Pricing Convertible Bonds with Call Protection

Stéphane Crépey[1,*], Abdallah Rahal[1,2,*]

[1] Équipe Analyse et Probabilité
Université d'Évry Val d'Essonne
91025 Évry Cedex, France [2]

[2] Université Libanaise

November 24, 2009

Contents

*The research of the authors benefited from the support of Ito33 and of the 'Chaire Risque de crédit', Fédération Bancaire Française.

1 Introduction

In this work, which is a follow-up paper to [15], we consider the issue of pricing numerically by simulation convertible bonds on an underlying stock S. A convertible bond pays coupons from time 0 onwards, until a *terminal payoff*

$$\mathbf{1}_{\zeta=\tau<T}\ell(\tau,S_\tau)+\mathbf{1}_{\vartheta<\tau}h(\vartheta,S_\vartheta)+\mathbf{1}_{\zeta=T}g(S_T) \tag{1}$$

occurs at the minimum $\zeta=\tau\wedge\vartheta$ of two $[0,T]$-valued stopping times ϑ and τ. Here the *put time* τ and the *call time* ϑ are $[0,T]$-valued stopping times under the control of the holder and the issuer of the bond, respectively. Convertible bonds are thus products with early exercise clauses both by the holder (put clauses, like with American options) and the issuer (call clauses) of the claim, and represent as such the main practical example of a *game option*. Moreover, convertible bond call times ϑ are typically subject to constraints, called *call protections*, preventing the issuer from calling the bond on certain random time intervals.

From the mathematical point of view, the study of game options with call protection leads to doubly reflected backward stochastic differential equations with an upper barrier which is only active on random time intervals (doubly reflected BSDE with an intermittent upper barrier, or RIBSDE for short henceforth, where the 'I' in RIBSDE stands for 'intermittent'). Such RIBSDEs and, in the Markovian case, the related variational inequality (VI for short henceforth) approach, were first introduced in Crépey [16].

Now, regarding the numerical solution of the pricing equations (stochastic RIBSDEs or analytic VIs), a major concern is that in practice, call protection is typically monitored at discrete times in a possibly very path-dependent way. Issuer calls are thus allowed or not on the following time period depending on the past values of the underlying stock S, which leads, after extension of the state space to markovianize the problem, to highly-dimensional pricing problems. Deterministic pricing schemes are then ruled out by the curse of dimensionality, and simulation methods appear to be the only viable alternative.

In [14], we established a convergence rate for a discrete time approximation scheme by simulation to an RIBSDE. The purpose of the present paper is to assess the practical value of this approach, on the benchmark problem of pricing by simulation convertible bonds with highly path-dependent call protection. The results are quite convincing, so that one ends up with a both practical and mathematically justified approach to such problems. More generally, this paper is a contribution to demonstration of the real abilities of simulation/regression numerical schemes for high to very high-dimensional pricing problems (up to $d=30$ in the context of the application at hand of this paper).

1.1 Outline of the paper

We thus consider in this paper problems corresponding to more and more complex, yet commonly encountered in practice, clauses of call protection. We propose in each case a reference, but heavy, if practical, deterministic pricing scheme, as well as a more efficient (as soon as the problem dimension exceeds a few units) and practical Monte Carlo simulation/regression pricing scheme.

In each case we derive the pricing equation, study (mainly by application of [14, 16]) the convergence of the pricing schemes, and provide reports on numerical experiments.

1.2 Standing Notation

Given a finite horizon $T > 0$, the evolution of a financial market model is modeled throughout in terms of stochastic processes defined on a continuous time stochastic basis $(\Omega, (\mathcal{F}_t)_{t\in[0,T]}, \mathbb{Q})$, where $\mathbb{Q}$ denotes a risk-neutral pricing measure. We denote by $\mathcal{T}_t$ (or simply $\mathcal{T}$, in case $t = 0$) the set of $[t, T]$-valued stopping times, and by $\mathbb{E}$ (resp. $\mathbb{E}_t$) the $\mathbb{Q}$-expectation (resp. $\mathbb{Q}$ – conditional expectation given $\mathcal{F}_t$) operator.

Note that pricing theory can also be developed in discrete time, on a time grid $0 = t_0 < t_1 < \ldots < t_n = T$. In the case of a discrete time set-up, that in the context of this paper will arise by time discretisation of a continuous-time model, we will often find convenient to denote the time by i rather than t_i. So, in this case:
• X_i and $u_i(X_i)$ will be used as short-hands for X_{t_i} and $u(t_i, X_{t_i})$, given a continuous-time process X and a function $u = u(t, x)$,
• $\mathbb{E}_i$ refers to the conditional expectation with respect to the discrete information flow $\mathcal{F}_i$ until time i, and
• $\mathcal{T}_i$ (with $\mathcal{T} = \mathcal{T}_0$) stands for the set of stopping times ν taking their values in $\{i, \ldots, n\}$.

For functions u of three arguments t, x, k where the third argument k takes its values in a discrete set, so that k can be thought of as referring to the index of a vector or system of functions of time t and the spatial variable x, we shall denote either $u(t, x, k)$, or $u^k(t, x)$, depending on what is more convenient in the context at hand.

We denote finally:
• $\mathbb{N}_n = \{0, 1, \ldots, n\}$, for every non-negative integer n,
• R^q and $R^{1\otimes q}$, the set of q-dimensional vectors and row-vectors with real components,
• $^\mathsf{T}$, the transposition operator.

2 Benchmark Model

2.1 Primary Market Model

For all the numerical experiments, we shall consider the following *local drift and volatility model* for a non-negative underlying process S, implicitly parameterized by the initial condition x of S :

$$dS_t = S_t(b(t, S_t)dt + \sigma(t, S_t)dW_t)\,,\ S_0 = x \tag{2}$$

where:
• W is a standard univariate $\mathbb{Q}$ – Brownian motion,
• $b(t, S)$ is a local drift coefficient, to be interpreted as a *risk-neutral drift*, possibly accounting for riskless interest-rate, dividend yields on S, and/or credit-risk adjustment on S (see, e.g., [16, 4]), and
• $\sigma(t, S)$ is a local volatility function.
Equivalently, S follows a one-dimensional diffusion with generator given by, denoting $\partial_t u = \frac{\partial u}{\partial t}\,, \partial u = \frac{\partial u}{\partial S}\,, \partial^2 u = \frac{\partial^2 u}{\partial S^2}$ for every function $u = u(t, S)$:

$$\mathcal{G}u \equiv \partial_t u + bS\partial u + \frac{1}{2}\sigma^2 S^2 \partial^2 u\ . \tag{3}$$

We postulate the standard Lipschitz and growth assumptions on the model coefficients ensuring that the SDE (2) admits a unique (strictly, say for simplicity) positive solution $(S_t)_{t\in[0,T]}$, where T stands for the maturity of a generic contingent claim on S.

2.1.1 First-Variation Process

Let $B(t,S) = b(t,S)S$, $\Sigma(t,S) = \sigma(t,S)S$. We assume that B and Σ are of class $\mathcal{C}^1_b$ with Lipschitz first derivatives. It is then well-known that the so-called *first-variation* or *flow* process ∇ of S, formally given by, writing explicitly the initial condition x of S as a superscript in this equation (cf. (2)),

$$\nabla_t = \lim_{\varepsilon\to 0+} (2\varepsilon)^{-1}(S_t^{x+\varepsilon} - S_t^{x-\varepsilon}) \;, \tag{4}$$

is a well-defined process, which can be characterized as the solution to the following SDE:

$$d\nabla_t = \nabla_t\Big(\partial B(t,S_t)dt + \partial\Sigma(t,S_t)dW_t\Big) \;,\;\; \nabla_0 = 1. \tag{5}$$

2.1.2 Time-Discretization

The previous processes are involved in the computation of financial derivatives prices and deltas. For numerical purposes, one needs to approximate both (2) and (5). Given a discrete time-grid $0 = t_0 < t_1 < \ldots < t_n = T$, we shall consider the Euler schemes defined by (cf. end of section 1.2 regarding our notational convention for time-discrete processes),

$$\begin{aligned} S_{i+1} &= S_i + B_i(S_i)(t_{i+1} - t_i) + \Sigma_i(S_i)(W_{t_{i+1}} - W_{t_i}) \;,\; S_0 = x \\ \nabla_{i+1} &= \nabla_i\Big(1 + \partial B_i(S_i)(t_{i+1} - t_i) + \partial\Sigma_i(S_i)(W_{t_{i+1}} - W_{t_i})\Big) \;,\; \nabla_0 = 1 \;. \end{aligned} \tag{6}$$

2.1.3 Model Parameterization

More precisely, we shall take,

$$b(t,S) = r(t) - q(t) + \eta\gamma(t,S) \;,\; \mu(t,S) = r(t) + \gamma(t,S) \;, \tag{7}$$

where:
• the *riskless short interest rate* $r(t)$, the *equity dividend yield* $q(t)$ on S, and the *local default intensity* $\gamma(t,S) \geq 0$ of the firm issuing the bond are bounded, Borel-measurable functions, and
• $\eta \leq 1$ is a real constant, to be interpreted as the *fractional loss* on S in case of a default of the firm issuing the bond.
We refer the reader to [4] for details and background, and for the credit-risk interpretation of this specification. To be even more specific, we shall let

$$\gamma(t,S) = \gamma_0(S_0/S)^\alpha \;,\; \sigma(t,S) = \sigma \;, \tag{8}$$

for non-negative default intensity parameters γ_0 and α, and for a constant volatility parameter σ. Therefore

$$\begin{aligned} &\partial\mu(t,S) = \partial\gamma(t,S) = -\alpha\gamma_0 S_0^\alpha/S^{1+\alpha} \\ &B(t,S) = (r(t) - q(t))S + \eta\gamma_0 S_0^\alpha S^{1-\alpha} \;,\; \Sigma(t,S) = \sigma S \\ &\partial B(t,S) = r(t) - q(t) + (1-\alpha)\eta\gamma_0(S_0/S)^\alpha \;,\; \partial\Sigma(t,S) = \sigma \;. \end{aligned} \tag{9}$$

By a suitable choice of the model parameters, this simple equity-to-credit framework allows one to account for rather different situations. For instance (see [4]):
• The 'total default' case with $\alpha > 0$ and $\eta = 1$ can be used in the situation where S represents the value of the equity of the reference entity (firm issuing the convertible bond), so S jumps to zero in case of default of the reference entity, assuming no recovery upon the stock of the reference entity upon default;
• The 'partial default' case with $\alpha = 0$ and $\eta = 0$ can be used in the situation where S represents the value of the equity of a firm different from that issuing the bond (case of the so-called *exchangeable bonds*).

2.2 Convertible Bond

Given a non-decreasing sequence of stopping times ϑ, which will represent the times of switching of call protection in the financial interpretation (see [16, 15]), let $\mathcal{T}^\vartheta$ denote the set of all the $\cup_{l>0}[\vartheta_{2l-1}, \vartheta_{2l}) \cup \{T\}$ – valued stopping times.

Using the above model of the underlying asset S, we shall consider a convertible bond continuously paying coupons $c(t, S_t)dt$ from time 0 onwards, until a terminal payoff (1) is paid at time $\tau \wedge \vartheta$, where $(\tau, \vartheta) \in \mathcal{T} \times \mathcal{T}^\vartheta$, with, specifically,

$$\ell(t, S_t) = \ell(S_t) = L_t = \bar{P} \vee S_t\,,\ h(t, S_t) = h(S_t) = \bar{C} \vee\ S_t\,,\ g(S_T) = \xi = \bar{N} \vee S_T \quad (10)$$

$$\partial\ell(S) = \mathbf{1}_{S \geq \bar{P}}\,,\ \partial h(S) = \mathbf{1}_{S \geq \bar{C}}\,,\ \partial g(S) = \mathbf{1}_{S \geq \bar{N}}\,, \quad (11)$$

for non-negative constants $\bar{P} \leq \bar{N} \leq \bar{C}$. One also sets

$$U_t = \mathbf{1}_{l_t \text{even}} \infty + \mathbf{1}_{l_t \text{odd}} h(S_t) \quad (12)$$

with l_t defined by $\vartheta_{l_t} \leq t < \vartheta_{l_t+1}$.

Accounting for credit risk and recovery on the bond upon default (see [3, 4, 16]), one assumes the following form for the coupon rate function c :

$$c(t, S) = \bar{c}(t) + \gamma(t, S)\left((1 - \eta)S \vee \bar{R}\right)\,, \quad (13)$$

where $\bar{c}$ denotes a *nominal coupon rate* function, and $\bar{R}$ stands for a *nominal recovery on the bond upon default*. Coupons $c(t, S_t)dt$ are thus assumed to be continuously paid.

