# State-of-the-art option pricing by simulation

## PRMIA Munich Chapter Meeting

Christian Bender

TU Braunschweig

June 2008

# Contents

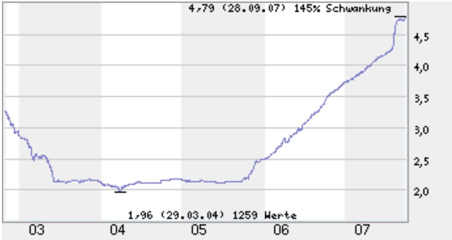
# Contents

# Example 1: Bermuda Swaption

- Swap: At some fixed time points $\{T_0, \ldots, T_{\mathcal{I}}\}$, say quarterly, there are the following payments

  Bank 1 pays coupons according to a fixed rate $\theta$;

  Bank 2 pays coupons according to the Euribor.

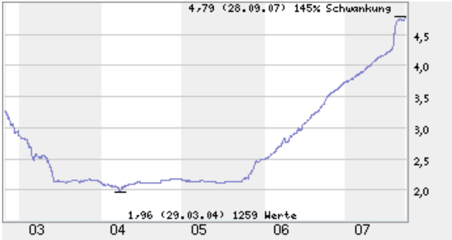


# Example 1: Bermuda Swaption

- Swap: At some fixed time points $\{T_0, \ldots, T_\mathcal{I}\}$, say quarterly, there are the following payments

  Bank 1 pays coupons according to a fixed rate $\theta$;

  Bank 2 pays coupons according to the Euribor.




- Bermudan Swaption: Bank 2 has the right to cancel the contract at one of the payment dates of its choice.

# Example 2: Cancelable snowball swap

- A cancelable snowball swap is an exotic swap: the Euribor is swaped at payment dates (e.g. semi-annually) against a complexly structured coupon, the snowball coupon.

  The swap can be canceled (terminated) at any payment date to be chosen by the payer of the snowball coupon.

# Example 2: Cancelable snowball swap

Notation:

- Payment dates $\mathcal{E} = \{T_0, \ldots, T_{\mathcal{I}}\}$
- $L_i(t)$: The interest rate at time $t$ for a loan over the period between $T_i$ and $T_{i+1}$ where $t \leq T_i$.

# Example 2: Cancelable snowball swap

Notation:

- Payment dates $\mathcal{E} = \{T_0, \ldots, T_{\mathcal{I}}\}$
- $L_i(t)$: The interest rate at time $t$ for a loan over the period between $T_i$ and $T_{i+1}$ where $t \leq T_i$.

Specification of the coupons:

- Bank A pays the spot-Libor in arrears, i.e. at time $T_i$:

$$\text{Nominal} \times L_{i-1}(T_{i-1})(T_i - T_{i-1}).$$

- Bank B pays at time $T_i$ the snowball coupon:

$$\text{Nominal} \times K_{i-1}(T_i - T_{i-1})$$

# Example 2: Cancelable snowball swap

Notation:

- Payment dates $\mathcal{E} = \{T_0, \ldots, T_{\mathcal{I}}\}$
- $L_i(t)$: The interest rate at time $t$ for a loan over the period between $T_i$ and $T_{i+1}$ where $t \leq T_i$.

Specification of the coupons:

- Bank A pays the spot-Libor in arrears, i.e. at time $T_i$:

$$\text{Nominal} \times L_{i-1}(T_{i-1})(T_i - T_{i-1}).$$

- Bank B pays at time $T_i$ the snowball coupon:

$$\text{Nominal} \times K_{i-1}(T_i - T_{i-1})$$

where

$$\begin{aligned} K_i &:= I, \quad i = 0, 1, \\ K_i &:= (K_{i-1} + A_i - L_i(T_i))^+ \quad i = 2, \ldots, \mathcal{I} - 1. \end{aligned}$$

and $I$, $A_i$ are specified in the contract.

# The pricing problem

Problem: How to compute the fair price of such Bermudan products numerically?

