot) \equiv g$

- Discretize the HJB PDE
- Directly solve the dynamic programming principle by Monte Carlo regression

Let $v(t, s)$ be time and asset dependent value function.

Dynamic programming principle

Value $v(t, s)$ equals expected value at future time, or value of exercising right now, whichever is larger:

$$v(t, s) \approx \max\{E[v(t + \Delta t, S_{t+\Delta t}) \mid S_t = s], g(s)\}$$

Making this rigorous leads to two state of the art algorithms that determine $v(t, s)$ backwards in time, starting with $t = T$ where $v(T, \cdot) \equiv g$

- Discretize the HJB PDE
- Directly solve the dynamic programming principle by Monte Carlo regression

For $t_N := T, t_{N-1} := T - T/N, \dots, t_0 := 0$:

- Assume we have approximation $v_n(\cdot)$ of $v(t_n, \cdot)$ that can be evaluated at arbitrary points
- Generate samples $s^{(m)} \sim S_{t_{n-1}}, 1 \le m \le M$
- For each sample, generate a number of future samples

$$s^{(m,k)} \sim \mathcal{L}\big(S_{t_n} \mid S_{t_{n-1}} = s^{(m)}\big), \quad 1 \le k \le K$$

- $c^{(m)} := \frac{1}{K} \sum_{k=1}^{K} v_n(s^{(m,k)})$, expected value of continuation from $s^{(m)}$
- Determine $p_{n-1}(\cdot)$ in some ansatz space V (e.g. some space of polynomials) by discrete L^2 regression:

$$p_{n-1} := \arg\min_{p \in V} \sum_{m=1}^{M} \left| p(s^{(m)}) - c^{(m)} \right|^2$$

- Let $v_{n-1}(s) := \max\{g(s), p_{n-1}(s)\}$

- Typically, v_n is only used to construct an approximation to the optimal stopping time τ^*, not for actual pricing.
- The more well-known Longstaff – Schwartz algorithm is a variant of the above.
- Actual implementations avoid inner simulations.

Problems

- Value function v has only one continuous derivative at boundary of E_∞
- Large ansatz spaces and many samples necessary for good accuracy
- Number of necessary samples to alleviate error propagation further grows exponentially in number of time steps

Dynamic programming principle for $\Delta t \to 0$ leads to a nonlinear free-boundary partial differential equation, for v and simultaneously for the optimal exercise boundary. For $d = 1$, the optimal exercise boundary is a function $L\colon [0,T] \to \mathbb{R}_+$ and the equation for a put option is

$$\begin{cases} v_t(t,s) + rsv_s(t,s) + \frac{1}{2}\sigma^2 s^2 v_{ss}(t,s) - rv(t,s) = 0, & s \geq L(t) \\ v(T,s) = (K-s)^+ \\ v(t,s) = (K-s)^+, & 0 \leq s \leq L(t) \\ v(t,\cdot) \in C^1, & 0 \leq t < T \end{cases}$$

Problems

Same problems with regularity of v; curse of dimensionality with regular grids; have to deal with a difficult nonlinear PDE and all the problems that come with it

Outline

In Markovian market models, future stock development only depends on current prices.

Optimal strategies only exploit current state

We may thus restrict optimization to *hitting times* of sets $B \subset [0,T] \times \mathbb{R}^d$:

$$v(s_0) = \sup_{B \in \mathcal{B}([0,T]\times\mathbb{R}^d)} \Psi(B) := \sup_{B \in \mathcal{B}([0,T]\times\mathbb{R}^d)} E[Y_{\tau_B \wedge T} \mid S_0 = s_0]$$

- $\tau_B := \inf\{t \geq t_0 : (t, S_t) \in B\}$ is the hitting time of $B \subset [0,T] \times \mathbb{R}^d$
- Technical condition: S is càdlàg and the probability space is complete.

In Markovian market models, future stock development only depends on current prices.

Optimal strategies only exploit current state

We may thus restrict optimization to *hitting times* of sets $B \subset [0,T] \times \mathbb{R}^d$:

$$v(s_0) = \sup_{B \in \mathcal{B}([0,T] \times \mathbb{R}^d)} \Psi(B) := \sup_{B \in \mathcal{B}([0,T] \times \mathbb{R}^d)} E[Y_{\tau_B \wedge T} \mid S_0 = s_0]$$

- $\tau_B := \inf\{t \geq t_0 : (t, S_t) \in B\}$ is the hitting time of $B \subset [0,T] \times \mathbb{R}^d$
- Technical condition: S is càdlàg and the probability space is complete.