Remark 2.1 In practice coupons are of course discrete rather then continuously paid, which should result in a discrete stream of nominal coupons instead of the continuously paid nominal coupon rate $\bar{c}$ in (13). In the theoretical description of the model and algorithms in this paper we consider continuously paid coupons for simplicity of presentation. However, following the guidelines of [3, 4, 16], the results and methods of the present paper can be extended in a straightforward way to the case of discrete coupons. In the numerical experiments we thus work with discrete coupons, using the methodology of [3, 4, 16] in this regard.

We finally denote by $\beta_t = e^{-\int_0^t \mu(s, S_s)ds}$ a risk-neutral *credit-risk adjusted discount factor*, where $\mu(t, S) = r(t) + \gamma(t, S)$ is the *credit-risk adjusted interest rate* (see [4, 16, 3]), and we set $f(t, S, y) = c(t, S) - \mu(t, S)y$.

2.3 General Conditions for the Numerical Experiments

The numerical data of Table 1 will be used by default throughout, m standing for the number of Monte Carlo simulations (trajectories) which are used in the stochastic pricing schemes.

$\overline{P}$	$\overline{N}$	$\overline{C}$	μ	σ	r	q	γ_0	α	m
0	100	103	1	0.2	0.05	0	0.02	1.2	10^4

Table 1: *General Data.*

In all the numerical experiments we use a constant time-step $t_{i+1} - t_i = \mathrm{h}$, where the time step index varies from 0 to n, and with:
• h = six hours (four time steps per day) in the case of simulation methods, and
• h = one day in the case of deterministic schemes.

The space-steps in the S variable are, the superscript ^{j} referring to a generic space step index (space step in the sense of a trajectory's index varying between 1 and m , in the case of simulation pricing schemes):
• $S^{j+1} - S^j = 0.5$ in the case of deterministic schemes, and
• Cells of diameter one in the case of simulation/regression methods involving a method of cells in the direction of the S variable (see the Appendix); so, at every time step i, the S_i^js are partitioned into segments of $\mathbb{R}_+$ of length one.

Regarding the deterministic numerical schemes, fully implicit finite differences schemes are used throughout.

3 No Call

Sections 3 and 4 respectively deal with the basic cases of no call and no call protection. Sections 5 to 7 correspond to the special case of a call protection before a stopping time ϑ_1 (call possible on $[\vartheta_1, T]$), whereas section 8 corresponds to the case of a more general 'truly intermittent' call protection.

Let us now start by the easiest case in which $\vartheta_1 = T$, so $\cup_{l>0}[\vartheta_{2l-1}, \vartheta_{2l}) \cup \{T\} = \{T\}$. This corresponds to the case of dividend-paying American options. If not for dividends, this case has been studied in depth in the literature. See, among others, Longstaff and Schwartz [25], Tsitsiklis and VanRoy [31, 32], Broadie and Glasserman [10, 11], Lions and Régnier [24], Bouchard et al. [6], Pages and Bally [27], or Chapter 6 of Glasserman [19] for a survey.

3.1 Pricing Equation

Let us first present the pricing equations. As is well known, in the risk-neutral model (2), the discounted $\mathbb{Q}$ – price process $\beta\Pi$ of an American claim with coupon rate c, early payoff process L_t and payoff at maturity ξ, is given by the *Snell envelope* of the cumulative discounted payoff process defined by, for $t \in [0, T]$,

$$\int_0^t \beta_s c_s ds + \beta_t \big(\mathbf{1}_{t<T} L_t + \mathbf{1}_{t=T} \xi \big) \ .$$

So, for $t \in [0,T]$,

$$\beta_t \Pi_t = \text{esssup}_{\tau \in \mathcal{T}_t} \mathbb{E}_t \left\{ \int_t^\tau \beta_s c_s ds + \beta_\tau \big(\mathbf{1}_{\{\tau<T\}} L_\tau + \mathbf{1}_{\{\tau=T\}} \xi \big) \right\} .$$

The connection with hedging can be established (see [16]) in terms of the related BSDE $(\mathcal{E})$, which in this simple case assumes the following form (see, e.g., [4]),

$$\begin{array}{l} \Pi_T = \xi, \text{ and for } t \in [0,T], \\ \left\{ \begin{array}{l} -d\Pi_t = (c_t - \mu_t \Pi_t)dt + dA_t - \Delta_t \sigma(t,S_t) S_t dW_t \\ L_t \leq \Pi_t \,, \; (\Pi_t - L_t) dA_t = 0 \end{array} \right. \end{array} \tag{14}$$

to be solved in (Π, Δ, A) in the usual spaces of square integrable processes (see, for instance, [15, 16]). In particular, A is sought for as a continuous and non-decreasing process, and Π is thus a continuous process as well.

One then has by application of the results of [18] that the reflected BSDE (14) is well-posed, and that

$$\Pi_t = u(t,S_t) \,, \; \Delta_t = \partial u(t,S_t) \tag{15}$$

(provided u is sufficiently regular, regarding the delta), where $u = u(t,S)$ is the *pricing function*, unique solution[1] to the following *pricing VI*:

$$\left\{ \begin{array}{l} \max \Big(\mathcal{G} u + c - \mu u, \ell - u \Big) = 0 \text{ on } [0,T) \times (0,+\infty) \\ u = g \text{ at } T \,. \end{array} \right. \tag{16}$$

Let us briefly recall the generic multinomial recombining tree algorithm for solving (16), on a rectangular time-space grid (t_i, S^j) discretizing the time-state space, where i and j index the time and space step in the algorithm: $u_n(j) = g(S^j)$ for $j = 1, \ldots, m$, and then for $i = n-1 \ldots 0$, for $j = 1 \ldots m$,

$$u_i^j = \max \left(\ell_i \left(S^j \right) , \, e^{-\mu_i^j \mathrm{h}} \sum_k p_i^{j,k} (u_{i+1}^{j+k} + \mathrm{h} c_{i+1}^{j+k}) \right) \,, \tag{17}$$

where ℓ_i is the call payoff function at time step i (time ih) in the algorithm, and where the $p_i^{j,k}$s are suitable weights. These weights are typically obtained by substitution of Taylor expansions for u and its derivatives in the pricing VI (16), possibly followed by resolution of a linear system in the case of implicit schemes (see, for instance, Morton and Mayers [26], or Duffy [17]).
Such deterministic approximation schemes are stable if $p \geq 0$ and convergent if, additionally, consistent, where the latter condition is satisfied for a vast class of discretisation schemes. An approximation $(\delta_i^j)_{1 \leq i \leq n}^{2 \leq j \leq m}$ for the *delta function* $\Delta(t,S) = \partial u(t,S)$ at the interior points of the time-space grid can then be obtained by the formula:

$$\delta_i^j = \frac{u_i^{j+1} - u_i^{j-1}}{S_i^{j+1} - S_i^{j-1}} \,. \tag{18}$$

[1] Viscosity solution with polynomial growth in S, see [16].

3.2 Simulation Schemes

For solving (16) by simulation, a possible procedure consists in writing a dynamic programming equation as of (17), but on a *stochastically generated (hence, non recombining) mesh* $(S_i^j)_{0\le i\le n}^{1\le j\le m}$, using an appropriate discretization scheme S_i for the underlying diffusion S_t (like for instance the Euler scheme of section 2.1.2). We thus get the following amendment to (17): $u_n^j = g(S_n^j)$ for $j = 1 \dots m$, and then for $i = n-1 \dots 0$, for $j = 1 \dots m$,

$$u_i^j = \max\left(\ell_i(S_i^j),\, e^{-\mu_i^j \mathrm{h}} \mathbb{E}_i^j\big(u_{i+1} + \mathrm{h}c_{i+1}\big)\right) \tag{19}$$

where $\mathbb{E}_i^j$ stands for the conditional expectation given $S_i = S_i^j$. A numerical approximation $\delta = (\delta_i^j)$ of the delta function on the grid could be deduced from (u_i^j) by the formula (18), however convergence of the related estimate is not established, and this estimate typically exhibits a very high variance in practice. A better estimate is the following *regression estimate*,

$$\delta_i^j = \frac{\mathbb{E}_i^j\left\{u_{i+1}(W_{i+1} - W_i)\right\}}{\sigma_i(S_i^j) S_i^j \mathrm{h}}\ . \tag{20}$$

The simulation pricing scheme thus ultimately hinges on the computation of conditional expectations sitting in (19) or (20) (for $i \ge 1$, since for $i = 0$ the conditional expectation reduces to a simple expectation). At step $i \ge 1$, these conditional expectations can be computed by *non-linear regression* of $(u_{i+1}^j + \mathrm{h}c_{i+1}^j)^{1\le j\le m}$ or $(u_{i+1}^j(W_{i+1}^j - W_i^j))^{1\le j\le m}$, which are already known at the time step i of the algorithm, against $(S_i^j)^{1\le j\le m}$. We refer the reader to the Appendix regarding non-linear regressions.

Rates of convergence regarding the discretization in time which is implicit in (19) are derived in [5]. A complete time-space convergence analysis could then be conducted by proceeding along the lines of Lemor, Gobet and Warin [23] (see also Lemor [22, section III.2]).

Remark 3.1 **(i)** As opposed to the case of standard Monte Carlo pricing schemes, *confidence intervals* are not available in this case. It is possible however to derive an upper bound on the price by resorting to a suitable dual Monte Carlo approach (see Rogers [30]). Since most pricing methods provide lower bounds, one thus end up with an interval. Moreover it is of course still possible to derive a *confidence interval of the method* by running the simulation for various (say, 50) seeds of the generator and computing a standard deviation of the estimated prices.
(ii) One can recover from the pricing function estimated by (17) or (19) the following estimators of the put region and of the optimal put policy (starting from time 0, for the latter; cf. Crépey[16]:

$$\mathcal{E}_p = \left\{(i, S_i^j)\,;\, u_i^j = \ell_i(S_i^j)\right\},\ \tau^j = \inf\{i \in \mathbb{N}_n\,;\, S^j \in \mathcal{E}_p\} \wedge n\ . \tag{21}$$

3.2.1 Price and Delta at Time 0

The previous scheme directly suffers from the accumulation of errors that occur through the iterated computation of the conditional expectations that are involved in the dynamic programming equations (19). To limit the impact of these errors, an preferred alternative,

as far as the price *at time 0* is concerned, consists in only retaining from (19) the estimated optimal stopping policy ν as of (21). One then uses ν for computing an estimator $\widetilde{u}_0$ of the option price at time 0 alternative to u_0 in (19), defined by the empirical average

$$\widetilde{u}_0 = \mathbb{E}^m\Big\{\mathrm{h}\sum_{i=1}^{\tau^j}\beta_i^j c_i(S_i^j) + \beta_{\tau^j}^j\Big(\mathbf{1}_{\{\tau^j<n\}}\ell_{\tau^j}(S_{\tau^j}^j) + \mathbf{1}_{\{\tau^j=n\}}g(S_{\tau^j}^j)\Big)\Big\} \tag{22}$$

where we set $\beta_l^j = e^{-\mathrm{h}\sum_{k=0}^{l-1}\mu_k^j}$.

Whereas (19) typically overestimates the exact price, on the opposite (22) typically underestimates it (and is also typically more accurate than (19), see Remark 3.2(ii)). Computing u_0 and $\widetilde{u}_0$ thus gives another way to end up with an interval (cf. Remark 3.1(i)). If this interval is too large, it typically means that the functional basis used for computing the conditional expectations is not well chosen (see the Appendix and p. 479 of Glasserman [19]).