- First Step: Choice of the model.
- Second Step: Calibration of the model.
- Third Step: Choice of an appropriate pricing algorithm.

# The pricing problem

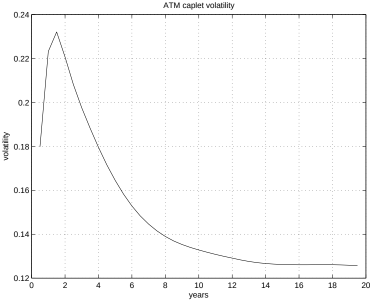
Problem: How to compute the fair price of such Bermudan products numerically?

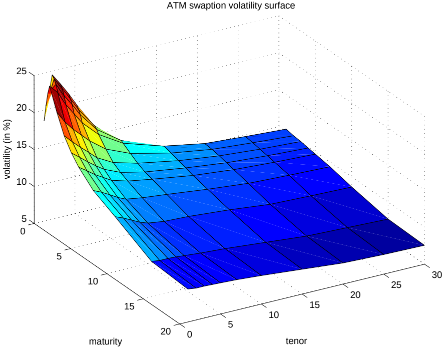
- First Step: Choice of the model.
- Second Step: Calibration of the model.
- Third Step: Choice of an appropriate pricing algorithm.

# Choosing a model for the Euribor

One- or two-factor short rate models, e.g. the Hull-White model:

- Model type: determined by a SDE driven by a one- or two-dimensional Brownian motion.
- Advantage: Bermudan products can be priced by straightforward implementation of trinomial trees.
- Disadvantage: Model cannot capture the term structure of caplet and swaption volatilities.


ATM caplet volatility
volatility
years
0.24
0.22
0.2
0.18
0.16
0.14
0.12
0
2
4
6
8
10
12
14
16
18
20


ATM swaption volatility surface
volatility (in %)
maturity
tenor
25
20
15
10
5
0
5
10
15
20
0
5
10
15
20
25
30

# Choosing a model for the Euribor

LIBOR market model

- Model type: determined by a high-dimensional system of SDEs driven by a possibly high-dimensional Brownian motion.
- Advantage: Reasonable fit to caplet and swaption prices is possible.
- Disadvantage: Pricing by tree methods is impossible due to the curse of dimensionality.

# Modeling error vs. numerical error

Choice between

- a (typically low-dimensional) model, in which Bermudan products can be priced with high accuracy, but which poorly fits the observed data.
- a (typically high-dimensional) model, which reasonably fits the observed data, but requires more sophisticated pricing tools.

# Contents

# An abstract framework for Bermudan products

Assumption: Arbitrage-free market of tradable securities, which is already calibrated to liquidly traded products.

$\longrightarrow$ We have fixed a pricing measure $Q$ connected to some discount factor $\mathcal{N}$.

# An abstract framework for Bermudan products

Assumption: Arbitrage-free market of tradable securities, which is already calibrated to liquidly traded products.

$\rightarrow$ We have fixed a pricing measure $Q$ connected to some discount factor $\mathcal{N}$.

**Definition**

A *Bermudan option* consists of a finite set of time points $\mathcal{E} = \{T_0, \ldots, T_{\mathcal{I}}\}$ and a cashflow $\mathcal{Z}(T_i)$.

- Interpretation: The holder of the Bermudan option is entitled to choose one time point out of the set $\mathcal{E}$, at which she exercises the cash-flow $\mathcal{Z}$, i.e. she receives e.g. $\mathcal{Z}(T_i)$.

# Connection to optimal stopping

- Consider the discounted cashflow $Z(i) = \mathcal{Z}(T_i)/\mathcal{N}(T_i)$.
- Assume w.l.o.g. $\mathcal{N}(0) = 1$.
- The fair price of the Bermudan product is determined by the optimal choice to exercise the cash-flow

$$\sup_{\tau \in \mathcal{T}_{0,\mathcal{I}}} E^Q[Z(\tau)]$$

  where $\mathcal{T}_{0,\mathcal{I}}$ is the set of $\{0, \ldots, \mathcal{I}\}$-valued non-anticipating random times
- From now on: All (conditional) expectations are taken under $Q$.