Strategy to find option price:

1. Choose parametrization of subsets of $[0, T] \times \mathbb{R}^d$
2. Choose initial guess $B_0 \subset [0, T] \times \mathbb{R}^d$
3. Update to get $B_n \to B_\infty$ and $\Psi(B_n) \to \Psi(B_\infty) = v(s_0)$

Not so easy:

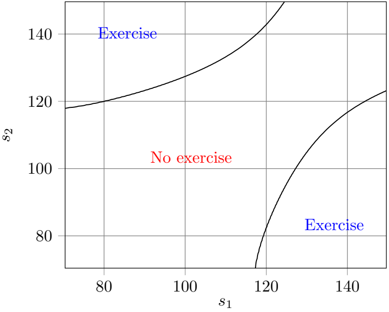
1. No obvious choice, no "orthogonal bases" of subsets
2. How to pick initial guess?
3. Recall lack of continuity of hitting times in general
4. Translates to lack of continuity $B \mapsto \frac{1}{M} \sum_{i=1}^{M} Y^i_{\tau_B \wedge T}$

Strategy to find option price:

1. Choose parametrization of subsets of $[0,T]\times\mathbb{R}^d$
2. Choose initial guess $B_0\subset[0,T]\times\mathbb{R}^d$
3. Update to get $B_n\to B_\infty$ and $\Psi(B_n)\to\Psi(B_\infty)=v(s_0)$

Not so easy:

1. No obvious choice, no "orthogonal bases" of subsets
2. How to pick initial guess?
3. Recall lack of continuity of hitting times in general
4. Translates to lack of continuity $B\mapsto\frac{1}{M}\sum_{i=1}^{M}Y^i_{\tau_B\wedge T}$

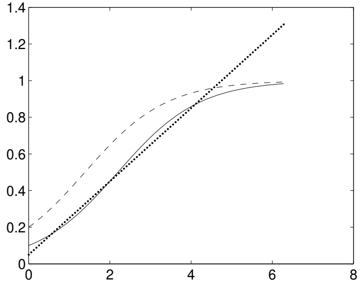


Strategy to find option price:

1. Choose parametrization of subsets of $[0, T] \times \mathbb{R}^d$
2. Choose initial guess $B_0 \subset [0, T] \times \mathbb{R}^d$
3. Update to get $B_n \to B_\infty$ and $\Psi(B_n) \to \Psi(B_\infty) = v(s_0)$

Not so easy:

1. No obvious choice, no "orthogonal bases" of subsets
2. How to pick initial guess?
3. Recall lack of continuity of hitting times in general
4. Translates to lack of continuity $B \mapsto \frac{1}{M} \sum_{i=1}^{M} Y^i_{\tau_B \wedge T}$

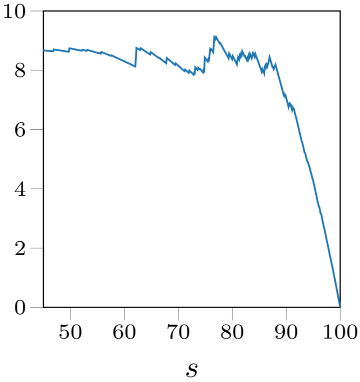


Strategy to find option price:

1. Choose parametrization of subsets of $[0,T] \times \mathbb{R}^d$
2. Choose initial guess $B_0 \subset [0,T] \times \mathbb{R}^d$
3. Update to get $B_n \to B_\infty$ and $\Psi(B_n) \to \Psi(B_\infty) = v(s_0)$

Not so easy:

1. No obvious choice, no "orthogonal bases" of subsets
2. How to pick initial guess?
3. Recall lack of continuity of hitting times in general
4. Translates to lack of continuity $B \mapsto \frac{1}{M}\sum_{i=1}^{M} Y^i_{\tau_B \wedge T}$



$X_t := \log S_t$

For $f : [0,T] \times \mathbb{R}^d \to \mathbb{R}_+$ exercise with rate $\lambda_t = f(t, X_t)$, i.e., at the first jump time of an independent Poisson process with rate λ_t. Explicitly, at time

$$\tau_f := \inf\left\{ t \geq 0 \;\middle|\; \int_0^t \lambda_u \,\mathrm{d}u \geq Z \right\}, \quad Z \sim \mathrm{Exp}(1).$$

Notation:

$$U_t := P\left(\tau_f \geq t \mid (S_u)_{u\in[0,T]}\right) = \exp\left(-\int_0^t \lambda_u \,\mathrm{d}u\right),$$

$$\phi\left(f, (S_u)_{u\in[0,T]}\right) := E\left[Y_{\tau_f \wedge T} \mid (S_u)_{u\in[0,T]}\right] = -\int_0^T Y_t \,\mathrm{d}U_t + Y_T U_T,$$

$$\psi(f) := E\left[\phi\left(f, (S_u)_{u\in[0,T]}\right)\right] = E\left[Y_{\tau_f \wedge T}\right]$$

$$X_t := \log S_t$$

For $f : [0,T] \times \mathbb{R}^d \to \mathbb{R}_+$ exercise with rate $\lambda_t = f(t, X_t)$, i.e., at the first jump time of an independent Poisson process with rate λ_t. Explicitly, at time

$$\tau_f := \inf \left\{ t \geq 0 \;\middle|\; \int_0^t \lambda_u \,\mathrm{d}u \geq Z \right\}, \quad Z \sim \mathrm{Exp}(1).$$