Likewise, a simulation estimate for the option's delta at time 0 (Δ_0 in (14)) alternative to $\delta_0^{j_0}$ in (20), where the superscript j_0 refers to the index of the space-grid point S^j closest to S_0, is given by,

$$\begin{aligned}\widetilde{\delta}_0 = \mathbb{E}^m\Big\{\mathrm{h}\sum_{i=1}^{\tau^j}\Big(\beta_i^j\partial c_i(S_i^j)\nabla_i^j + \varepsilon_i^j c_i(S_i^j)\Big) + \beta_{\tau^j}^j\Big(\mathbf{1}_{\{\tau^j<n\}}\partial\ell_{\tau^j}(S_{\tau^j}^j) + \mathbf{1}_{\{\tau^j=n\}}\partial g(S_n^j)\Big)\nabla_{\tau^j}^j \\ + \varepsilon_{\tau^j}^j\Big(\mathbf{1}_{\{\tau^j<n\}}\ell_{\tau^j}(S_{\tau^j}^j) + \mathbf{1}_{\{\tau^j=n\}}g(S_n^j)\Big)\Big\}\,,\end{aligned} \tag{23}$$

where (cf. (9), (10), (13))

$$\begin{aligned}&\partial\mu(t,S) = \partial\gamma(t,S) = -\alpha\gamma_0 S_0^\alpha/S^{1+\alpha}\,,\ \partial\ell(t,x) = \mathbf{1}_{S\geq\bar{P}}\,,\ \partial g(x) = \mathbf{1}_{S\geq\bar{N}}\\ &\partial c(t,S) = \partial\gamma(t,S)\big((1-\eta)S\vee\bar{R}\big) + (1-\eta)\gamma(t,S)\mathbf{1}_{(1-\eta)S\geq\bar{R}}\end{aligned} \tag{24}$$

and where $\varepsilon_i^j = -\mathrm{h}\beta_i^j\sum_{l=1}^i\partial\mu_l(S_l^j)\nabla_l^j$ is a discretization of the first-variation process

$$-\beta_t\int_0^t\partial\mu(s,S_s)\nabla_s ds$$

of $\beta_t = e^{-\int_0^t\mu(s,S_s)ds}$.

Remark 3.2 (i) We refer the reader to Remark 5.5 of Bouchard and Chassagneux [5] for the derivation of representations similar to (22)-(23) in the context of the discretely reflected BSDE associated to the Euler scheme of an underlying diffusion. Actually, working as they do but using a discrete time Euler scheme of the underlying diffusion instead of the continuous Euler scheme in their case, would give rise to space-continuous analogs of the representations (22) and (23) for the quantities denoted in [15] by $\widetilde{Y}_0$ and $\widehat{Z}_0$. These quantities represent the values of discrete time approximations, convergent with some established rates, to the option's price and delta at time 0 (to be precise, $\widehat{Z}$ is the integrand of the Brownian motion in the integral represention of the discrete time approximation of Z, itself denoted by $\bar{Z}$ in [15]). Also note that all these representations admit straightforward extensions to game options. This will be used without further justification in section 4.2 below.

(ii) An interpretation of the better practical behavior of (22) and (23) as compared to (19) and (20) (see section 4.3) is that, as mentioned above, (22)-(23) do not directly suffer from the accumulation of errors that is present in (19)-(20). Note however that (22)-(23) ultimately rely on the dynamic programming equation (19), which, in their case, is used in a first stage for deriving τ^j. Some people argue that the pricing function is typically not very sensitive to the optimal stopping policy, which in their view would explain why, altogether, (22)-(23) is more accurate than (19)-(20). It is our opinion however that a serious theoretical study of this phenomenon still remains to be done.
(iii) The previous approaches by simulation/regression are essentially the ones that were developed in Tsitsiklis and VanRoy [31, 32] using iteration on the values as of (19), Longstaff and Schwartz [25] using iteration on the policies as of (22), or Broadie and Glasserman [10, 11]. See also Chapter 6 of Glasserman [16] for a survey. Since all this is very well documented in the literature, we do not provide any numerical results.

4 No Call Protection

We now assume no call protection, so $\vartheta_1 = 0, \vartheta_2 = T$, and thus $\mathcal{T}_t^{\vartheta} = \mathcal{T}_t$.

4.1 Pricing Equation

By standard results on game options (see, e.g., [2, 4]), the discounted $\mathbb{Q}$ – price process $\beta\Pi$ of the bond is then given by the *value process of the related Dynkin game*, so for $t \in [0, T]$, with $\zeta = \tau \wedge \vartheta$,

$$\beta_t \Pi_t = \mathrm{essinf}_{\vartheta \in \mathcal{T}_t} \mathrm{esssup}_{\tau \in \mathcal{T}_t} \mathbb{E}_t \left\{ \int_t^{\zeta} \beta_s c_s ds + \beta_\zeta \left(\mathbf{1}_{\{\tau \leq \vartheta, \tau < T\}} L_\tau + \mathbf{1}_{\{\vartheta < \tau\}} U_\vartheta + \mathbf{1}_{\{\tau = \vartheta = T\}} \xi \right) \right\}$$

which, under mild assumptions, coincides with the analog expression with $\mathrm{esssup}_{\tau \in \mathcal{T}_t} \mathrm{essinf}_{\tau \in \mathcal{T}_t}$ instead of $\mathrm{essinf}_{\tau \in \mathcal{T}_t} \mathrm{esssup}_{\tau \in \mathcal{T}_t}$ therein, and where in the present case the general expression for U_t in (12) reduces to $U_t = h(S_t)$.

The related BSDE $(\mathcal{E})$ writes (see [4, 16]),

$$\begin{array}{l} \Pi_T = \xi, \text{ and for } t \in [0, T], \\ \left\{ \begin{array}{l} -d\Pi_t = (c_t - \mu_t \Pi_t) dt + dA_t - \Delta_t \sigma(t, S_t) S_t dW_t \\ L_t \leq \Pi_t \leq U_t \,, \; (\Pi_t - L_t) dA_t^+ = (U_t - \Pi_t) dA_t^- = 0 \end{array} \right. \end{array} \tag{25}$$

to be solved in (Π, Δ, A), where in particular A stands for a continuous finite variation process with square integrable *Jordan components*[2] $A^{\pm}$.

Moreover, we have as in section 3.1 that $\Pi_t = v(t, S_t)$, where the pricing function v solves the following pricing equation:

$$\left\{ \begin{array}{l} \min\left(\max\left(\mathcal{G}v + c - \mu v, \ell - v\right), h - v\right) = 0 \text{ on } [0, T) \times (0, +\infty) \\ v = g \text{ at } T \,. \end{array} \right. \tag{26}$$

[2]Terms $A^{\pm}$ of the unique decomposition $A = A^+ - A^-$ as the difference of two non-decreasing processes null at 0 and defining mutually singular random measures on $[0, T]$.

The 'no call' deterministic pricing algorithm (17) simply needs to be amended as: $v_n(j) = g(S^j)$ for $j = 1 \dots m$, and then for $i = n-1 \dots 0$, for $j = 1 \dots m$,

$$v_i^j = \min\left(h_i\left(S^j\right), \max\left(\ell_i\left(S^j\right), e^{-\mu_i^j \mathrm{h}} \sum_k p_i^{j,k} (v_{i+1}^{j+k} + \mathrm{h}c_{i+1}^{j+k})\right)\right) . \tag{27}$$

In (27), ℓ_i and h_i are the call and put payoff functions at time step i in the algorithm. A delta estimate (δ_i^j) is then deduced from (v_i^j) by the formula (18) with v instead of u therein.

Moreover it is possible to conduct a convergence analysis of (27) similar to that of (17) in the 'no call' case, see the comments following (17) in section 3.1.

4.2 Simulation Schemes

Likewise, (19) simply needs to be amended as: $v_n^j = g(S_n^j)$ for $j = 1 \dots m$, and then for $i = n-1 \dots 0$, for $j = 1 \dots m$,

$$v_i^j = \min\left(h_i(S_i^j), \max\left(\ell_i(S_i^j), e^{-\mu_i^j \mathrm{h}} \mathbb{E}_i^j\big(v_{i+1} + \mathrm{h}c_{i+1}\big)\right)\right) , \tag{28}$$

whereas (20) holds with v_i^j instead of u_i^j therein. For $i \geq 1$ the conditional expectations involved may be computed by non-linear regression of $(v_{i+1}^j + \mathrm{h}c_{i+1}^j)_{1 \leq j \leq m}$ against $(S_i^j)_{1 \leq j \leq m}$ as in section 3.2. The obvious analogs to Remark 3.1 may be formulated, except for the fact that it is not possible to surround the price by direct and dual approaches anymore. It is of course still possible to derive a confidence interval of the method by resimulation, see Remark 3.1(i).

This time convergence and convergence rates are covered by the results of Chassagneux [12].

We can recover from the pricing function estimated by (28) the following estimators of the put and call region and of optimal put and call policies (starting from time 0, for the latter; cf. [16]):

$$\begin{aligned} \mathcal{E}^p &= \left\{(i, S_i^j)\,;\, v_i^j = \ell_i(S_i^j)\right\},\ \tau^j = \inf\{i \in \mathbb{N}_n\,;\, S^j \in \mathcal{E}^p\} \wedge n \\ \mathcal{E}^c &= \left\{(i, S_i^j)\,;\, v_i^j = h_i(S_i^j)\right\},\ \theta^j = \inf\{i \in \mathbb{N}_n\,;\, S^j \in \mathcal{E}^c\} \wedge n\ . \end{aligned} \tag{29}$$

Moreover, alternatively to v_0 and δ_0, we have the following policy iteration estimators for the option price and delta at time 0, with $\zeta^j = \tau^j \wedge \theta^j$ (cf. (22), (23), (24)),

$$\begin{aligned} \widetilde{v}_0 &= \mathbb{E}^m\Big\{\mathrm{h}\sum_{i=1}^{\zeta^j} \beta_i^j c_i(S_i^j) + \beta_{\zeta^j}^j \Big(\mathbf{1}_{\{\zeta^j = \tau^j < n\}} \ell_{\tau^j}(S_{\tau^j}^j) + \mathbf{1}_{\{\theta^j < \tau^j\}} h_{\theta^j}(S_{\theta^j}^j) + \mathbf{1}_{\{\zeta^j = n\}} g(S_n^j)\Big)\Big\} \\ \widetilde{\delta}_0 &= \mathbb{E}^m\Big\{\mathrm{h}\sum_{i=1}^{\zeta^j} \beta_i^j \left(\beta_i^j \partial c_i(S_i^j) \nabla_i^j + \varepsilon_i^j c_i(S_i^j)\right) \\ &\quad + \beta_{\zeta^j}^j \left(\mathbf{1}_{\{\zeta^j = \tau^j < n\}} \partial \ell_{\tau^j}(S_{\tau^j}^j) + \mathbf{1}_{\{\theta^j < \tau^j\}} \partial h_{\theta^j}(S_{\theta^j}^j) + \mathbf{1}_{\{\tau^j \wedge \theta^j = n\}} \partial g(S_n^j)\right) \nabla_{\zeta^j}^j \\ &\quad + \varepsilon_{\zeta^j}^j \left(\mathbf{1}_{\{\zeta^j = \tau^j < n\}} \ell_{\tau^j}(S_{\tau^j}^j) + \mathbf{1}_{\{\theta^j < \tau^j\}} h_{\theta^j}(S_{\theta^j}^j) + \mathbf{1}_{\{\tau^j \wedge \theta^j = n\}} g(S_n^j)\right)\Big\}\ , \end{aligned} \tag{30}$$

where the v_i^js in the second line are given by (28), and where one has, along with (24), $h(t,x) = \mathbf{1}_{S \geq \bar{C}}$.

4.3 Numerical Experiments

Using the general data of Table 1 in section 2.3 along with a maturity $T = 125$ days and a nominal coupon rate $\bar{c} = 0$ in (13), Table 2 shows the standard deviation of fifty estimates of the option price obtained by changing the seed of the generator, using:
• On one hand, the MC backward estimators (28) for the price, and (20) (with (v_i^j) instead of (u_i^j) therein) for the delta,
• On the other hand, the MC forward estimators (30) for the price and the delta.
In both cases a global parametric regression basis $1, S, S^2$ is used for estimating the conditional expectations involved (see the Appendix).

The MC forward estimator has a much lower deviation, as expected.

	Value VI	Dev MC Bd	Dev MC Fd
Price	102.049	0.821	0.010
Delta	0.416	0.071	0.019

Table 2: *VI Values and MC Backward and Forward Standard Deviations over 50 trials for the option prices and deltas ($S_0 = 100.55$).*

Table 3 shows the option prices and deltas computed for various S_0s by the MC forward estimators (30), using as above a global parametric regression basis $1, S, S^2$ in (28), or by the deterministic scheme (27) for the VI (26). The errors (%Err) in this and the following Tables are the (unsigned) *percentage relative errors* of the MC price with respect to the VI price. So an error of 1 in the table means a relative error of 1% of the MC price with respect to the VI price.

S_0	VI Price	%Err MC Bd	%Err MC Fd	VI delta	%Err MC Bd	%Err MC Fd
98.55	101.246	1.90	0.04	0.376	1.07	0.07
99.55	101.637	1.92	0.01	0.396	0.95	0.50
100.55	102.049	1.99	0.01	0.416	2.77	0.67
101.55	102.479	1.65	0.07	0.435	3.97	3.47

Table 3: *MC versus VI prices and deltas.*

The MC forward estimates are more accurate than the MC backward estimates.