# Backward dynamic programming

- Idea: Find an optimal exercise time $\tau^*(i)$ provided the option has not been exercised before time $i$.

# Backward dynamic programming

- Idea: Find an optimal exercise time $\tau^*(i)$ provided the option has not been exercised before time $i$.
- At terminal time:

$$\tau^*(\mathcal{I}) = \mathcal{I}$$

(because no other time points are left).

At time $i$: Exercise immediately, if and only $Z(i)$ is at least as large as what you expect to get by waiting until time $i+1$ and proceeding optimally from that time on:

$$\tau^*(i) = \begin{cases} i, & Z(i) \geq E[Z(\tau^*(i+1))|\mathcal{F}_i] \\ \tau^*(i+1), & \text{otherwise} \end{cases}$$

# Backward dynamic programming

- Idea: Find an optimal exercise time $\tau^*(i)$ provided the option has not been exercised before time $i$.
- At terminal time:

$$\tau^*(\mathcal{I}) = \mathcal{I}$$

(because no other time points are left).

At time $i$: Exercise immediately, if and only $Z(i)$ is at least as large as what you expect to get by waiting until time $i+1$ and proceeding optimally from that time on:

$$\tau^*(i) = \begin{cases} i, & Z(i) \geq E[Z(\tau^*(i+1))|\mathcal{F}_i] \\ \tau^*(i+1), & \text{otherwise} \end{cases}$$

- Then $\tau^*(0)$ is an optimal exercise time and $E[Z(\tau^*(0))]$ is the fair price of the Bermudan option.

# Why Monte-Carlo is ill-suited (I)

General idea of Monte-Carlo simulation:

- Starting from today's prices of the underlying market, simulate future scenarios of the market (under the pricing measure $Q$);
- Approximate expectations under $Q$ by averaging over the simulated scenarios.

# Why Monte-Carlo is ill-suited (I)

General idea of Monte-Carlo simulation:

- Starting from today's prices of the underlying market, simulate future scenarios of the market (under the pricing measure $Q$);
- Approximate expectations under $Q$ by averaging over the simulated scenarios.

Problem:

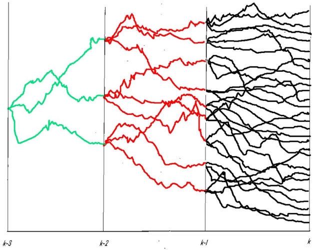
- Simulation is genuinely directed forwardly in time;
- The dynamic program is directed backwardly in time.

# Why Monte-Carlo is ill-suited (II)

- Problem: In each step of the backward dynamic program an expectation must be calculated which depends on the exercise time from the previous time step.

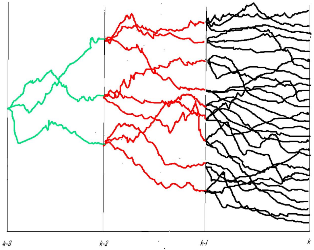
# Why Monte-Carlo is ill-suited (II)

- Problem: In each step of the backward dynamic program an expectation must be calculated which depends on the exercise time from the previous time step.
- Naive approach: Average over simulated paths (plain Monte Carlo) as suggested by the Law of Large Numbers.



# Why Monte-Carlo is ill-suited (II)

- Problem: In each step of the backward dynamic program an expectation must be calculated which depends on the exercise time from the previous time step.
- Naive approach: Average over simulated paths (plain Monte Carlo) as suggested by the Law of Large Numbers.




Infeasible: Computational cost explodes rapidly with the number of exercise dates.

# Contents

# Lower bounds by simulation: General ideas

- Trivial: Any sub-optimal exercise time $\sigma$ induces a lower bound by
$$E[Z(\sigma)].$$

- If a simulation mechanism is available, simulate $L$ independent copies of $Z(\sigma)$ and calculate the expectation by averaging.

  $\longrightarrow$ estimator which is biased low.