Notation:

$$U_t := P\left(\tau_f \geq t \mid (S_u)_{u\in[0,T]}\right) = \exp\left(-\int_0^t \lambda_u \,\mathrm{d}u\right),$$

$$\phi\left(f, (S_u)_{u\in[0,T]}\right) := E\left[Y_{\tau_f \wedge T} \mid (S_u)_{u\in[0,T]}\right] = -\int_0^T Y_t \,\mathrm{d}U_t + Y_T U_T,$$

$$\psi(f) := E\left[\phi\left(f, (S_u)_{u\in[0,T]}\right)\right] = E\left[Y_{\tau_f \wedge T}\right]$$

$$v(s_0) = \sup\left\{ \psi(f) \;\middle|\; f : [0,T] \times \mathbb{R}^d \to \mathbb{R}_+ \text{ measurable} \right\}$$

Proof.

Economical Randomized stopping rules are available to investors

Mathematical "$\leq$" Any hitting time τ_B corresponds to

$$f_B(t,x) := \begin{cases} +\infty, & (t, e^x) \in B, \\ 0, & \text{else.} \end{cases}$$

"$\geq$" Conditioning on X yields stopping times, i.e.,

$$\psi(f) = E\left[E\left[Y_{\tau_f \wedge T} \mid X\right]\right] \leq v(s_0).$$

□

$$v(s_0) = \sup \left\{ \psi(f) \;\middle|\; f : [0,T] \times \mathbb{R}^d \to \mathbb{R}_+ \text{ measurable} \right\}$$

Proof.

Economical Randomized stopping rules are available to investors

Mathematical "$\leq$" Any hitting time τ_B corresponds to

$$f_B(t,x) := \begin{cases} +\infty, & (t, e^x) \in B, \\ 0, & \text{else.} \end{cases}$$

"$\geq$" Conditioning on X yields stopping times, i.e.,

$$\psi(f) = E\left[E\left[Y_{\tau_f \wedge T} \mid X\right]\right] \leq v(s_0). \qquad \square$$

$$v(s_0) = \sup \left\{ \psi(f) \;\middle|\; f : [0,T] \times \mathbb{R}^d \to \mathbb{R}_+ \text{ measurable} \right\}$$

Proof.

Economical Randomized stopping rules are available to investors

Mathematical "$\leq$" Any hitting time τ_B corresponds to

$$f_B(t,x) := \begin{cases} +\infty, & (t, e^x) \in B, \\ 0, & \text{else.} \end{cases}$$

"$\geq$" Conditioning on X yields stopping times, i.e.,

$$\psi(f) = E\left[E\left[Y_{\tau_f \wedge T} \mid X\right]\right] \leq v(s_0). \qquad \square$$

$$v(s_0) = \sup\left\{ \psi(f) \,\middle|\, f : [0,T] \times \mathbb{R}^d \to \mathbb{R}_+ \text{ measurable} \right\}$$

Proof.

Economical Randomized stopping rules are available to investors

Mathematical "$\leq$" Any hitting time τ_B corresponds to

$$f_B(t,x) := \begin{cases} +\infty, & (t, e^x) \in B, \\ 0, & \text{else.} \end{cases}$$

"$\geq$" Conditioning on X yields stopping times, i.e.,

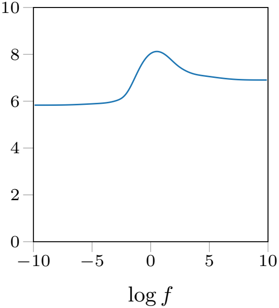
$$\psi(f) = E\left[E\left[Y_{\tau_f \wedge T} \mid X\right]\right] \leq v(s_0).$$

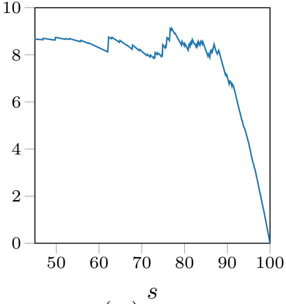
□

Pathwise smoothness

$$\left\langle \nabla_f \phi\left(f, (S_t)_{t\in[0,T]}\right), h \right\rangle = -\int_0^T Y_t \, \mathrm{d}\left\langle \nabla_f U_t, h\right\rangle + \left\langle \nabla_f U_T, h\right\rangle Y_T,$$

$$\left\langle \nabla_f U_t, h\right\rangle = -U_t \int_0^t h(u, X_u)\, \mathrm{d}u$$





a) $\frac{1}{M}\sum_{m=1}^{M} \phi^{(m)}\left(f_c, S_\cdot^m\right),\ f_c \equiv c$ **b)** $\frac{1}{M}\sum_{m=1}^{M} Y^{(m)}_{\tau_{B_s}\wedge T},\ B_s = [0,T]\times[0,s]$

Rate parametrization

Parametrization of rates by polynomials of degree $\leq k$ in (t, x)

$$F_k := \{ \, f_p(t,x) = \mathbb{1}_{y>0} \exp\left(p(t,x)\right) \mid p \in \mathcal{P}_k \, \}$$