MC forward estimates are used by default henceforth. *MC estimate* is thus to be understood in the sequel as MC *forward* estimate.

Call protections until a stopping time In the following sections we shall consider various forms of call protection *until a stopping time* ϑ_1. Given a nominal call payoff process $h(t, X_t)$, the effective call payoff process U_t (12) accounting for the call protection is thus given by

$$U_t = \mathbf{1}_{\{t<\vartheta_1\}}\infty + \mathbf{1}_{\{t\geq\vartheta_1\}}h(t, X_t)\ . \tag{31}$$

The related BSDE $(\mathcal{E})$ is still given by (25), but with U therein given by (31) (using then

the convention that $0 \times \infty = 0$ in the last line of (25)). The process A, and thus Π, are still sought for as continuous processes (see [16, 15]).

It is intuitively clear and established in [16, 4] that the $\mathbb{Q}$-price process Π of a convertible bond with call protection before ϑ_1 coincides on $[\vartheta_1, T]$ with the no protection convertible bond price process of section 4. The only remaining issue is thus the (pre-lifting time of a call) protection pricing problem.

The most basic kind of lifting time of a call protection of course consists of a constant $\vartheta_1 = T_1 \in (0, T)$. In this case the protection pricing problem reduces to a no call pricing problem as of section 3.

5 Standard Call Protection

The next case we consider is the case of a continuously-monitored call protection (we call it 'standard call protection') corresponding to

$$\vartheta_1 = \inf\{t \in \mathbb{R}_+ \,;\, S_t \geq \bar{S}\} \wedge T \tag{32}$$

for some trigger level $\bar{S} > S_0$.

Note that this case falls outside the scope of [14], since ϑ_1 in (32) does not take discrete values. But this case was studied in details in [16, 4].

5.1 Pricing Equation

The following equation for the related protection pricing function $u = u(t, S)$ on the domain $[0, T] \times (0, \bar{S}]$ was established in [4]:

$$\begin{cases} \max\left(\mathcal{G}u + c - \mu u, \ell - u\right) = 0 \text{ on } [0, T) \times (0, \bar{S}) \\ u = v \text{ at } T \cup \bar{S} \end{cases} \tag{33}$$

where v is the no call protection pricing function of section 4.1, solution to (26). Moreover, provided that the functions u and v are sufficiently regular for Itô formulas to be applicable, one has, for $t \in [0, T]$ (see [16, 4]):

$$\Delta_t = \mathbf{1}_{\{t \leq \vartheta_1\}} \partial_S u(t, S_t) + \mathbf{1}_{\{t > \vartheta_1\}} \partial_S v(t, S_t) \,. \tag{34}$$

Knowing an approximation (v_i^j) of v, computed for instance by the deterministic scheme (27) of section 4.1, the Cauchy-Dirichlet problem (33) can be solved by standard finite differences deterministic numerical schemes, like

$$u_i^j = \max\left(\ell_i\left(S^j\right), e^{-\mu_i^j \mathrm{h}} \sum_k p_i^{j,k} (u_{i+1}^{j+k} + \mathrm{h} c_{i+1}^{j+k})\right) \tag{35}$$

at the grid points interior to the domain $[0, T] \times [0, \bar{S}]$, with a Dirichlet boundary condition $u = v$ at the grid points on the boundary $(\{T\} \times [0, \bar{S}]) \cup ([0, T] \times \{\bar{S}\})$. Convergence results for this scheme can be found in Crépey [16].

Remark 5.1 In fact the convergence results of [16] can only be considered as partial results, since one only gets the convergence of the scheme for u on $[0,T]\times[0,\bar{S}]$ conditionally on its convergence on $[0,T)\times\{\bar{S}\}$, for which no explicit criterion is given. Moreover the convergence analysis of [16] is conducted under the working assumption that the true value for v is plugged on $[0,T)\times\{\bar{S}\}$ in the approximation scheme for u, whereas in practice one has to use an approximation (v_i^j) of v. Finally this analysis only yields convergence, not convergence rates.

5.2 Simulation Scheme

Given a stochastically generated mesh $(S_i^j)_{0\le i\le n}^{1\le j\le m}$, and setting (cf. (32))

$$\vartheta_1^j = \inf\{i\in\mathbb{N}_n\,;\, S_i^j\ge\bar{S}\}\wedge n\ ,$$

a simulation algorithm for estimating the $u(t_i,S_i^j)$'s writes as follows: Set $u_n=v_n=g$, then do for $i=n-1\dots0$, for $j=1\dots m$:

- $v_i^j=\min\Big(h_i(S_i^j)\,,\ \max\Big(\ell_i(S_i^j),\,e^{-\mu_i^j\mathrm{h}}\mathbb{E}_i^j\big(v_{i+1}+\mathrm{h}c_{i+1}\big)\Big)\Big)$,
- If $i\ge\vartheta_1^j$, $u_i^j=v_i^j$, else

$$u_i^j=\max\Big(\ell_i\left(S_i^j\right),\,e^{-\mu_i^j\mathrm{h}}\mathbb{E}_i^j\big(u_{i+1}+\mathrm{h}c_{i+1}\big)\Big)\ . \tag{36}$$

So (v_i^j) here is but the no call protection price of section 4.2 that was estimated by simulation in (28).
For $i\ge1$ the conditional expectations in (36) may be computed by non-linear regression of $(u_{i+1}^j+\mathrm{h}c_{i+1}^j)_{j\in\Omega_i}$ against $(S_i^j)_{j\in\Omega_i}$, where Ω_i denotes the subset of the trajectories j such that $\{i<\vartheta_1^j\}$.

Convergence of this scheme to the solution (Π,Δ,A) of the pricing BSDE $(\mathcal{E})$, in the form here of (25) with U given by (31)-(32) therein, could be dealt with by combining arguments of Chassagneux [13] with approximation of the call time by techniques a la Bouchard–Menozzi [8]. Note that the related convergence results regarding Π would not suffer from the same limitations than those regarding the deterministic approximation of u in section 5.1 (cf. Remark 5.1). Moreover, one would also get in this way convergence and convergence rates regarding Δ, which in view of (34) essentially corresponds to the gradient of u and v.

5.2.1 Price and Delta at Time 0

We can recover from the protection pricing function u above the following estimators of the protection put region and of the optimal protection put policy (starting from time 0, for the latter; see Crépey [4, 16]):

$$\widetilde{\mathcal{E}}_p=\big\{(i,S_i^j)\,;\,u_i^j=\ell_i(S_i^j)\big\}\,,\ \widetilde{\tau}^j=\inf\{i\in\mathbb{N}_{\vartheta_1^j}\,;\,S^j\in\widetilde{\mathcal{E}}_p\}\wedge n \tag{37}$$

One then has the following policy iteration estimator for the option price at time 0, with $\widetilde{\zeta}^j=\widetilde{\tau}^j\wedge\vartheta_1^j$,

$$\widetilde{u}_0=\mathbb{E}^m\left\{\mathrm{h}\sum_{i=1}^{\widetilde{\zeta}^j}\beta_i^j c_i(S_i^j)+\beta_{\widetilde{\zeta}^j}^j\Big(\mathbf{1}_{\{\widetilde{\tau}^j<\vartheta_1^j\}}\ell_{\widetilde{\tau}^j}(S_{\widetilde{\tau}^j}^j)+\mathbf{1}_{\{\vartheta_1^j\le\widetilde{\tau}^j\}}v_{\vartheta_1^j}^j\Big)\right\}\ . \tag{38}$$

S_0	78.55	79.55	80.55	81.55
MC	103.188	103.114	103	103
VI	103.187	103.111	103	103
%Err	0.085	0.33	0	0

Table 4: *Standard Call Protection: MC price versus VI price for four values of* S_0 *(*$\bar{S} = 80$*).*

S_0	100.55	101.55	102.55	103.55
MC	103.841	103.724	103.713	103.55
VI	103.874	103.785	103.693	103.55
%Err	0.032	0.059	0.019	0

Table 5: *Same as Table 4 for* $\bar{S} = 103$.

Remark 5.2 It would be possible to write down an analogous forward estimator for the option delta at time 0 (cf. (30)), however the convergence of this estimator is not established (see Remark 3.2(i)).

5.3 Numerical Experiments

Using the general data of Table 1 in section 2.3 along with a maturity $T = 180$ days and a nominal coupon of $\bar{c} = 1.2$ per month (cf. (13) and Remark 2.1), Tables 4, 5 and 6 show the bond prices computed in two ways:
• First, by the MC forward method associated to the scheme of section 5.2, where the conditional expectations are estimated by a method of cells in S,
• Second, by the deterministic numerical scheme of section 5.1.
Again the accuracy of the simulation pricing scheme is quite satisfactory.

6 Path dependent Call Protection

Real-life protection call clauses are typically discretely monitored, e.g., reexamined at the end of each trading day, rather than continuously monitored like in section 5. Given $\bar{S} > 0$ and a fixed, increasing sequence of *monitoring times* $\mathfrak{T} = \{T_0 = 0, T_1 \ldots, T_N = T\}$, let H_t for $t \in [0, T]$ stand for the number of consecutive monitoring dates T_Is with $S_{T_I} \geq \bar{S}$ from time t backwards, capped at some fixed integer level l. So in particular at any given time t one has $H_t = 0$ if S was smaller than $\bar{S}$ at the last monitoring date before or at t. We

S_0	100.55	101.55	102.55	103.55
MC	110.082	110.819	111.351	111.809
VI	110.324	110.896	111.488	112.099
%Err	0.22	0.069	0.12	0.26

Table 6: *Same as Table 4 but for* $\bar{S} = 120$.

consider the call protection $t \geq \vartheta_1$ with

$$\vartheta_1 = \inf\{t \in \mathbb{R}_+ \, ; H_t \geq l\} \wedge T \, . \tag{39}$$

The lifting time of the call protection ϑ_1 is thus given as the first time that S_t has been $\geq \bar{S}$ at the last l monitoring dates, capped at T.

6.1 Pricing Equation

One then has by application of the results of [16] that $\Pi_t = u(t, S_t, H_t)$ on $[0, \vartheta_1)$, for a protection pricing function $u = u(t, S, k) = u_k(t, S)$, with $k \in \mathbb{N}_{l-1}$, where the restrictions of the u_ks to every set $[T_{I-1}\, T_I) \times [0, +\infty)$ are continuous, and where the limit

$$u_k(T_I-, S) = \lim_{(t,x)\to(T_I,S) \text{ with } t\leq T_I} u_k(t, x) \tag{40}$$

exists for every $k \in \mathbb{N}_{l-1}$, $I \geq 1$ and $S \neq \bar{S}$.

Moreover (see [16]), u solves the following Cauchy–Dirichlet cascade of VIs, in which v is the no-protection pricing function of section 4.1,

For I decreasing from N to 1,
• At $t = T_I$, for $k \in \mathbb{N}_{l-1}$,

$$u_k(T_I-, S) = \left\{ \begin{array}{l} u_{k+1}(T_I, S), \text{ or } v(T_I, S) \text{ if } k = l-1, \text{ on } \{S > \bar{S}\} \, , \\ u_0(T_I, S) \text{ on } \{S < \bar{S}\} \, , \end{array} \right. \tag{41}$$

or, in case $I = N$, $u_k(T-, S) = v(T, S) = g(S)$ for $S > 0$,
• On the time interval $[T_{I-1}, T_I)$,

$$\max\left(\mathcal{G}u + c - \mu u \, , \; \ell - u\right) = 0 \, . \tag{42}$$

Knowing an approximation (v_i^j) of v, this cascade of VIs can be solved by standard deterministic numerical schemes as of section 5.1. Partial convergence results for this deterministic scheme can be obtained by application of the general results of [16], with the complication however that in the jump condition (41) for u_k at T_I for $I < N$, the limit $u_k(T_I-, S)$ may not exist at $S = \bar{S}$.

6.2 Simulation Schemes

To solve this problem by simulation, given a time mesh $(t_i)_{0\leq i\leq n}$ *refining the tenor* $\mathfrak{T}$ and a given pair (S_0, H_0) at time 0, we generate a stochastic grid $(S_i^j, H_i^j)_{0\leq i\leq n}^{1\leq j\leq m}$ by an Euler scheme for S, using past values of S to fill H.