- Many algorithms have been proposed to find a 'good' approximative strategy $\sigma$.

# The Longstaff-Schwartz algorithm

Basic idea: approximate all (conditional) expectations in the backward dynamic program by least-squares Monte-Carlo.

- Markovian setting: $\mathbb{R}^D$-valued Markov process $X(i)$ such that $Z(i) = h(i, X(i))$.
- Then: $E[f(X(j))|\mathcal{F}_i] = E[f(X(j))|X(i)] = u(X(i))$.
- Aim: Estimate the function $u$ as a linear combination of basis functions with the coefficients estimated by simulation.

# Conditional expectations via least squares Monte Carlo

Pseudo-Algorithm:

1. Choose a vector of basis functions

$$\psi(i,x) = (\psi_1(i,x), \ldots, \psi_K(i,x));\ x \in \mathbb{R}^D;$$

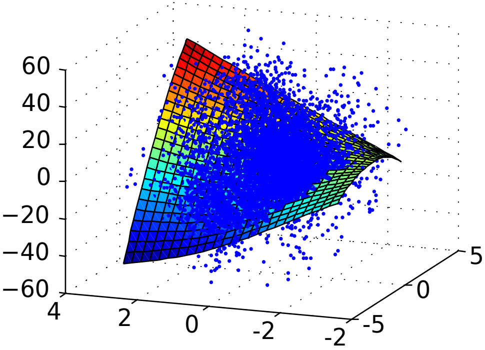
2. Simulate $L$ independent copies $X_\lambda(i),\ \lambda = 1, \ldots, L$ of $X$;
3. Solve the least squares problem

$$\begin{aligned} a(i,j;f) &= \arg\min_{a\in\mathbb{R}^K} \frac{1}{L}\sum_{\lambda=1}^{L} (f(X_\lambda(j)) - \psi(i, X_\lambda(i))a)^2 \\ &\approx \arg\min_{a\in\mathbb{R}^K} E\left[(f(X(j)) - \psi(i, X(i))a)^2\right]; \end{aligned}$$

4. Define, as estimator for $E[f(X(j))|\mathcal{F}_i] = E[f(X(j))|X(i)]$,

$$\hat{E}[f(X(j))|X(i)] = \psi(i, X(i))a(i,j;f).$$

# Conditional expectations via least squares Monte Carlo



# Numerical results for Bermudan swaption

- Exercise dates annually over 10 years;
- Setting: 3M-LIBOR market model;
- Model driven by $D$-dim. Brownian motion;
- Basis: low-order monomials on the cashflow; approximations of the price for European swaptions.

| | D | LS-Lower Bound $Y_0$ | K&S Price Interval |
|---|---|---|---|
| ITM | 1 | 1108.8±1.41 | [1108.9±2.4, 1109.4±0.7] |
| | 2 | 1101.6±1.53 | [1100.5±2.4, 1103.7±0.7] |
| | 10 | 1096.4±1.61 | [1096.9±2.1, 1098.1±0.6] |
| OTM | 1 | 121.0±0.71 | [121.0±0.6, 121.3±0.4] |
| | 2 | 113.3±0.75 | [113.8±0.5, 114.9±0.4] |
| | 10 | 100.1±0.83 | [100.7±0.4, 101.5±0.3] |

# Numerical results for the cancelable snowball swap

- Exercise dates semiannually over 10 years;
- Setting: 6M-LIBOR market model;
- Model driven by 19-dim. Brownian motion;
- Basis: low order monomials on explanatory variables, here: snowball coupon, spot LIBOR rate, long swap rate; cp. Piterbarg.

# Numerical results for the cancelable snowball swap

- Exercise dates semiannually over 10 years;
- Setting: 6M-LIBOR market model;
- Model driven by 19-dim. Brownian motion;
- Basis: low order monomials on explanatory variables, here: snowball coupon, spot LIBOR rate, long swap rate; cp. Piterbarg.

