Addendum to

An Iterative Method for Multiple Stopping: Convergence and Stability

by C. Bender and J. Schoenmakers

Some general remarks

In this paper we suggest a numerical implementation based on plain Monte Carlo simulation of the conditional expectations in the following Markovian setting: Suppose $(X(i), \mathcal{F}_i)$, $0 \le i \le k$, is a possibly high-dimensional Markov process and the cashflow is of the form $Z(i) = f(i, X(i))$. Assume a consistent stopping family $\tau(i)$ depends on ω only through the path of X and that for each i the event $\{\tau(i) > i\}$ is $X(i)$ measurable. Precisely, we suppose τ may be represented by

$$\tau(i) = \mathcal{T}(i; X(i), \ldots, X(k))$$

for explicitly known functions $\mathcal{T}(i; \cdot)$. A typical example of such stopping families is

$$\tau_g(i) = \inf\{j \ge i; f(i, X(i)) \ge g(i, X(i))\}$$

for a given function $g(i, x)$.

Then all conditional expectations to be evaluated throughout the theoretical iteration of (6)–(9) are of the form $(j \ge i)$

$$\begin{aligned} E^{\mathcal{F}_i} Z(\tau(j)) &= E^{\mathcal{F}_i} \sum_{p=j}^{k} 1_{\tau(j)=p} f(p, X(p)) \\ &= E^{X_i} \sum_{p=j}^{k} 1_{\tau(j)=p} f(p, X(p)) = E^{X_i} f(\tau(j), X(\tau(j))) = (*) \end{aligned}$$

To obtain a sample of $(*)$ simulate a trajectory X^{sample} of X up to time i. From time i to time k simulate under the conditional measure, say $P^{X_i^{\text{sample}}}$, N trajectories $(X^1(p), \ldots, X^N(p); p = i, \ldots, k)$, i.e. such that $X^n(i) = X^{\text{sample}}(i)$ for all $n = 1, \ldots, N$. Along each trajectory $n = 1, \ldots, N$ we determine $\tau^n(i) = \mathcal{T}(i; X^n(i), \ldots, X^n(k))$. Then a sample estimate of $(*)$ is defined as

$$\hat{E}^{\mathcal{F}_i, \text{sample}} Z(\tau(j)) := \frac{1}{N} \sum_{n=1}^{N} f(\tau^n(j), X^n(\tau^n(j))) \tag{1}$$

Obviously, this plain Monte Carlo estimator doesn't require any approximation of the state space and thus, in particular, does not invoke curse of dimensionality in dependence of the dimension of the underlying Markov process. Since the Monte Carlo estimator (1) requires N samples of the Markov process to obtain one sample of the conditional expectation, it becomes costly to approximate high order nestings of conditional expectations this way. (A nesting of order m requires about N^{m+1} samples of X to get N samples of the mth nested conditional expectation. Of course this is only feasible for small m, say 1 or 2.) This is why we recommend to perform only one or two improvement steps.

Algorithm for two exercise rights

We now provide a pseudo-code for the implementation in the case of two exercise rights. It is clear how to extend this code to any number of exercise rights.

Suppose we are given an input stopping family $(\sigma_1(i), \sigma_2(i))$ which gives a lower approximation ($\delta = 1$),

$$Y(i, \sigma_1, \sigma_2) = E^{\mathcal{F}_i}\left[Z\left(\sigma_2(i)\right) + Z\left(\sigma_1(\sigma_2(i)+1)\right)\right].$$

The new family

$$\begin{aligned} \widetilde{\sigma}_1(i) &= \inf\left\{ j \geq i : Z(j) \geq \max_{j+1 \leq p \leq k} E^{\mathcal{F}_j} Z(\sigma_1(p)) \right\} \\ \widetilde{\sigma}_2(i) &= \inf\left\{ j \geq i : Z(j) + E^{\mathcal{F}_j} Z(\sigma_1(j+1)) \right. \\ &\qquad \left. \geq \max_{j+1 \leq p \leq k} E^{\mathcal{F}_j}\left[Z\left(\sigma_2(p)\right) + Z\left(\sigma_1(\sigma_2(p)+1)\right)\right] \right\} \end{aligned}$$

gives improved approximation

$$Y(i, \widetilde{\sigma}_1, \widetilde{\sigma}_2) = E^{\mathcal{F}_i}\left[Z\left(\widetilde{\sigma}_2(i)\right) + Z\left(\widetilde{\sigma}_1(\widetilde{\sigma}_2(i)+1)\right)\right] \geq \max_{i \leq p \leq k} Y(p, \sigma_1, \sigma_2).$$

The following algorithm supposes the Markovian setting described in the previous section.

Algorithm for computing $\widetilde{\sigma}_1(i)$ on a trajectory $(X(j),\ j = i, \dots, k)$.