Example 6.1 Assuming here for simplicity $r = q = \eta = 0$ and $\sigma = 20\%$: and setting $t_{i+1} - t_i = \mathrm{h} =$ six hours (four time-steps per day):
• Simulate, starting from S_0 given, $S_{6\mathrm{h}} = S_0(1 + \sigma\sqrt{\mathrm{h}}\varepsilon_1)$, $S_{12\mathrm{h}} = S_{6\mathrm{h}}(1 + \sigma\sqrt{\mathrm{h}}\varepsilon_2)$, $S_{18\mathrm{h}} = .., S_{24\mathrm{h}}, S_{30\mathrm{h}}, ..., S_{100days}$ for standard IID Gaussian random variables ε_i;
• Whenever t_i coincides with one the T_Is (i.e., one every fourth i), update the variable H,

so: H_0 =number of consecutive monitoring dates with $S \geq \bar{S}$ viewed from time 0 backwards, $H_{6\mathrm{h}} = H_0$, $H_{12\mathrm{h}} = H_0$, $H_{18\mathrm{h}} = H_0$, $H_{1day} = H_0 + 1$ if $S_{1day} \geq \bar{S}$, otherwise $H_{1day} = 0$, etc..until $H_{100days}$;
• Redo this $m = 10^5$ times, hence 10^5 trajectories $(S_i, H_i)_{0\leq i\leq n}^{1\leq j\leq m}$.

The analog to the algorithms of sections 5.2 and 5.2.1 may then be formulated, ϑ_1^j therein being now understood as (cf. (39))

$$\vartheta_1^j = \inf\{i \in \mathbb{N}_n \,;\, H_i^j \geq l\} \wedge T \,, \tag{43}$$

and $\mathbb{E}_i^j$ in (36) being now understood as the conditional expectation given $t = t_i, S_i = S_i^j, H_i = H_i^j$. For computing $\mathbb{E}_i^j$, one can then perform, for every $k \in \mathbb{N}_{l-1}$, a non-linear regression of the $(u_{i+1}^j + \mathrm{h}c_{i+1}^j)_{j\in\Omega_i^k}$ against $(S_i^j)_{j\in\Omega_i^k}$, where Ω_i^k denotes the subset of the indices j such that $i < \vartheta_1^j$ and $H_i^j = k$. Denoting by $\rho_i(\cdot, k)$'s the estimators of the maps $S \mapsto u_k(t_i, S)$'s obtained in this way, one then sets (cf. (36)),

$$u_i^j = \max\left(\ell_i\left(S_i^j\right),\, e^{-\mu_i^j \mathrm{h}}\rho_i(S_i^j, N_i^j)\right) \,.$$

For the indices i, k's such that the set Ω_i^k is empty or too small for the non-linear regression over Ω_i^k to be doable or significant, the 'missing regression functions' $\rho_i(\cdot, k)$'s are set equal to zero.

This procedure for computing the conditional expectations in (36) can be interpreted as using a method of cells in the direction of the k variable and whatever method of choice in the direction of the S variable, a method of cells again being a simple and robust alternative, for estimating the protection pricing function $u = u_k(t, S)$ (see the Appendix).

Convergence of the scheme is covered by the results of [14].

6.3 Numerical Experiments

Using the same data as in section 5.3 but a path-dependent call protection with daily monitoring $T_{I+1} - T_I$ = one day, Tables 7 and 8 show the bond prices computed in two ways:
• First, by the MC scheme of section 6.2 (MC_l in the tables below), where the conditional expectations are estimated by a method of cells in S and N,
• Second, by solving numerically the cascade of VIs of section 6.1 (VI_l in the tables below).
The simulation results appear to be quite accurate.

l	1	5	10	20	30
Price MC_l	103.823	105.302	106.211	107.169	107.882
Price VI_l	103.874	105.105	106.066	107.273	108.021
%Err	0.05	0.19	0.14	0.10	0.13

Table 7: *Path-Dependent Call Protection: MC Forward price versus VI Cascade price for five values of l ($S_0 = 100.5$)*

l	1	5	10	20	30
Price MC_l	103.738	105.300	106.330	107.528	108.213
Price VI_l	103.693	105.103	106.172	107.545	108.543
%Err	0.04	0.19	0.15	0.02	0.30

Table 8: *Same as Table 8 for* $S_0 = 102.55$

7 Highly Path dependent Call Protection

Given a further integer d such that $l \leq d \leq N$, let H_t for $t \leq T$ represent the vector of the indicator functions of the events $S_{T_I} \geq \bar{S}$ at the last d monitoring dates preceding time t. We now consider the case:

$$\vartheta_1 = \inf\{t \in \mathbb{R}_+ \, ; |H_t| \geq l\} \wedge T \tag{44}$$

with

$$|H_t| = \sum_{1 \leq k \leq d} H_t^k \ . \tag{45}$$

So ϑ_1 represents the first time, capped at T, such that $S \geq \bar{S}$ on at least l among the last d monitoring dates.

Remark 7.1 For $l = 0$, resp. $l = d$, we are back to the no call protection case of section 4, resp. to the l consecutive monitoring dates call protection of section 6. Moreover, in case of a discretization time grid defined by the call protection monitoring time $\mathfrak{T}$, the case $l = 1$ corresponds to a time-discretized version of the standard call protection of section 5.

7.1 Pricing Equation

Let us set $K = \{k \in \{0,1\}^d \, ; \sum_{1 \leq p \leq d} k_p < l\}$. By application of the results of [16], we now have $\Pi_t = u(t, S_t, H_t)$ on $[0, \vartheta_1)$, for a protection pricing function $u = u(t, S, k) = u_k(t, S)$, with $k \in K$, where the restrictions of the u_ks to every set $[T_{I-1}\, T_I) \times [0, +\infty)$ are continuous, and where the limit $u_k(T_I-, S)$ exists in the sense of (40) exists for every $k \in K$, $I \geq 1$ and $S \neq \bar{S}$. Moreover, u solves the following Cauchy-Dirichlet cascade of VIs, in which v denotes the no-protection pricing function of section 4.1, and with

$$k_+ = k_+(k, S) = (\mathbf{1}_{S \geq \bar{S}}, k_1, \ldots, k_{d-1}) \tag{46}$$

in the jump condition (47):

For I decreasing from N until 1:
• At $t = T_I$, for $k \in K$,

$$u_k(T_I-, S) = u_{k_+}(T_I, S), \text{ or } v(T_I, S) \text{ if } k_+ \notin K, \text{ for every } S \neq \bar{S} \ , \tag{47}$$

Or, in case $I = N$, $u_k(T-, S) = v(T, S) = g(S)$ for every $S > 0$,
• On the time interval $[T_{I-1}, T_I)$,

$$\max\left(\mathcal{G}u + c - \mu u \, , \ \ell - u\right) = 0 \ . \tag{48}$$

Like in section 6.1, this cascade of VIs can in theory be solved by standard deterministic numerical schemes, knowing an approximation (v_i^j) of v. (Partial) convergence results for such schemes can be obtained by application of the general results of [16].

However note that this is a VIs cascade of size $\mathrm{Card}(K)$ $(= 2^d - 1$ in case $l = d)$, which precludes in practice the use of such deterministic schemes for l and d more than a few units.

7.2 Simulation Schemes

To solve this system by simulation, given a pair (S_0, H_0) and a time mesh $(t_i)_{0\leq i\leq n}$ refining the tenor (T_I), we generate a stochastic grid $(S_i^j, H_i^j)_{0\leq i\leq n}^{1\leq j\leq m}$ in the obvious way, using past values of S to fill H.

Example 7.2 In the set-up of Example 6.1, and for $d = 30$:
• Simulate $S_0, S_{6\mathrm{h}}, \ldots, S_{100days}$ as before;
• Whenever t_i coincides with one the T_Is (i.e., one every fourth i), update the vector H, so: $H_0 = 30$ given values of S (representing past values of S corresponding to the 30 last monitoring dates T_I preceding the pricing time $t = 0$), $H_{6\mathrm{h}} = H_0$, $H_{12\mathrm{h}} = H_0$, $H_{18\mathrm{h}} = H_0$, $H_{1day} = (\mathbf{1}_{S_{1day}\geq \bar{S}}, \bar{H}_0)$ where $\bar{H}_0$ stands for the vector made of the components 1 to 29 of H_0, $H_{30\mathrm{h}} = H_{1day}$, and so on until $H_{100days}$;
• Redo this $m = 10^5$ times, hence 10^4 trajectories $(S_i, H_i)_{0\leq i\leq n}$.

The exact analogs to the algorithms of section 6.2 may then be formulated, with $|H_i^j|$ instead of H_i^j, and $\mathbb{E}_i^j$ as the conditional expectation given $t = t_i, S_i = S_i^j, H_i = H_i^j$. For computing the related conditional expectations in (36), one can perform, for every $k \in K$, a non-linear regression of $(u_{i+1}^j + \mathrm{hc}_{i+1}^j)_{j\in\Omega_i^k}$ against$(S_i^j)_{j\in\Omega_i^k}$, where Ω_i^k denotes the subset of the indices j such that $i < \vartheta_1^j$ and $H_i^j = k$. Denoting by $\rho_i(S, k)$'s the estimators of the $u_k(t_i, S)$'s obtained in this way, one then sets (cf. (36))

$$u_i^j = \max\left(\ell_i\left(S_i^j\right),\, e^{-\mu_i^j \mathrm{h}}\rho_i(S_i^j, H_i^j)\right)\ .$$

In case Ω_i^k is empty or too small for performing a non-linear regression with respect to S over Ω_i^k, the 'missing estimates' $\rho_i(S, k)$'s are set equal to zero.

Of course it is rather clear in this new setting that for large (but typical, like $d = 30$) values of d, values of k for which Ω_i^k is empty or too small will become the rule rather than the exception (since there are now potentially up to 2^d possible states of the vector k). However this is in a sense the power of the simulation approach, which automatically selects the most likely states of the vector k relative to a starting point (S_0, H_0), as opposed to the deterministic scheme, which loops over *all the possible states* of k.

Convergence and convergence rates are granted by application of the results of Crépey [16].

7.3 Numerical Experiments

We first test the simulation pricing scheme of section 7.2 in cases which are reducible (see Remark 7.1) to no call protection ($l = 0$, section 7.3.1), standard call protection ($l = 1$ and $T_{I+1} - T_I =$ one day, section 7.3.2) or path-dependent call protection ($l = d$, section

7.3.3). The Monte Carlo results can then be validated by VI results, included for 'large' d's (up to a typical market value of $d = 30$). Then in section 7.3.4 we present examples in the general case where $1 < l < d$ for d up to 10. In the general case where $1 < l < d$, VI results are not available for $d > 10$ due to the curse of dimensionality. So, for $d = 10$, the numerical solution of the VIs cascade of section 7.3.4 takes about four days of computation on a standard PC, versus about twenty minutes by simulation. Twenty minutes is of course already much too long, but there is lots of room for improvement here, thinking in particular of massively parallel computation techniques ('graphics card programming') which could be fruitfully used for this purpose. In any case deterministic methods are ruled out by the curse of dimensionality, and simulation methods are the only viable computational alternative.

7.3.1 Case reducible to No Call Protection

In case $l = 0$ the 'l out of d' call protection clause effectively reduces to 'no call protection'. Using the data of Section 4.3, we thus computed the related MC prices by the simulation pricing scheme of section 7.2 ($\mathrm{MC}_{l,d}$ in the tables), and we validated the results by the deterministic scheme (27). The results are given in Table 9.
Note that in the situation of this section, the deterministic scheme (27) and the one of section 7.1 produce the same numbers, but in time independent of d for (27) versus exponential in d for the scheme of section 7.1.

In Table 10, we displayed the average computation times for the MC results of Table 9, versus those ($\mathrm{VI}_{l,d}$ in the table) for the scheme of section 7.1 (at least, for $d \leq 10$, since for greater values of d, computations via the scheme of section 7.1 become prohibitively long).

S_0	$\mathrm{MC}_{0,d}$	$d = 1$	5	10	20	30
98.55	0.04	0.04	0.04	0.04	0.04	0.04
99.55	0.01	0.02	0.02	0.01	0.02	0.02
100.55	0.01	0.05	0.05	0.01	0.05	0.05
101.55	0.07	0.07	0.07	0.08	0.07	0.07

Table 9: *% Err MC No Call Protection versus $MC_{l,d}$ for various '$l = 0$ out of d' cases.*

d	1	5	10	20	30
$\mathrm{VI}_{0,d}$	332s	5332s	44h	—	—
$\mathrm{MC}_{0,d}$	154s	212s	313s	474s	628s

Table 10: *$VI_{l,d}$ versus $MC_{l,d}$ Computation Times corresponding to Table 9.*

7.3.2 Case reducible to a Standard Call Protection

In case $l = 1$ and $T_{I+1} - T_I = \frac{1}{4}$ day, the 'l out of d' call protection clause effectively reduces to standard protection. The deterministic scheme (35) and the one of section 7.1 therefore produce the same numbers, but in time independent of d for (27) versus exponential in d for the scheme of section 7.1.