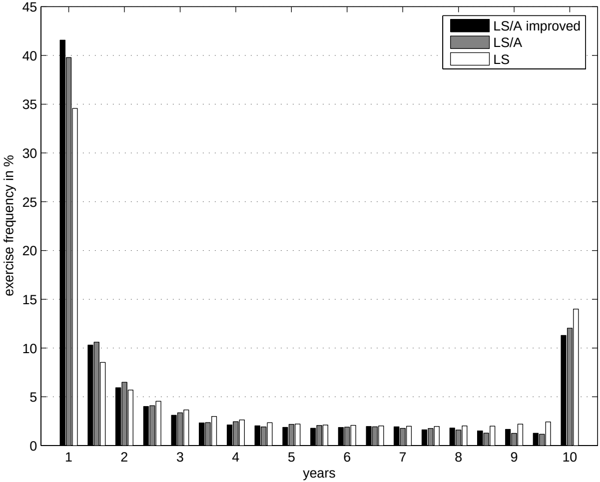
LS-lower bound: 77.54 (bp) $\pm$ 0.36
Reference price interval: $[106.47 \pm 0.84, 110.22 \pm 0.55]$
(B./Kolodko/Schoenmakers)

Problems:

- LS-lower price bound is significantly off;
- Not clear, how to tailor the basis to the problem.

# Numerical results for the cancelable snowball swap

Exercise profile:



# Discussion of the Longstaff-Schwartz algorithm

Advantages

- Easy to implement and quite fast;
- Estimator is biased low (since it is based on sub-optimal policies);
- Convergence to the Bermudan price, when the basis exhausts a complete system and the simulated paths tend to infinity (see Clement, Lamberton & Protter; Egloff);
- Simple basis functions (low order polynomials) and moderate sample size often yield very good lower bounds.

# Discussion of the Longstaff-Schwartz algorithm

## Disadvantages

- The interplay of several error sources is difficult to handle:
  - Choice of basis
  - Simulation error
  - Error propagation backwards through time.
- Theoretical convergence may be slow in specific examples:

  Exponential growth in the samples when the number of basis functions increases (Glasserman & Yu)
- Simple choice of basis may yield poor lower bounds in some difficult situations.

# Discussion of the Longstaff-Schwartz algorithm

Disadvantages

- The interplay of several error sources is difficult to handle:
  - Choice of basis
  - Simulation error
  - Error propagation backwards through time.
- Theoretical convergence may be slow in specific examples:

  Exponential growth in the samples when the number of basis functions increases (Glasserman & Yu)
- Simple choice of basis may yield poor lower bounds in some difficult situations.

Questions:

- How to improve upon the LS-lower bounds?
- How to assess the quality of the lower bound?

# Policy improvement (B./Kolodko/Schoenmakers)

- Denote by $(\tau(0), \ldots, \tau(\mathcal{I}))$ the exercise times constructed by the LS-algorithm.
- Basic idea: Compare
  1. the reward from immediate exercise at time $i$;
  2. the highest expected reward by choosing one of the remaining LS-exercise times $\tau(j)$, $j \geq i + 1$.

  Hence,

$$\widetilde{\tau}(i) := \inf \left\{ j : \; i \leq j \leq \mathcal{I}, \; Z(j) \geq \max_{j+1 \leq p \leq \mathcal{I}} E\left[ Z(\tau(p)) | \mathcal{F}_j \right] \right\}.$$

# Policy improvement (B./Kolodko/Schoenmakers)

- Denote by $(\tau(0), \dots, \tau(\mathcal{I}))$ the exercise times constructed by the LS-algorithm.
- Basic idea: Compare
  1. the reward from immediate exercise at time $i$;
  2. the highest expected reward by choosing one of the remaining LS-exercise times $\tau(j),\ j \geq i+1$.

  Hence,

$$\widetilde{\tau}(i) := \inf\left\{ j :\ i \leq j \leq \mathcal{I},\ Z(j) \geq \max_{j+1 \leq p \leq \mathcal{I}} E\left[Z(\tau(p)) | \mathcal{F}_j\right] \right\}.$$

- Result: The lower bound based on $\tilde{\tau}(0)$ is always better than the LS-lower bound.