$A^{(1)}(i)$:

- Set $l := i$;

  (A) To decide whether $l \approx \widetilde{\sigma}_1(i)$ or not we do the following:

  - ∗ Simulate M trajectories $(X^{(p)}(q),\ q = l, \dots, k),\ p = 1, \dots, M$, with start value $X(l)$;
  - ∗ Along each trajectory (p) determine the family $(\sigma_1^{(p)}(q),\ q \geq l)$ (recall the stopping family is a function of the path);
  - ∗ Then, for $q = l+1, \dots, k$ compute $Dummy[q] := \frac{1}{M}\sum_{p=1}^{M} Z(\sigma_1^{(p)}(q)) \approx E^{\mathcal{F}_l} Z(\sigma_1(q))$;
    Next determine
    $Max\_Dummy := \max_{l+1 \leq q \leq k} Dummy[q] \approx \max_{l+1 \leq q \leq k} E^{\mathcal{F}_l} Z(\sigma_1(q))$;
  - ∗ Check whether $Z^{(l)} \geq Max\_Dummy$:
    - · If yes, $\widetilde{\sigma}_1(i) :\approx l$;
    - · If no, do $l := l+1$; if $l < k$ go to (A) and repeat; if $l = k$ set $\widetilde{\sigma}_1(i) = k$;

  – We so end with up an estimation of $\widetilde{\sigma}_1(i)$.

Algorithm for computing $\widetilde{\sigma}_2(i)$ on a trajectory $(X(j),\ j = i, \dots, k)$.

$A^{(2)}(i)$:

- Set $l := i$;

  (A) To decide whether $l \approx \widetilde{\sigma}_2(i)$ or not we do the following:

  - ∗ Simulate M trajectories $(X^{(p)}(q),\ q = l, \dots, k),\ p = 1, \dots, M$, with start value $X(l)$;
  - ∗ Along each trajectory (p) search $\sigma_1(l+1)$.
  - ∗ Then, compute
    $Dummy := \frac{1}{M}\sum_{p=1}^{M} Z(\sigma_1^{(p)}(l+1)) \approx E^{\mathcal{F}_l} Z(\sigma_1(l+1))$;
    Next compute for $q = l+1, ..., k$, $Dummy[q]$ via the routine $A^{(1,2)}(l, q)$ (with the same paths as used to compute $Dummy$ as input). Determine
    $Max\_Dummy := \max_{l+1 \leq q \leq k} Dummy[q]$
    $\approx \max_{l+1 \leq q \leq k} E^{\mathcal{F}_l}\left[Z\left(\sigma_2(q)\right) + Z\left(\sigma_1(\sigma_2(q)+1)\right)\right]$;
  - ∗ Check whether $Z^{(l)} + Dummy \geq Max\_Dummy$:

- · If yes, $\widetilde{\sigma}_2(i) :\approx l$;
  - · If no, do $l := l + 1$; if $l < k$ go to (A) and repeat; if $l = k$ set $\widetilde{\sigma}_2(i) = k$;
- – We so end with up an estimation of $\widetilde{\sigma}_2(i)$.

Algorithm for $E^{\mathcal{F}_i}\left[Z\left(\sigma_2(q)\right) + Z\left(\sigma_1(\sigma_2(q)+1)\right)\right]$, $i \leq q \leq k$.

$A^{(1,2)}(i,q)$:

- Input: M trajectories $(X^{(p)}(q),\ q = i, \ldots, k)$, $p = 1, \ldots, M$, with start value $X(i)$;
- Along each trajectory (p) search $\sigma_2^{(p)}(q)$ and $\sigma_1^{(p)}(\sigma_2^{(p)}(q)+1)$
- return $\frac{1}{M}\sum_{p=1}^{M}\left[Z\left(\sigma_2^{(p)}(q)\right) + Z\left(\sigma_1^{(p)}(\sigma_2^{(p)}(q)+1)\right)\right]$

Now, having a procedure for constructing $\widetilde{\sigma}_1$ and $\widetilde{\sigma}_2$, we obtain an algorithm for
$Y(0, \widetilde{\sigma}_1, \widetilde{\sigma}_2)$:

- Simulate M trajectories $(X^{(p)}(q),\ q = 0, \ldots, k)$, $p = 1, \ldots, M$, with start value $X(0)$;
- Along each trajectory (p) construct $\widetilde{\sigma}_2^{(p)}(0)$ and $\widetilde{\sigma}_1^{(p)}(\widetilde{\sigma}_2^{(p)}(0)+1)$
- return $\frac{1}{M}\sum_{p=1}^{M}\left[Z\left(\widetilde{\sigma}_2^{(p)}(0)\right) + Z\left(\widetilde{\sigma}_1^{(p)}(\widetilde{\sigma}_2^{(p)}(0)+1)\right)\right]$

Example: the (Libor) chooser cap

Consider a set of tenor dates $T_1, \ldots, T_n$ with periods $\delta_j := T_{j+1} - T_j$ (nearly) equi-distant, for example 6 months. Let L_j be the EurIBOR (or LIBOR) over the period $[T_j, T_{j+1}]$, fixed at T_j and settled at T_{j+1}, and B_* be the spot Libor rolling over account. I.e. the numeraire which starts with $B_*(0) = 1$ Euro (or \$1), and during the period $T_j < t < T_{j+1}$ is invested in the zero bond B_{j+1} which ends up with face value 1 at T_{j+1}. For proper modelling of the Libor process L we refer to the literature.

A chooser cap over a period $[T_p, T_q]$ with strike κ and pre-specified exercise number m, $m \leq q - p$, can be regarded as a standard cap with strike κ over this period, except that only m caplets, to be chosen by the option holder, are payed off. The chooser cap price at $t = 0$ can then be represented by the solution of a multiple stopping problem, i.e.

$$\text{ChooserCap}(0) := \sup_{p \leq \tau_1 < \tau_2 < \cdots < \tau_m < q} E_* \sum_{j=1}^{m} \frac{(L_{\tau_j}(T_{\tau_j}) - \kappa)^+ \delta_{\tau_j}}{B_*(T_{\tau_j+1})}.$$