Using the data of Section 5.3 and $\bar{S} = 103$, we computed the MC prices by the simulation pricing scheme of section 7.2, and we validated the results by the deterministic scheme (35). The results are given in Table 11.

In Table 12 we displayed the average computation times for the MC results of Table 9, versus those of the scheme of section 7.1 (whenever practical).

S	VI Price	$d = 1$	5	10	20	30
100.55	103.874	0.053	0.041	0.025	0.045	0.045
101.55	103.786	0.022	0.026	0.045	0.011	0.028
102.55	103.694	0.021	0.071	0.036	0.005	0.019
103.55	103.55	0.000	0.000	0.000	0.000	0.000

Table 11: *VI Prices and $MC_{l,d}$ % Errors for various 'l = 1 out of d' cases.*

d	1	5	10	20	30
VI	737s	11125s	49 h	—	—
$MC_{l,d}$	342s	526s	702s	1111s	1518s

Table 12: *VI versus $MC_{l,d}$ Computation Times corresponding to Table 11.*

7.3.3 Case reducible to a Path-Dependent Call Protection

Using the data of section 6.3, Tables 13 and 14 show the relative error with respect to the VIs cascade prices of section 6.1 for two MC methods: the path-dependent MC method of section 6.2 (cf. Tables 7, 8) versus the highly path-dependent MC method of section 7.2 (with $d = l$ therein).

l	1	5	10	20	30
%Err1	0.05	0.19	0.14	0.10	0.13
%Err2	0.03	0.03	0.15	0.64	0.41

Table 13: *Path-Dependent Call Protection: Error with respect to the VI_l price by MC_l versus $MC_{l,l}$ ($S_0 = 100.5$).*

Table 15 gives the computation times corresponding to Tables 7 and 13.

7.3.4 General Case

We now consider the general case where $1 < l < d$. We use the data of section 6.3, except for the nature of the call protection.

Tables 16 ($d = 5$) and 17, 18 ($d = 10$) show the relative error of the MC price with respect to the VIs cascade price of section 7.1, using two alternative methods for estimating the conditional expectations involved in the MC prices, based on simulated trajectories of the process (S, H) as of section 7.2:

l	1	5	10	20	30
%Err MC_l	0.04	0.19	0.15	0.02	0.30
%Err $\mathrm{MC}_{l,l}$	0.04	0.03	0.24	0.53	1.02

Table 14: *Same as Table 13 for* $S_0 = 102.55$.

l	1	5	10	20	30
VI	8m	21m	38m	72m	105m
MC_l	212s	221s	275s	294s	332s
$\mathrm{MC}_{l,l}$	173s	252s	374s	721s	1008s

Table 15: *Computation times relative to Tables 7 and 13.*

$\mathrm{MC}_{l,d}$ Conditional expectations computed by a method of cells in (S, H) as described in section 7.2,

%Err $\mathrm{MC}^{\sharp}_{l,d}$ Conditional expectations computed by a method of cells in $(S, |H|^{\sharp})$, where $|H|^{\sharp}$ is defined as the number of ones in H, starting from the $(l - |H|)^{th}$ zero in H.

For instance, assuming $d = 10$, $l = 8$:
• If $H = (1, 1, 1, 1, \mathbf{0}, 1, 1, 1, 0, 0)$, then $l - |H| = 8 - 7 = 1$ and $|H|^{\sharp} = 3$ (number of ones on the right of the first zero, in bold in H),
• If $H = (1, 1, 1, 0, 1, 1, 1, \mathbf{0}, 0, 0)$, then $l - |H| = 8 - 6 = 2$ and $|H|^{\sharp} = 0$ (number of ones on the right of the second zero, in bold in H).

The $\mathrm{MC}^{\sharp}_{l,d}$-algorithm can be thought of as an approximate algorithm based on the 'good regressor' $|H|^{\sharp}$ for estimating highly path-dependent conditional expectations. The rationale for using the regressor $|H|^{\sharp}$ is that in the 'l out of d' case, the entries of H preceding its $(l - |H|)^{th}$ zero are irrelevant to the price, since these entries will necessarily have to be superseded by new ones before the bond may become callable. This approximate algorithm in the highly path-dependent case is in fact inspired by the 'exact algorithm' of section 6.2 in the path-dependent case. Note in particular that in case $l = d$, a highly path-dependent call protection reduces to a path-dependent call protection, $|H|^{\sharp} = |H|$, and the $\mathrm{MC}^{\sharp}_{l,d}$-algorithm of this section reduces to the MC_l-algorithm of section 6.2.

In view of the results of Tables 16, 17 and 18, the 'approximate' $\mathrm{MC}^{\sharp}_{l,d}$-algorithm appears to be reasonably accurate.
In case $l = d = 10$ it actually gives more accurate results than the 'exact' $\mathrm{MC}_{l,d}$ algorithm. This makes sense, since, in the case $l = d$, high path-dependence reduces to path-dependence, and both algorithms are 'exact'. But the $\mathrm{MC}^{\sharp}_{l,d}$-algorithm (alias MC_l, in the case $l = d$) works in a lower state-dimension, which can explain its more accurate results.
In the general 'l out of d' case, the interest of the $\mathrm{MC}^{\sharp}_{l,d}$-algorithm with respect to the $\mathrm{MC}_{l,d}$ Cell algorithm is of course that it is faster (see Table 15).

The ability to work with a 'good approximate' (as opposed to 'exact'), low-dimensional regressor, is also an interesting feature of simulation as opposed to deterministic numerical schemes.

In the general 'l out of d' case, as soon as d exceeds a few units, $\mathrm{VI}_{l,d}$ prices cannot be computed, so that the accuracy of $\mathrm{MC}_{l,d}$ and $\mathrm{MC}^{\sharp}_{l,d}$ prices cannot be assessed anymore. In

l	1	3	5
Price $\mathrm{VI}_{l,d}$	103.693	104.434	105.103
%Err $\mathrm{MC}_{l,d}$	0.048	0.062	0.011
%Err $\mathrm{MC}^{\sharp}_{l,d}$	0.048	0.090	0.187

Table 16: *Error as a function of* l *(*$d = 5, S_0 = 102.55$*).*

l	1	5	10
Price $\mathrm{VI}_{l,d}$	103.693	104.906	106.172
%Err $\mathrm{MC}_{l,d}$	0.045	0.021	0.236
%Err $\mathrm{MC}^{\sharp}_{l,d}$	0.043	0.054	0.149

Table 17: *Error as a function of* l *(*$d = 10, S_0 = 102.55$*).*

case $d = 30$ and for increasing values of l, we compare in Table 19 the $\mathrm{MC}_{l,d}$ and the $\mathrm{MC}^{\sharp}_{l,d}$ estimates in terms of standard deviations over 50 trials corresponding to different seeds of the random generator. The Table also displays the % Err, in the sense here of the relative difference between the average estimates obtained by the two methods.

8 Intermittent Call Protection

We now come to 'truly intermittent' protection with call payoff processes of the form, given a non-decreasing sequence of $[0, T]$-valued stopping times $\vartheta = (\vartheta_l)_{l \geq 0}$:

$$U_t = \Omega^c_t \infty + \Omega_t h(t, X_t) \ , \tag{49}$$

with $\Omega_t = \mathbf{1}_{\{l_t \text{ odd}\}}$ for l_t defined by $\vartheta_{l_t} \leq t < \vartheta_{l_t+1}$, rather than more specifically (cf. (31))

$$U_t = \mathbf{1}_{\{t < \vartheta_1\}} \infty + \mathbf{1}_{\{t \geq \vartheta_1\}} h(t, X_t) \tag{50}$$

for a stopping time ϑ_1 as in the previous sections.

In the benchmark model of section 2 and assuming

$$\Omega_t = \Omega(t, S_t, \mathcal{S}_t) \tag{51}$$

for a suitably extended finite-dimensional Markovian factor process $(S_t, \mathcal{S}_t)$ and a Boolean function Ω of $(t, S, \mathcal{S})$, so

$$U_t = U(t, S_t, \mathcal{S}_t) := \Omega^c(t, S_t, \mathcal{S}_t) \infty + \Omega(t, S_t, \mathcal{S}_t) h(t, S_t) \ , \tag{52}$$

l	1	5	10
Price $\mathrm{VI}_{l,d}$	103.874	104.931	106.066
%Err $\mathrm{MC}_{l,d}$	0.026	0.056	0.154
%Err $\mathrm{MC}^{\sharp}_{l,d}$	0.049	0.065	0.138

Table 18: *Error as a function of* l *(*$d = 10, S_0 = 100.55$*).*

l	5	10	20	30
Dev $\mathrm{MC}_{l,d}$	0.081	0.089	0.126	0.251
Dev $\mathrm{MC}^{\sharp}_{l,d}$d	0.052	0.102	0.173	0.165
% Err	0.11	0.34	0.88	1.31

Table 19: *Standard Deviations over 50 trials and % Err : $MC_{l,d}$ vs $MC^{\sharp}_{l,d}$ as a function of l ($d = 30, S_0 = 102.55$).*

it is expected that one should then have $\Pi_t = u(t, S_t, \mathcal{S}_t)$ on $[0, T]$, for a pricing function $u = u(t, S, \mathcal{S})$. This is precisely what comes out from the results of [16] in case of a call protection discretely monitored at the dates of a finite time grid $\mathfrak{T}$.

Generic Simulation Scheme Given a stochastically generated mesh $(S^j_i, \mathcal{S}^j_i)^{1\le j\le m}_{0\le i\le n}$, the generic simulation pricing scheme of [14] for estimating $u(t_i, S^j_i, \mathcal{S}^j_i)^{1\le j\le m}_{0\le i\le n}$ writes: $u_n = g$, and then for $i = n-1 \dots 0$, for $j = 1 \dots m$

$$u^j_i = \min\left(U_i\left(S^j_i, \mathcal{S}^j_i\right)\,,\ \max\left(\ell_i\left(S^j_i\right),\, e^{-rh}\mathbb{E}^j_i\big(u_{i+1} + \mathrm{hc}_{i+1}\big)\right)\right)\,. \tag{53}$$

Here the conditional expectations $\mathbb{E}^j_i\big(u_{i+1} + \mathrm{hc}_{i+1}\big)$ stand for the conditional expectations given $t = t_i, S_i = S^j_i, \mathcal{S}_i = \mathcal{S}^j_i$, which can be computed by non-linear regression of $(u_{i+1} + \mathrm{hc}_{i+1})_{1\le j\le m}$ against $(S_i, \mathcal{S}_i)_{1\le j\le m}$, using for example a method of cells in $(S, \mathcal{S})$.

Note that in (53) the min plays no role outside the support of Ω, where $U_i\,(S, \mathcal{S})$ is equal to $+\infty$.

8.1 Intermittent Standard Protection

We first consider the 'intermittent analogue' of the standard protection clause of section 5. Given activating and desactivating trigger levels $\bar{S}$ and $\underline{S}$ with $\bar{S}, \underline{S} < \bar{S}$, let thus the non-decreasing sequence ϑ of stopping times be defined by $\vartheta_0 = 0$ and, for every $l \ge 0$:

$$\vartheta_{2l+1} = \inf\{t > \vartheta_{2l}\,;\, S_t \ge \bar{S}\} \wedge T\,,\ \vartheta_{2l+2} = \inf\{t > \vartheta_{2l+1}\,;\, S_t \le \underline{S}\} \wedge T\,. \tag{54}$$

Remark 8.1 For $\underline{S} = \bar{S}$, (54) would typically not define increasing sequences of stopping times, and one would get an ill-posed problem. This is why we assume $\underline{S} < \bar{S}$.

8.1.1 Pricing Equation

Let H_t be defined as the parity of l_t, i.e., $H_t = 0$ whenever l_t is even ('call protection'), and $H_t = 1$ whenever l_t is odd ('no call protection'). So $H_0 = 0$, H jumps from 0 to 1 at the ϑ_ls such that $S_{\vartheta_l} \ge \bar{S}$ and $H_{\vartheta_l -} = 0$, and H jumps from 1 to 0 at the ϑ_l's such that $S_{\vartheta_l} \le \underline{S}$ and $H_{\vartheta_l -} = 1$. In particular one has $H_t = 1$ on $\{(\omega, t); S_t \ge \bar{S}\}$ and $H_t = 0$ on $\{(\omega, t); S_t \le \underline{S}\}$.