# Policy improvement: Algorithm

Markovian setting: $Z(i) = h(i, X(i))$

1. Suppose $(\tau(0), \ldots, \tau(\mathcal{I}))$ are constructed by the LS-algorithm.
2. Simulate $L$ outer samples ${}_\lambda X$ of $X$
3. Given $i$ and ${}_\lambda X$, estimate e.g.

$$E\left[Z(\tau(p))|\mathcal{F}_j\right] = E[h(\tau(p), X(\tau(p; X)))|X(i)]$$

   by plain Monte Carlo, averaging over inner samples which are sampled according to the conditional law given $X(i) = {}_\lambda X(i)$.
4. Find $L$ approximations of $\tilde{\tau}(i)$ by approximating the exercise criterion

$$\tilde{\tau}(i) = i \;\Leftrightarrow\; Z(i) \geq \max_{i+1 \leq p \leq \mathcal{I}} E\left[Z(\tau(p))|\mathcal{F}_i\right]$$

   accordingly.
5. Average over the outer samples to approximate $E[Z(\tilde{\tau}(0))]$.

# Discussion of the improvement algorithm

Advantage:

- Always yields tighter lower bounds than the LS-algorithm, see the snowball example.

Disadvantages:

- One layer of nested simulation is required.
- Application of the plain algorithm to serious problems (e.g. the snowball example) may require long computing times (several hours). However, efficient variance reduction techniques are available.

# Contents

# Dual upper bounds (Rogers; Haugh and Kogan)

Rogers and Haugh & Kogan suggest:

- Start with some martingale $M$ (fair game) such that $M(0) = 0$.
- Define

$$Y_{up}(i; M) = M(i) + E[\max_{i \le j \le \mathcal{I}} (Z(j) - M(j)) \,|\mathcal{F}_i].$$

  Then $Y_{up}(0; M)$ is an upper bound for the Bermudan price.
- Simulate the upper bound $Y_{up}(0; M)$ by plain Monte Carlo

$$Y_{up}(0; M) \approx \frac{1}{L} \sum_{\lambda=1}^{L} \max_{0 \le j \le \mathcal{I}} ( {}_\lambda Z(j) - {}_\lambda M(j))$$

  to get an estimator which is biased high.

Question: How to choose the martingale?

# Upper bounds from lower bounds

- Given exercise times $\tau = (\tau(0), \dots, \tau(\mathcal{I}))$ define

$$Y_{low}(i; \tau) = E[Z(\tau(i))|\mathcal{F}_i].$$

(Expected gain when employing strategy $\tau$)

- Consider the martingale part from the Doob-decomposition,

$$M(i+1; \tau) - M(i; \tau) = Y_{low}(i+1; \tau) - E[Y_{low}(i+1; \tau)|\mathcal{F}_i].$$

# Upper bounds from lower bounds

- Given exercise times $\tau = (\tau(0), \ldots, \tau(\mathcal{I}))$ define

$$Y_{low}(i;\tau) = E[Z(\tau(i))|\mathcal{F}_i].$$

(Expected gain when employing strategy $\tau$)

- Consider the martingale part from the Doob-decomposition,

$$M(i+1;\tau) - M(i;\tau) = Y_{low}(i+1;\tau) - E[Y_{low}(i+1;\tau)|\mathcal{F}_i].$$

- The duality gap of the strategy $\tau$ is

$$\Delta\tau = Y_{up}(0; M(\cdot,\tau)) - Y_{low}(0;\tau).$$

- For the optimal strategy $\tau^*$ we have (Rogers; Haugh & Kogan)

$$\Delta\tau^* = 0.$$

# Estimating the Doob martingale: problems

- Procedure requires to estimate

$$M(i+1;\tau) - M(i;\tau) = Y_{low}(i+1;\tau) - E[Y_{low}(i+1;\tau)|\mathcal{F}_i].$$

- Estimating the conditional expectation on the right hand side typically destroys the martingale property of the estimator $\hat{M}(\cdot;\tau)$.