(a) $s^* \approx K - (T-t)^{1/2}$

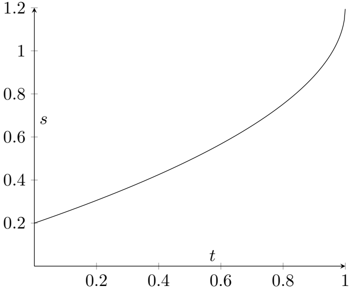
(b) $\{T - t - (K-s)^2 = 0\} \cap \{g(s) \geq 0\}$

Figure: Univariate put option, $g(s) := (K-s)^+$, $K = 1, T = 1$

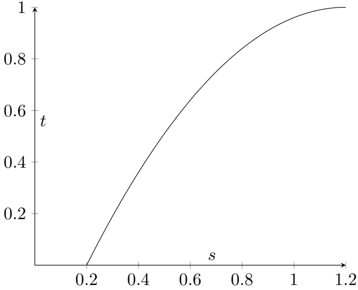
Rate parametrization

Parametrization of rates by polynomials of degree $\leq k$ in (t, x)

$$F_k := \{ f_p(t,x) = \mathbb{1}_{y>0} \exp\left(p(t,x)\right) \mid p \in \mathcal{P}_k \}$$



(a) $s^* \approx K - (T-t)^{1/2}$



(b) $\{T - t - (K-s)^2 = 0\} \cap \{g(s) \geq 0\}$

Figure: Univariate put option, $g(s) := (K-s)^+$, $K = 1, T = 1$

(i) Replace $(S_t)_{0\le t\le T}$ by discretization with $N < \infty$ time steps

(ii) Approximate expectation by MC based on M samples

(iii) Choose polynomials $(\psi_j)_{j=1}^{K}$ on $\mathbb{R}^{1+d}$ and let

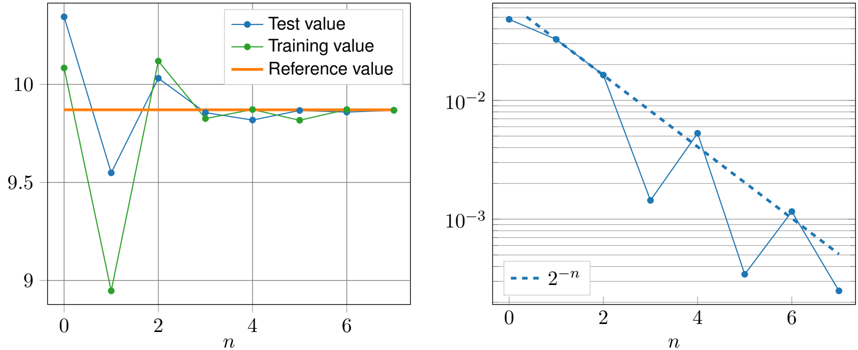
$$\mathbb{R}^K \ni \boldsymbol{c} \mapsto f_{\boldsymbol{c}} := \exp\left(\sum_{j=1}^{K} c_j \psi_j\right) \mathbb{1}_{g>0}$$

(iv) Using standard algorithms (e.g., L-BFGS-B), maximize the (discretized) surrogate function $\overline{\Psi} \colon \mathbb{R}^K \to \mathbb{R}$

$$\boldsymbol{c} \mapsto \frac{1}{M} \sum_{m=1}^{M} \left[-\int_0^T Y_t^m \,\mathrm{d}U_t^{m,\boldsymbol{c}} + Y_T^m U_T^{m,\boldsymbol{c}} \right],$$

where $U_t^{m,\boldsymbol{c}} := \exp\left(-\int_0^t \lambda_u^{m,\boldsymbol{c}} \,\mathrm{d}u\right)$ and $\lambda_t^{m,\boldsymbol{c}} := f_{\boldsymbol{c}}(t, X_t^{(m)})$

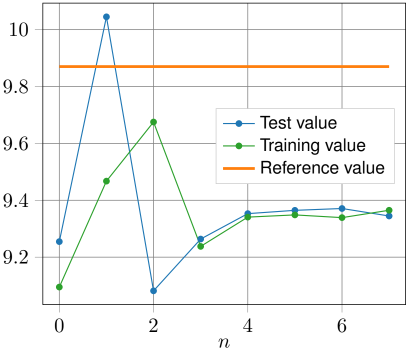
(v) Optionally, resample paths to compute option price based on $f_{\boldsymbol{c}^*}$



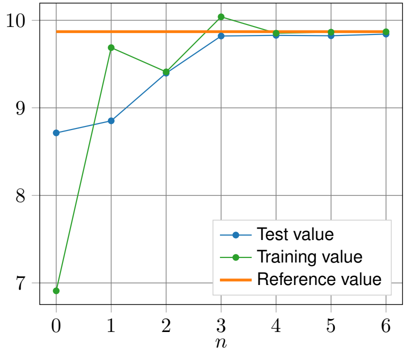
(a) Test and training prices

(b) Relative error of test price

Figure: ERO with polynomial degree $k = 2$ (in (t, x)), $M = M_n = 400 \times 4^n$ samples, $N = N_n = 2^n$ time-steps, error $\mathcal{O}(M^{-1/2} + N^{-1})$