The pair (S, H) is a Markov process, with an effective upper barrier process U of the form (51), for $\Omega(t, S, \mathcal{S}) = \mathcal{S}$. It is thus expected that $\Pi_t = u(t, S_t, H_t)$, for a pricing function $u = u(t, S, k) = u_k(t, S)$, with $k \in \{0, 1\}$.

Note however that the call protection ϑ is continuously monitored in time, and not discretely monitored at the dates of a finite time grid $\mathfrak{T}$. Continuously monitored intermittent call protection clauses falls outside the scope of the results of [16]. However, existence (uniqueness is easy) of a solution to the related RIBSDE $(\mathcal{E})$ follows by application of the results of Peng and Xu [29]. More precisely, existence follows from an immediate extension of these results to the case of an $\mathbb{R} \cup \{+\infty\}$ – valued upper barrier U, noting that the results of Peng and Xu, even if stated for real-valued barriers, only use the square-integrability of the random variable $\sup_{t\in[0,T]} U_t^-$, a condition which is satisfied in our case (note $U_t^- = h(t, X_t)^-$).

Building upon this existence result, one can then proceed much like in [16] to deduce that $\Pi_t = u(t, S_t, \mathcal{S}_t)$ on $[0, T]$, for a pricing function $u = u(t, S, \mathcal{S})$. The formally related analytic problem writes,

$$\begin{cases} u = g \text{ at } T \\ u_0(t, \bar{S}) = u_1(t, \bar{S}) \\ \max\left(\mathcal{G}u_0 + c - \mu u_0, \ell - u_0\right) = 0 \text{ on } [0, T) \times (0, \bar{S}) \\ u_1(t, \underline{S}) = \min(u_0(t, \underline{S}), h(t, \underline{S})) \\ \min\left(\max\left(\mathcal{G}u_1 + c - \mu u_1, \ell - u_1\right), h - u_1\right) = 0 \text{ on } [0, T) \times (\underline{S}, +\infty) \end{cases} \tag{55}$$

We thus get a system of two equations in the pair of functions (u_0, u_1), where u_0 and u_1 are defined on $[0, T] \times (0, \bar{S}] \times \{0, 1\}$ and $[0, T] \times [\underline{S}, +\infty) \times \{0, 1\}$, respectively. From a practical point of view this system of equations can be solved by standard finite differences deterministic numerical schemes on a fixed time-space grid, like (cf. (35)):
$u_n^{j,0} = u_n^{j,1} = g(S_n^j)$ for $j = 1 \dots m$, and then for $i = n - 1 \dots 0$, for $j = 1 \dots m$:

$$\begin{cases} u_i^{j,0} = \max\left(\ell_i\left(S^j\right),\ e^{-\mu_i^j \mathrm{h}} \sum_l p_i^{j,l} (u_{i+1}^{j+l,0} + \mathrm{h} c_{i+1}^{j+l})\right) \\ u_i^{j,1} = \min\left(h_i\left(S^j\right), \max\left(\ell_i\left(S^j\right),\ e^{-\mu_i^j \mathrm{h}} \sum_l p_i^{j,l} (u_{i+1}^{j+l,1} + \mathrm{h} c_{i+1}^{j+l})\right)\right) \end{cases} \tag{56}$$

where, in the right hand side, $u_{i+1}^{j+l,0}$ is to be understood as $u_{i+1}^{j+l,1}$ for $S^{j+l} \geq \bar{S}$, and $u_{i+1}^{j+l,1}$ is to be understood as $\min(u_{i+1}^{j+l,0}, h_{i+1}^{j+l})$ for $S^{j+l} \leq \underline{S}$.

Remark 8.2 Note however that the analytic characterization of the pricing function $u = (u_0, u_1)$ as unique solution in some sense to (56) (continuous viscosity solution with growth conditions, presumably) is not established yet. This is due to the absence of stability results so far (beyond existence and uniqueness) for the related Markovian RIBSDE $(\mathcal{E})$. To be more precise, the absence of stability results makes it difficult to establish the continuity of the pricing function $u = (u_0, u_1)$, which would be the first step in proving that u solves (55) in the viscosity sense.

8.1.2 Simulation Scheme

The related simulation pricing algorithm is (53), with $U(t, S, k)$ defined by (52) for $\Omega(t, S, k) = k$.

Note however that the convergence of this scheme falls outside the scope of [14], since one deals here with a continuously monitored intermittent form of call protection.

8.2 Intermittent Path-Dependent Protection

Let us further be given a sequence $\mathfrak{T} = (T_I)_{I \leq N}$ of monitoring dates as of sections 6 or 7. The cases considered in section 6, resp. 7, correspond to an upper payoff process of the form (50) for

$$\vartheta_1 = \inf\{t \in \mathbb{R}_+ \, ; \, H_t \geq l\} \wedge T \ , \text{ resp. } \inf\{t \in \mathbb{R}_+ \, ; \, |H_t| \geq l\} \wedge T \ ,$$

where the respective interpretations of H are given in sections 6 and 7. In this section and in section 8.3, we consider the corresponding intermittent call protections, with upper payoff processes of the form (49) for

$$\Omega_t = \mathbf{1}_{\{H_t \geq l\}} \ , \text{ resp. } \mathbf{1}_{\{|H_t| \geq l\}} \ .$$

Let thus for now H_t stand, as in section 6, for the number of consecutive monitoring dates with $S \geq \bar{S}$ from time t backwards, capped at l.

8.2.1 Pricing Equation

One thus has by application of [16] that $\Pi_t = u(t, S_t, H_t)$ on $[0, T]$, for a pricing function $u = u(t, S, k) = u_k(t, S)$ with $k \in \mathbb{N}_l$, where the restrictions of the u_ks to every set $[T_{I-1}\, T_I) \times [0, +\infty)$ are continuous, and where the limit $u_k(T_I-, S)$ as defined by (40) exists for every $k \in \mathbb{N}_l$, $I \geq 1$ and $S \neq \bar{S}$. By application of the results of [16], the pricing equation now assumes the form of the following Cauchy cascade of VIs (to be compared with the Cauchy–Dirichlet cascade of VIs of section 6.1):

For I decreasing from N to 1:
• At $t = T_I$, for $k \in \mathbb{N}_l$,

$$u_k(T_I-, S) = \begin{cases} u_{k+1}(T_I, S), \text{ or } u_k(T_I, S) \text{ if } k = l, \text{ on } \{S > \bar{S}\} \ , \\ u_0(T_I, S), \text{ or } \min(u_0(T_I, S), h(T_I, S)) \text{ if } k = l, \text{ on } \{S < \bar{S}\} \ , \end{cases} \tag{57}$$

Or, in case $I = N$, $u_k(T_I-, S) = g(S)$ for $S > 0$,
• On the time interval $[T_{I-1}, T_I)$,

$$\begin{aligned} &\max\left(\mathcal{G}u_k + c - \mu u_k \, , \ \ell - u_k\right) = 0 \, , \ k = 0 \ldots l-1 \\ &\min\left(\max\left(\mathcal{G}u_l + c - \mu u_l \, , \ \ell - u_l\right), h - u_l\right) = 0 \ . \end{aligned} \tag{58}$$

Once more, this Cauchy problem can be solved by standard deterministic numerical schemes (cf. section 5.1).

8.2.2 Simulation Scheme

The related simulation pricing algorithm is (53), with $U(t, S, k)$ defined by (52) for $\Omega(t, S, k) = \mathbf{1}_{\{k \geq l\}}$.

Convergence and convergence rates are granted by application of the results of [14].

8.3 Intermittent Highly Path-Dependent Protection

Let now H_t be defined, as in section 7, as the vector of the indicator functions of the events $S_{T_I} \geq \bar{S}$ at the last d monitoring dates preceding time t.

8.3.1 Pricing Equation

One has by application of the results of [16] that $\Pi_t = u(t, S_t, H_t)$ on $[0, T]$, for a pricing function $u = u(t, S, k) = u_k(t, S)$ over $[0, T] \times (0, +\infty) \times \{0, 1\}^d$, where the restrictions of the u_ks to every set $[T_{I-1}\, T_I) \times [0, +\infty)$ are continuous, and where the limit $u_k(T_I-, S)$ exists in the sense of (40) for every $k \in \{0, 1\}^d$, $I \geq 1$ and $S \neq \bar{S}$. Moreover,

By application of the results of [16], the pricing equation now assumes the following form , with as before $K = \{k \in \{0, 1\}^d ; \sum_{1 \leq p \leq d} k_p < l\}$, and where $k_+ = k_+(k, S)$ in the jump condition (59) is defined by (46),

For I decreasing from N to 1:
• At $t = T_I$, for every $k \in \{0, 1\}^d$,

$$u_k(T_I-, S) = u_{k_+}(T_I, S), \quad \text{or } \min(u_{k_+}(T_I, S), h(T_I, S)) \text{ if } k \notin K \text{ and } k_+ \in K, \quad \text{for every } S \neq \bar{S}\ , \tag{59}$$

Or, in case $I = N$, $u_k(T_I-, S) = g(S)$ for $S > 0$,
• On the time interval $[T_{I-1}, T_I)$, for every $k \in \{0, 1\}^d$,

$$\max\left(\mathcal{G}u_k + c - \mu u_k\ ,\ \ell - u_k\right) = 0\ ,\ k \in K$$
$$\min\left(\max\left(\mathcal{G}u_k + c - \mu u_k\ ,\ \ell - u_k\right), h - u_k\right) = 0\ ,\ k \notin K\ .$$

Like in section 7.1, this Cauchy cascade of VIs can in theory be solved by standard deterministic numerical schemes, but this is a system of 2^d equations, which precludes the use of deterministic schemes for l and d more than a few units.

8.3.2 Simulation Scheme

The related simulation pricing algorithm is (53), with $U(t, S, k)$ defined by (52) for $\Omega(t, S, k) = \mathbf{1}_{\{|k| \geq l\}}$.
Convergence and convergence rates are covered by the results of [14].

8.4 Numerical Experiments

For the numerical experiments relative to all the forms of intermittent call protection considered above, we shall use the general data and conventions of section 4.3, and the specific data of Table 20.

T	$\bar{c}$	$\bar{S}$	$\underline{S}$
180 days	1.2/month	103	97

Table 20: *Data specific to section 8.4.*

S_0	92	93	94	102	103	104
VI price	104.128	104.147	104.154	103.716	103	104
MC %Err	0.094	0.051	0.012	0.001	0	0

Table 21: *Intermittent standard call protection.*

l	1	5	10	20	30
VI_l price	103.91	105.10	106.03	107.22	108.01
MC_l %Err	0.04	0.16	0.47	0.88	1.34
$\mathrm{MC}_{l,l}$ %Err	0.04	0.15	0.03	0.04	0.24

Table 22: *Intermittent path-dependent call protection ($S_0 = 100$).*

8.4.1 Standard Protection

We first consider convertible bonds with intermittent standard call protection as of section 8.1, that we evaluate by the deterministic scheme of section 8.1.1, or, alternatively, by the (forward variant of the) standard protection Monte Carlo method of section 8.1.2.

The results are presented in Table 21: reference prices computed by the deterministic scheme versus relative errors of the MC method.

8.4.2 Path-dependent Protection

We first consider convertible bonds with intermittent path-dependent call protection as of section 8.2, that we evaluate by the deterministic scheme of section 8.2.1, or, alternatively, the following Monte Carlo methods:

MC$_l$ The path-dependent scheme of section 8.2.2, using a method of cells for the computation of the conditional expectations, or

MC$_{l,d}$ The highly path-dependent scheme of section 8.3.2 with $d = l$, using a method of cells therein for the computation of the conditional expectations.

The results are presented in Tables 22 for $S_0 = 100$ and 23 for $S_0 = 90$: reference prices computed by a deterministic scheme versus relative errors of the various Monte Carlo methods. The highly path-dependent scheme happens to be more accurate than the path-dependent scheme.

l	1	5	10	20	30
VI_l price	104.07	104.50	104.81	105.17	105.37
MC_l %Err	0.04	0.20	0.32	0.40	0.60
$\mathrm{MC}_{l,l}$ %Err	0.098	0.087	0.066	0.007	0.037

Table 23: *Intermittent path-dependent call protection ($S_0 = 90$).*

l	2	3	5
$\mathrm{VI}_{l,d}$ price	104.07	104.43	105.10
$\mathrm{MC}_{l,d}$ %Err	0.21	0.15	0.15
$\mathrm{MC}^{\sharp}_{l,d}$ %Err	0.19	0.23	0.18

Table 24: *Intermittent highly path-dependent call protection ($d = 5, S_0 = 100$).*

l	2	3	5
$\mathrm{VI}_{l,d}$ price	104.10	104.25	104.50
$\mathrm{MC}_{l,d}$ %Err	0.01	0.01	0.10
$\mathrm{MC}^{\sharp}_{l,d}$ %Err	0.155	0.108	0.034

Table 25: *Intermittent highly path-dependent call protection ($d = 5, S_0 = 90$).*

8.4.3 Highly Path-dependent Protection

We now come to convertible bonds with intermittent highly path-dependent call protection as of section 8.3. The bonds are evaluated by the deterministic scheme of section 8.3.1, or, alternatively, by the highly path-dependent Monte Carlo scheme of section 8.3.2. using a method of cells in (S, H) ($\mathrm{MC}_{l,d}$) or in $(S, |H|^{\sharp})$ ($\mathrm{MC}^{\sharp}_{l,d}$) therein for the computation of the conditional expectations (see section 7.3.4 for the definition of $|H|^{\sharp}$).