- Hence, $Y_{up}(0;\hat{M}(\cdot;\tau))$ may fail to be an upper bound.

# The Andersen-Broadie algorithm

Markovian setting: $Z(i) = h(i, X(i))$

1. Compute the exercise times $\tau(i, X)$ by the Longstaff-Schwartz algorithm;
2. Simulate $L$ outer samples ${}_\lambda X$ of $X$
3. Given $i$ and ${}_\lambda X$, estimate e.g.

$$E[Y_{low}(i+1;\tau)|\mathcal{F}_i] = E[h(\tau(i+1), X(\tau(i+1;X)))|X(i)]$$

   by plain Monte Carlo, averaging over inner samples which are sampled according to the conditional law given $X(i) = {}_\lambda X(i)$.
4. This yields $L$ samples ${}_\lambda \hat{M}(i;\tau)$ estimating $M(i;\tau)$
5. Define

$$Y_{up}^{AB} = \frac{1}{L}\sum_{\lambda=1}^{L} \max_{0\leq j\leq \mathcal{I}} ({}_\lambda Z(j) - {}_\lambda \hat{M}(j;\tau)).$$

# Discussion of the Andersen-Broadie algorithm

Advantages:

- $Y_{up}^{AB}$ is biased high, (although ${}_{\lambda}\hat{M}(i;\tau)$ fail to be martingales in general).

  Reason: Use of plain Monte Carlo and convexity of the max-operator.

- Converges to $Y_{up}(0; M(\cdot;\tau))$ as the number of inner and outer simulations increases.

Disadvantage:

- One layer of nested simulation is required.

Note: The reference upper bounds in the numerical examples were computed this way.

# Fast upper bounds (Belomestny/B./Schoenmakers)

Aim: Find an estimator $\hat{M}$ for the martingale

$$M(i+1;\tau) - M(i;\tau) = Y_{low}(i+1;\tau) - E[Y_{low}(i+1;\tau)|\mathcal{F}_i].$$

such that

1. $\hat{M}$ is a martingale;
2. No need for nested simulations;

# Fast upper bounds (Belomestny/B./Schoenmakers)

Aim: Find an estimator $\hat{M}$ for the martingale

$$M(i+1;\tau) - M(i;\tau) = Y_{low}(i+1;\tau) - E[Y_{low}(i+1;\tau)|\mathcal{F}_i].$$

such that

1. $\hat{M}$ is a martingale;
2. No need for nested simulations;

Framework: $Z(i) = h(T_i, X(T_i))$, where

$$dX(t) = a(t, X(t))dt + b(t, X(t))dW(t), \quad X_0 = x,$$

- $W$ is a $\mathcal{D}$-dim. Brownian motion on $[0, T]$;
- the coefficient functions $a, b$ are Lipschitz in space and $1/2$-Hölder in time;
- $X$ is $D$-dimensional.

# Fast upper bounds: Idea

Idea:

- Thanks to the martingale representation theorem there is an adapted process $U$ such that

$$M(i+1;\tau) - M(i;\tau) = \int_{T_i}^{T_{i+1}} U(s)dW(s).$$

- Given a partition $\pi \supset \mathcal{E}$ of $[0, T]$, find a non-anticipating estimator $U^\pi$ for $U$, and consider the martingale

$$M^\pi(i) = \sum_{t_j \in \pi;\ t_j < T_i} U^\pi(t_j)(W(t_{j+1}) - W(t_j)).$$

# Algorithm

- Compute the exercise family $\tau(i, X)$ utilizing the Longstaff-Schwartz algorithm;
- Estimate the martingale integrand

$$E\left[\frac{W_d(t_{j+1}) - W_d(t_j)}{t_{j+1} - t_j} h(\tau(i), X(\tau(i)))\middle| X(t_j)\right]; \quad T_{i-1} \leq t_j < T_i.$$

  via least-squares Monte-Carlo;
- Use this expression as estimator $\hat{U}_d^\pi(t_j, X)$ for the martingale integrand and the associated estimator $\hat{M}^\pi(i, X)$ for the martingale $M(i; \hat{\tau})$.
- Simulate $M$ new copies ${}_\mu X$ of $X$ and estimate the Bermudan price by

$$\hat{Y}_{up}(\hat{M}^\pi) = \frac{1}{M} \sum_{\mu=1}^{M} \max_{0 \leq j \leq \mathcal{I}} (h(j, {}_\mu X_j) - \hat{M}^\pi(j, {}_\mu X)).$$