(a) $k = 0$



(b) $k = 1$

Figure: ERO with polynomial degree $k = 0, 1$ (in (t, x)), $M = M_n = 400 \times 4^n$ samples, $N = N_n = 2^n$ time-steps

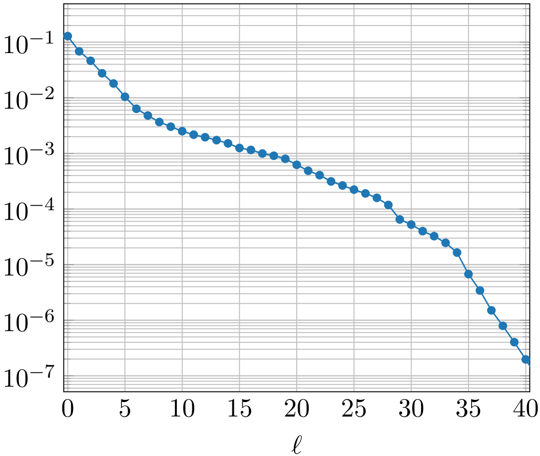
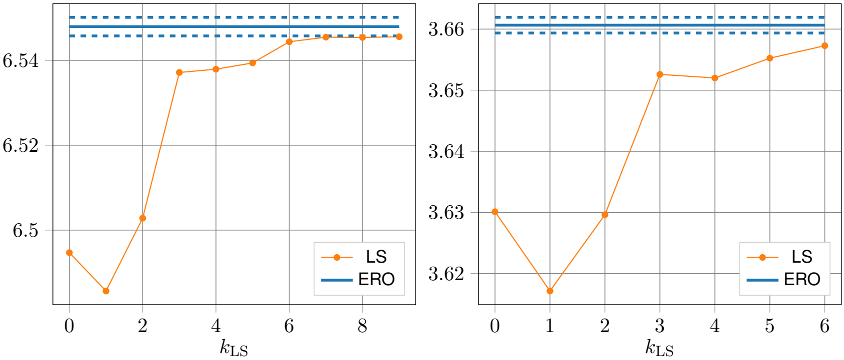


Figure: Convergence with respect to the number of iterations of L-BFGS-B $(n = 4)$. We see exponential convergence.



(a) $d = 2$, # of basis: 10 (ERO), 28 (LS)

(b) $d = 5$, # of basis: 28 (ERO), 462 (LS)

Figure: Convergence of Longstaff–Schwartz algorithm (LS) for $\{2, 5\}$-dimensional basket put options with increasing polynomial degree. Reference value computed using ERO with quadratic polynomials and 95% confidence bands (dashed).

Max call option, Black-Scholes model, $d = 2$, training

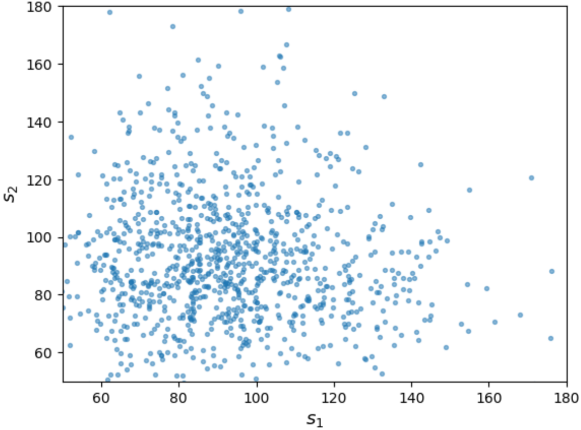


Figure: Learning exercise rates at time $t = 0.5$ for an American max call option with parametrization based on cubic polynomials. Point cloud.

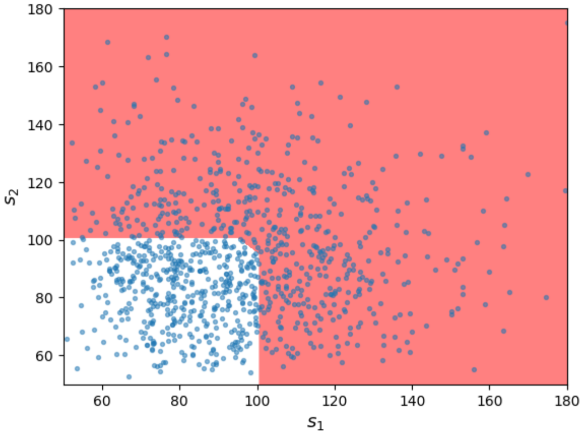


Figure: Learning exercise rates at time $t = 0.5$ for an American max call option with parametrization based on cubic polynomials. Iteration 0.