Results for $S_0 = 100$ or 90 and $d = 5$ or 10 are presented in Tables 24 to 27. For larger values of d, Table 28 compares $\mathrm{MC}_{l,d}$ and $\mathrm{MC}^{\sharp}_{l,d}$ in terms of standard deviations over 50 trials corresponding to different seeds of the random generator, and of relative difference.

A Computing Conditional Expectations by Simulation/Regression

Pricing game options by simulation ultimately reduces to the numerical computation of conditional expectations. An embedded issue is therefore the computation of conditional expectations by simulation, which can simply be done by a combination of simulation and regression tools. In this Appendix we provide a brief and informal review about this, referring the interested reader to, for instance, Chapter 6 of Glasserman [19], for more details and references.

Let ξ and X denote real- and $\mathbb{R}^q$-valued square integrable random variables. Under suitable conditions, the conditional expectation $\mathbb{E}(\xi|X)$ is equal to the Hilbert space $\mathcal{L}^2$-projection of ξ over the vector space of random variables $\langle X \rangle$ spanned by the measurable and bounded functions of X. So, in terms of a basis $(\varphi^l)_{l\in\mathbb{N}}$ of the set of the functions from $\mathbb{R}^q$ to $\mathbb{R}$,

$$\mathbb{E}(\xi|X) = \mathbb{E}\mathbb{L}(\xi|(\varphi^l(X))_{l\in\mathbb{N}}) \ ,$$

l	2	5	10
$\mathrm{VI}_{l,d}$ price	104.27	104.87	106.03
$\mathrm{MC}_{l,d}$ %Err	0.01	0.15	0.03
$\mathrm{MC}^{\sharp}_{l,d}$ %Err	0.04	0.26	0.38

Table 26: *Intermittent highly path-dependent call protection ($d = 10, S_0 = 100$).*

l	2	5	10
$\mathrm{VI}_{l,d}$ price	104.24	104.41	104.82
$\mathrm{MC}_{l,d}$ %Err	0.01	0.02	0.07
$\mathrm{MC}^{\sharp}_{l,d}$ %Err	0.05	0.09	0.32

Table 27: *Intermittent highly path-dependent call protection* ($d = 10, S_0 = 90$).

l	5	10	20	30
Dev $\mathrm{MC}_{l,d}$	0.056	0.061	0.086	0.152
Dev $\mathrm{MC}^{\sharp}_{l,d}$	0.060	0.069	0.092	0.175
% Err	0.09	0.24	0.72	1.06

Table 28: *Intermittent Standard Deviations over 50 trials and % Err :* $MC_{l,d}$ *vs* $MC^{\sharp}_{l,d}$ *as a function of* l ($d = 30, S_0 = 102.55$).

where $\mathbb{EL}$ stands for the $\mathcal{L}^2$-projection operator. Given pairs $(X^j, \xi^j)^{1\leq j\leq m}$ simulated independently according to the law of (X, ξ), the conditional expectation $\mathbb{E}(\xi|X)$ may thus be simulated by *linear* regression of the ξ^js against the $(\varphi^l(X^j))^{1\leq j\leq m}_{1\leq l\leq p}$, a procedure called *non-linear regression* in the sequel, where the truncation order p is a parameter in the method. The computational cost of the regression is of the order of $\mathrm{O}(mp^2)$.

We refer the interested reader to the monograph by Györfi et al. [21] for every detail about these simulation/regression approaches for computing conditional expectations, or, more precisely in the terminology of [21], *regression functions*

$$x \mapsto \rho(x) = \mathbb{E}(\xi|X = x) \ .$$

In a nutshell, the (truncated) regression basis may be:
• Either *parametric*, i.e., made of functions parameterized by a few parameters, or *non-parametric*, meaning in practice that it is made of functions parameterized by a very large set, like one function by point of a discretization of the state space, and
• Either *global*, that is, made of functions supported by the whole state space or with 'large' support, or *local*, to be understood as made of functions with 'small' support.
One typically deals with either a *parametric and global* regression basis, like a regression basis made of a few monomials parameterized by their coefficients, or a *non-parametric and local* basis, like a regression basis made of the indicator functions of the cells of a grid of hyperrectangles partitioning the state space. Statistic theory tells us that a global basis is preferable in case of a 'regular' regression function $\rho(x)$, especially in case where a good guess is available as for the shape (used to define the regression basis) of ρ. Otherwise a local basis is better (and it is often simpler and more robust in terms of implementation).

In the context of mathematical finance, these approaches were successfully introduced in the late 90's for pricing American options by simulation, by Longstaff and Schwartz [25] (in combination with iteration on the stopping policies as of section 3.2.1), Tsitsiklis and VanRoy [31, 32] (with iteration on the values) or Broadie and Glasserman [10, 11], among others. They were subsequently developed by, for instance, Gobet et al. [20, 23], in a more general context of numerical simulation methods for BSDEs. The latter references deal in particular with the issue of controlling the *cumulative* regression approximation error

that arises in the context of a dynamic model in which non-linear regressions in space are performed repeatedly over a discrete time grid.

Remark A.1 Regarding alternative methods for computing conditional expectations in the context of pricing by simulation: *Malliavin Calculus* methods, *quantization* methods, etc., we refer the reader to, for instance, Lions and Régnier [24], Bouchard et al. [6], or Pagès and Bally [27].
Note however that Malliavin Calculus methods are typically harder to implement than non-linear regression methods, and that quantization methods suffer significantly the curse of dimensionality. Therefore we only resort in this paper to non-linear regression methods as exposed above, either parametric and global with respect to a simple polynomial basis, or non-parametric and local with respect to a basis of hypercubes partitioning the state space, referred to for the latter as *methods of cells* in this article.

References

[1] Bally, V. and Pagès, G.: Error analysis of the quantization algorithm for obstacle problems, *Stochastic Processes and their Applications*, 106, 1–40, 2003.

[2] Bielecki, T.R., Crépey, S., Jeanblanc, M. and Rutkowski, M.:Arbitrage pricing of defaultable game options with applications to convertible bonds. *Quantitative Finance*, Vol. 8, Issue 8 (December 2008) , pp. 795–810 (Preprint version available online at `www.defaultrisk.com`).

[3] Bielecki, T.R., Crépey, S., Jeanblanc, M. and Rutkowski, M.:Defaultable options in a Markovian intensity model of credit risk. *Mathematical Finance*, vol. 18, pp. 493-518, 2008 (Updated version available online at `www.defaultrisk.com`).

[4] Bielecki, T.R., Crépey, S., Jeanblanc, M. and Rutkowski, M.:Convertible bonds in a defaultable diffusion model. *Working Paper* (available online at `www.defaultrisk.com`).

[5] Bouchard, B. and Chassagneux, J.-F.: Discrete time approximation for continuously and discretely reflected BSDE's. *Stochastic Processes and Applications*, 118(12), pp 2269-2293.

[6] Bouchard B., Ekeland I. and Touzi N.: The Malliavin approach to Monte Carlo approximations to conditional expectations. *Finance and Stochastics*, Volume 8, Number 1, 45–71, 2004.

[7] Bouchard, B. and Elie, R. Discrete time approximation of decoupled Forward-Backward SDE with jumps, *Stochastic Processes and their Applications*, 2008, 118 (1), 53-75.

[8] Bouchard, B. and Menozzi, S.: Strong Approximations of BSDEs in a domain. To appear in *Bernoulli*.

[9] Bouchard, B. and Touzi, N.: Discrete-time approximation and Monte Carlo simulation of backward stochastic differential equations. *Stochastic Processes and their Applications* 111(2) (2004), 175–206.

[10] M. Broadie P. Glasserman. Pricing American-style securities using simulation, *J. of Economic Dynamics and Control*, 21, 1323–1352, 1997.

[11] M. Broadie P.Glasserman. A Stochastic Mesh method for Pricing High-Dimensional American Options, *Journal of Computational Science* 7, pp. 35-72, 2004.

[12] Chassagneux, J.-F.: Processus réfléchis en finance et probabilité numérique. *PhD Thesis* Université Paris Diderot – Paris 7, 2008.

[13] Chassagneux, J.-F.: Discrete time approximation of doubly reflected BSDEs. *Advances in Applied Probability*, 41, pp. 101 - 130,2009.

[14] Chassagneux, J.-F. and Crépey, S.: Doubly reflected BSDEs with Call Protection and their Approximation. *In Preparation.*

[15] Chassagneux, J.-F., Crépey, S. and Rahal, A.: Pricing Convertible Bonds with Call Protection. *In Preparation.*

[16] Crépey, S.: About the pricing equations in finance. Forthcoming in *Paris-Princeton Lectures in Mathematical Finance*, Lecture Notes in Mathematics, Springer.

[17] Duffy, D.J.: Financial Difference Methods in Financial Engineering, *Wiley*, 2006.

[18] El Karoui, N., Kapoudjian, E., Pardoux, C., Peng, S., and Quenez, M.-C.: Reflected solutions of backward SDEs, and related obstacle problems for VIs. *Annals of Probability* 25 (1997), 702–737.

[19] Glasserman, P:. Monte Carlo Methods in Financial Engineering, *Springer*, 2004.

[20] Gobet, E., Lemor, J.-P. and Warin, X.: A regression-based Monte-Carlo method for backward stochastic differential equations, *Ann. Appl. Probab.*, 15, 2172-2202, 2005.

[21] Györfi, L., Kohler, M., Krzyzak, A., Walk, H.: A Distribution-Free Theory of Nonparametric Regression, *Springer*, 2002.

[22] Lemor, J.-P.: Approximation par projections et simulations de Monte-Carlo des équations différentielles stochastiques rétrogrades. *PhD Thesis* Ecole Polytechnique (in French), 2005.

[23] Lemor, J.-P., Gobet, E. and Warin, X.: Rate of convergence of an empirical regression method for solving generalized backward stochastic differential equations, *Bernoulli*, Volume 12, Number 5, 2006.

[24] Lions, P.-L., Regnier, H.: Calcul du prix et des sensibilités d'une option américaine par une méthode de Monte Carlo, *Working Paper*, 2001.

[25] Longstaff, F.A., Schwartz, E.S.: Valuing American Options by simulations: A Simple Least-Squares Approach, *Review of Financial Studies*, Volume 14, Number 1, 2001, pp. 113-147(35).

[26] Morton, K. and Mayers, D.: Numerical Solution of Partial Differential Equations, *Cambridge university press*, 1994.

[27] Pages, G. and Bally, V.: A quantization method for the discretization of BSDE's and Reflected BSDE's, *Technical report 628, université Paris 6*, 2000.

[28] Peng, S.: Monotonic limit theorem of BSDE and nonlinear decomposition theorem of Doob–Meyer's type, *Probab. Theory Related Fields*, 113 (1999), no. 4, 473–499.

[29] Peng, S. and Xu, M.: The smallest g-supermartingale and reflected BSDE with single and double L_2 obstacles, *Annales de l'Institut Henri Poincare (B) Probability and Statistics*, Volume 41, Issue 3, May-June 2005, Pages 605-630.

[30] Rogers, C.: Monte Carlo Valuation of American Options, *Mathematical Finance*, 32(4):1077–1088, 1995. Vol. 12, pp. 271-286, 2002.

[31] Tsitsiklis, J. and Van Roy, B.: Regression methods for Pricing complex American-style Options, *Working Paper MIT*, 1-22, 2000.

[32] Tsitsiklis, J. and Van Roy, B.:. Optimal Stopping of Markov Processes: Hilbert Spaces theory, Approximations Algorithms and an application to pricing high-dimensional financial derivatives, *IEEE Transactions on Automatic Control*, 44, 10, 1840-1851, 1999.