# Numerical results for the Bermudan swaption

- 'Fast' upper bounds calculated with roughly 100 times less simulated paths than Andersen/Broadie upper bounds.
- Basis for upper bounds: 3 basis functions derived from approximations of the delta for European swaptions.

| | D | Lower Bound $Y_0$ | Upper Bound $Y_{up}(\widehat{M}^{\pi})$ | K&S Price Interval |
|---|---|---|---|---|
| ITM | 1 | 1108.8±1.41 | 1109.6±0.86 | [1108.9±2.4, 1109.4±0.7] |
| | 2 | 1101.6±1.53 | 1104.7±0.91 | [1100.5±2.4, 1103.7±0.7] |
| | 10 | 1096.4±1.61 | 1103.2±0.98 | [1096.9±2.1, 1098.1±0.6] |
| OTM | 1 | 121.0±0.71 | 122.4±0.87 | [121.0±0.6, 121.3±0.4] |
| | 2 | 113.3±0.75 | 115.2±0.89 | [113.8±0.5, 114.9±0.4] |
| | 10 | 100.1±0.83 | 103.4±0.96 | [100.7±0.4, 101.5±0.3] |

# Discussion of the algorithm

Advantages

- Fast and easy to implement;
- Converges to $Y_{up}(0; M(\cdot; \tau))$, when the mesh of the time grid decreases, the basis exhausts a complete system and the simulated paths tend to infinity.

Disadvantages

- The interplay of several error sources is difficult to handle;
- Quality of the upper bounds depends on the choice of basis more heavily than for the lower bounds.

# References


- Glasserman, P. (2004). Monte Carlo methods in financial engineering. Springer.

  Excellent monograph on MC methods in finance.

- Hertz, D.B. (1964) Risk analysis in capital investment. *Harvard Business Review*.

  MC in finance.

- Boyle, P. (1977). Options: a Monte Carlo approach. *J. Financial Economics*

  MC pricing without early-exercise features.

- Carriere, J. (1996) Valuation of early-exercise price of options using simulation and nonparametric regression. *Insurance: Math. and Economics*.

- Grau, A. (2008) Applications of least-squares regressions to pricing and hedging of financial derivatives.

  Quadratic hedging by least squares MC.

- Andersen, Broadie (2004). A primal-dual simulation algorithm for pricing multidimensional American options. *Management Sciences*.
- Belomestny, Bender, Schoenmakers (2008) True upper bounds for Bermudan products via non-nested Monte Carlo, *Math. Finance*.
- Bender, Kolodko, Schoenmakers (2006) Iterating cancellable snowballs and related exotics. *Risk*.
- Clément, Lamberton, Protter (2002). An analysis of a least squares regression algorithm for American option pricing. *Finance Stoch.*
- Egloff (2005) Monte Carlo algorithms for optimal stopping and statistical learning. *Ann. Appl. Probab.*
- Glasserman, Yu (2004) Number of paths vs. number of basis functions in American Option pricing. *Ann. Appl. Probab.*
- Haugh, Kogan (2004). Pricing American options: a duality approach. *Operations Research*.
- Kolodko, Schoenmakers (2006). Iterative construction of the optimal Bermudan stopping time. *Finance Stoch.*
- Longstaff, Schwartz (2001). Valuing American options by simulation: a simple least-squares approach. *Rev. Financial Stud.*
- Piterbarg (2004). Pricing and hedging callable Libor exotics in forward Libor models. *Journal of Computational Finance*.
- Rogers (2001). Monte Carlo valuation of American options. *Math. Fin.*