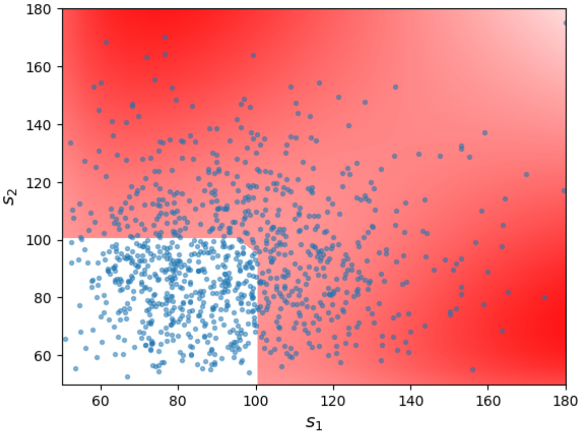


Figure: Learning exercise rates at time $t = 0.5$ for an American max call option with parametrization based on cubic polynomials. Iteration 10.

Max call option, Black-Scholes model, $d = 2$, training

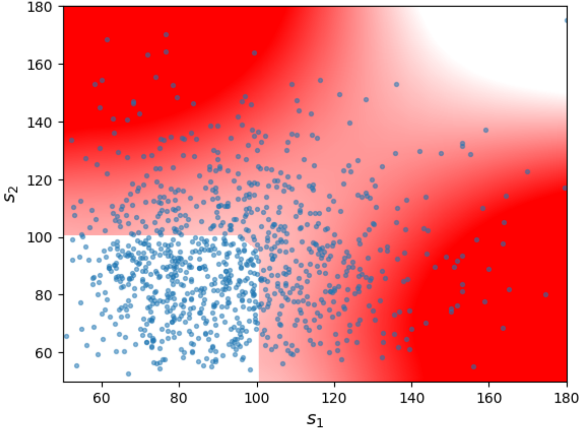


Figure: Learning exercise rates at time $t = 0.5$ for an American max call option with parametrization based on cubic polynomials. Iteration 20.

Max call option, Black-Scholes model, $d = 2$, training

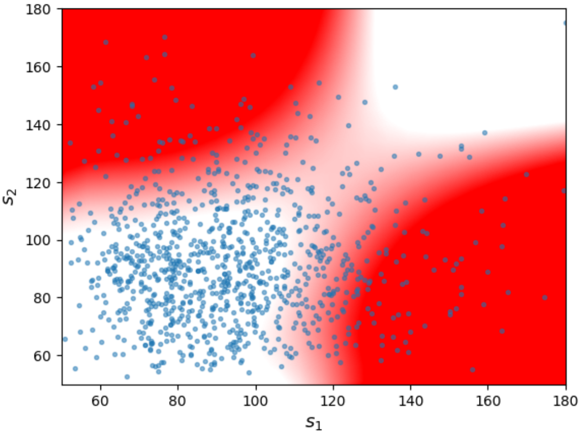


Figure: Learning exercise rates at time $t = 0.5$ for an American max call option with parametrization based on cubic polynomials. Iteration 30.

Max call option, Black-Scholes model, $d = 2$, training

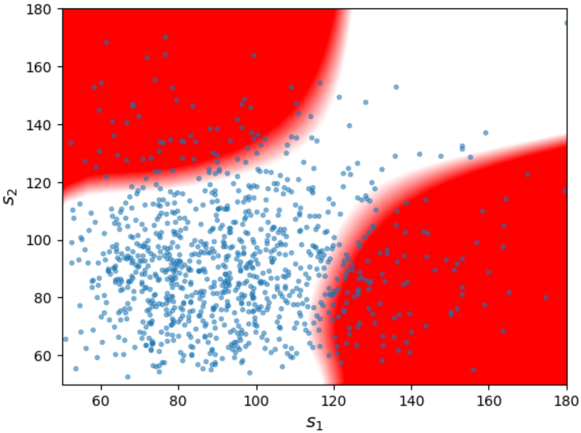


Figure: Learning exercise rates at time $t = 0.5$ for an American max call option with parametrization based on cubic polynomials. Iteration 40.

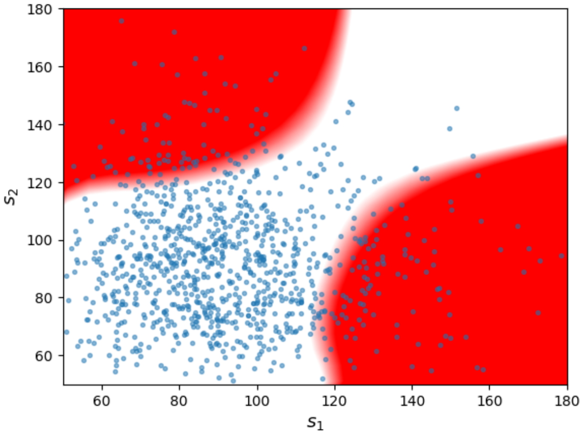


Figure: Learning exercise rates at time $t = 0.5$ for an American max call option with parametrization based on cubic polynomials. Iteration 46.

Max call option, Black-Scholes model, $d = 2$

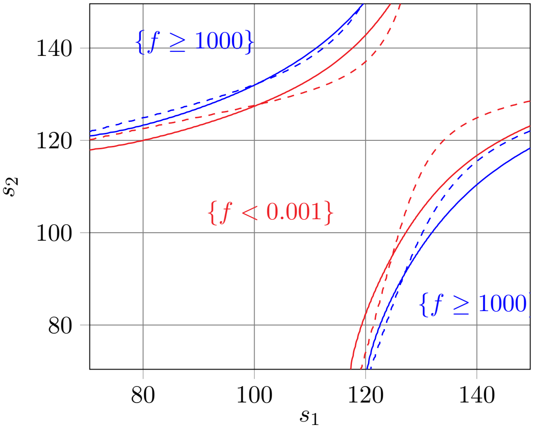


Figure: Level sets of optimal exercise rate at time $t = 0.5$ for American max call option with quadratic (dashed) and cubic (solid) polynomials. Here, first order polynomials cannot capture the shape of the exercise region.

- Rate $\lambda_t = f(t, X_t, v_t)$ for stochastic variance v_t
- Multivariate asset $S_t = (S_t^1, \ldots, S_t^d)$ driven by a joint, one-dimensional variance process v_t
- Example on the right: American put option in Heston model ($d = 1$, $K = 110$, $S_0 = 100$, $v_0 = 0.15$)

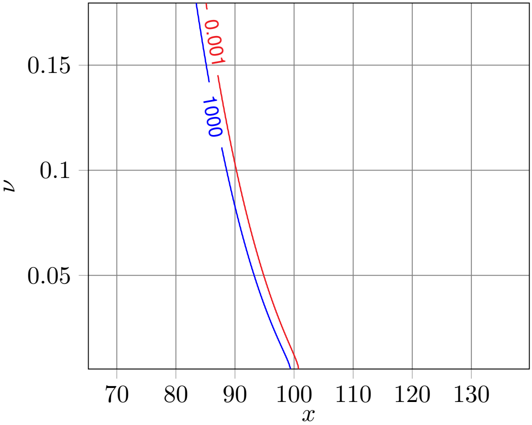
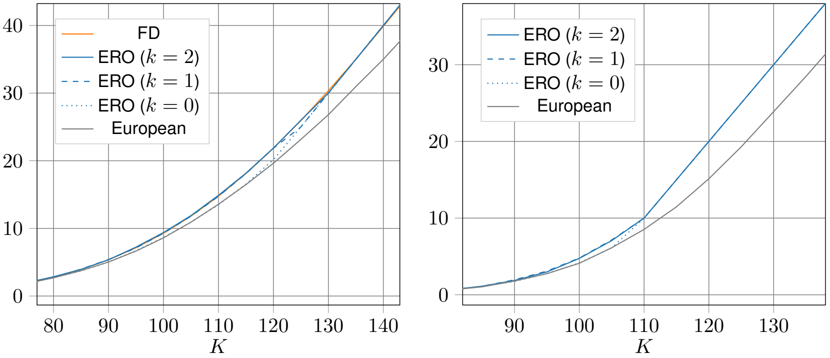


Figure: Level sets of exercise rate at time $t = 0.5$



(a) $d = 1$ **(b)** $d = 10$

Figure: Convergence of ERO in the polynomial degree for American put options in multivariate Heston models

$$dS_t = S_t \sqrt{v_t} dZ_t, \quad S_0 = s_0$$
$$v_t = \xi_0 \mathcal{E}\left(\eta \widehat{W}_t\right), \quad \widehat{W}_t = \sqrt{2H} \int_0^t (t-s)^{H-1/2} dW_s$$

- $H \ll 1/2$ is typically used
- Not a Markov process!

Extended state space

For $J \geq 0$ choose $\lambda_t = f(t, \mathbf{X}_t)$ with

$$\mathbf{X}_t := \left(\log S_t, \log S_{t-\Delta_1}, \ldots, \log S_{t-\Delta_J}, v_t, v_{t-\Delta_1}, \ldots, v_{t-\Delta_J}\right).$$

$$dS_t = S_t\sqrt{v_t}dZ_t, \quad S_0 = s_0$$
$$v_t = \xi_0 \mathcal{E}\left(\eta \widehat{W}_t\right), \quad \widehat{W}_t = \sqrt{2H}\int_0^t (t-s)^{H-1/2} dW_s$$

- $H \ll 1/2$ is typically used
- Not a Markov process!

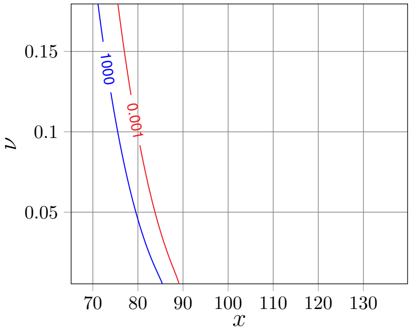
Extended state space

For $J \geq 0$ choose $\lambda_t = f(t, \mathbf{X}_t)$ with

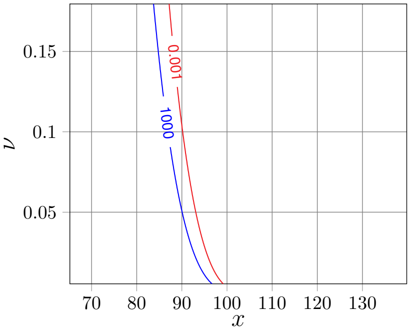
$$\mathbf{X}_t := \left(\log S_t, \log S_{t-\Delta_1}, \ldots, \log S_{t-\Delta_J}, v_t, v_{t-\Delta_1}, \ldots, v_{t-\Delta_J}\right).$$

		K							
		70	80	90	100	110	120	130	140
Euro.		1.83	3.13	5.06	7.98	12.21	17.99	25.35	33.88
J	0	1.88	3.23	5.32	8.51	13.24	20	30	40
	1	1.88	3.23	5.31	8.50	13.22	20	30	40
	3	1.88	3.21	5.31	8.50	13.22	20	30	40
	7	1.88	3.22	5.30	8.50	13.23	20	30	40

Table: Prices of American put option in the rough Bergomi model, $S_0 = 100$, $v_0 = 0.09$, $H = 0.07$, $\eta = 1.9$, $\rho = -0.9$.



(a) $K = 100$



(b) $K = 110$

Figure: Level sets of exercise rates at $t = 0.5$ with $J = 0$

Consider a *Bermudan option*, with stopping times restricted to a finite set of times, w.l.o.g. $\{0, 1, \dots, J\}$.

Randomized exercise region optimization

$$v(0, S_0) = \sup_{(h_1,\dots,h_J)\in\mathcal{H}^J} E\left[\sum_{j=0}^{J} Y_j h_j(X_j) \prod_{\ell=0}^{j-1} (1 - h_\ell(X_\ell))\right],$$

where $\mathcal{H}$ denotes the space of measurable functions taking values in $[0, 1]$.

- Obvious adaptation of ERO to Bermudan options
- Implementation: Replace $\mathcal{H}$ by a parameterized, finite-dimensional subspace $\widehat{\mathcal{H}}$

Example (DNN, Becker, Cheridito, Jentzen, Welti '19)

Here, $\widehat{\mathcal{H}}$ is the space of deep neural networks of a given architecture.

Consider a *Bermudan option*, with stopping times restricted to a finite set of times, w.l.o.g. $\{0, 1, \ldots, J\}$.

Randomized exercise region optimization

$$v(0, S_0) = \sup_{(h_1,\ldots,h_J)\in\mathcal{H}^J} E\left[\sum_{j=0}^{J} Y_j h_j(X_j) \prod_{\ell=0}^{j-1} (1 - h_\ell(X_\ell))\right],$$

where $\mathcal{H}$ denotes the space of measurable functions taking values in $[0, 1]$.

- Obvious adaptation of ERO to Bermudan options
- Implementation: Replace $\mathcal{H}$ by a parameterized, finite-dimensional subspace $\widehat{\mathcal{H}}$

Example (DNN, Becker, Cheridito, Jentzen, Welti '19)

Here, $\widehat{\mathcal{H}}$ is the space of deep neural networks of a given architecture.

- Let $N(\delta)$ denote the covering number of $\widehat{\mathcal{H}}$ w.r.t. $L^2(X)$, i.e., the number of balls of radius δ needed to cover $\widehat{\mathcal{H}}$. Assume that

$$N(\delta) \leq A\delta^{-\rho}.$$

- Assume that the continuation value C_j is close to Y_j in the sense that

$$P\left(|C_j(X_j) - Y_j| \leq \delta\right) \leq B\delta^{\alpha}.$$

Theorem

Let $\overline{v}^M$ denote the Monte Carlo approximation of $v(0, S_0)$ after re-sampling. Then, with probability at least $1-\delta$,

$$0 \leq v(0, S_0) - \overline{v}^M \leq C\left(\frac{\log(1/\delta)^2}{M}\right)^{\frac{1+\alpha}{2+\alpha(1+\nu)}},$$

where $\nu := \frac{2(1+\alpha)}{2+\alpha(1+\rho/2)}$.

References

- L. Andersen: *A simple approach to the pricing of Bermudan swaptions in the multi-factor LIBOR market model*, J. Comp. Fin., 1999.
- C. Bayer, P. K. Friz, J. Gatheral: *Pricing under rough volatility*, Quant. Fin. 2016.
- S. Becker, P. Cheridito, A. Jentzen, T. Welti: *Solving high-dimensional optimal stopping problems using deep learning*, arXiv, 2019.
- C. Costantini, E. Gobet, N. El Karoui: *Boundary sensitivities for diffusion processes in time dependent domains*, Appl. Math. Optim., 2006.
- D. Grant, G. Vora, D. Weeks: *Path-dependent options: Extending the Monte Carlo simulation approach*, Management Science, 1997.
- F. A. Longstaff, E. S. Schwartz: *Valuing American options by simulations: A simple least-squares approach*, Rev. Fin. Studies, 2001.
- D. Z. Zanger: *Convergence of least-squares Monte Carlo algorithm for American option pricing with dependent sample data*, Math. Fin., 2018.