International Journal of Theoretical and Applied Finance
Vol. 14, No. 2 (2011) 197–219

DOI: 10.1142/S0219024911006334

PERTURBATION STABLE CONDITIONAL ANALYTIC MONTE-CARLO PRICING SCHEME FOR AUTO-CALLABLE PRODUCTS

CHRISTIAN P. FRIES

DZ BANK, D 60265 Frankfurt am Main, Germany
email@christian-fries.de

MARK S. JOSHI

Department of Economics
University of Melbourne
VIC 3010, Australia
mark@markjoshi.com

Received 26 April 2010
Accepted 21 October 2010

In this paper, we present a generic method for the Monte-Carlo pricing of (generalized) auto-callable products (aka. trigger products), i.e., products for which the payout function features a discontinuity with a (possibly) stochastic location (the trigger) and value (the payout).

The Monte-Carlo pricing of products with discontinuous payout is known to come with a high Monte-Carlo error. The numerical calculation of sensitivities (i.e., partial derivatives) of such prices by finite differences gives very noisy results, since the Monte-Carlo approximation (being a finite sum of discontinuous functions) is not smooth. Additionally, the Monte-Carlo error of the finite-difference approximation explodes as the shift size tends to zero.

Our method combines a product specific modification of the underlying numerical scheme, which is to some extent similar to an importance sampling and/or partial proxy simulation scheme *and* a reformulation of the payoff function into an equivalent smooth payout.

From the financial product we merely require that hitting of the stochastic trigger will result in an conditionally analytic value. Many complex derivatives can be written in this form. A class of products where this property is usually encountered are the so called auto-callables, where a trigger hit results in cancellation of all future payments except for one redemption payment, which can be valued analytically, conditionally on the trigger hit.

From the model we require that its numerical implementation allows for a calculation of the transition probability of survival (i.e., non-trigger hit). Many models allows this, e.g., Euler schemes of Itô processes, where the trigger is a model primitive.

The method presented is effective across a large range of cases where other methods fail, e.g. small finite difference shift sizes or short time to trigger reset (approaching maturity); this means that a practitioner can use this method and be confident that it will work consistently.

The method itself can be viewed as a generalization of the method proposed by Glasserman and Staum (2001), both with respect to the type (and shape) of the boundaries, as well as, with respect to the class of products considered. In addition we explicitly consider the calculation of sensitivities.

Keywords: Monte-Carlo simulation; pricing; greeks; variance reduction; auto-callable; trigger product; target redemption note.

1. Introduction

In this paper, we present a method which greatly improves the accuracy of the pricing and sensitivity calculation of (generalized) trigger products, such as for example some auto-callables. An auto-callable product is a financial derivative where some trigger index introduces a discontinuity in the payout such that

- An optional final payment (redemption) is made and
- All future payments are cancelled.

Our method can be applied to all products where a trigger index is observed and a hit of the trigger will result in an conditionally analytic value. This is the case for auto-callables if, conditionally, the redemption payment may be analytically evaluated.

Examples are digital options,[1] barrier options and target redemption notes. We allow for path-dependent triggers since they appear in target redemption notes.

Our method relies on the following modifications of the Monte-Carlo pricing algorithm:

(1) We modify the numerical scheme to generate only paths in the survival (non-trigger hit) domain.
(2) We modify the Monte-Carlo weights (probabilities) to attribute for the change of sampling domain.
(3) If necessary, we will modify the payout of the financial derivative such that
    (a) Pricing under the modified numerical scheme gives (converges to) the product price. This modification is only required if hitting the trigger results in an additional non-zero payout.
    (b) The payoff is continuous (smooth) at the trigger boundary. The necessity of this modification depends on properties of the numerical scheme, e.g., if approximations are involved.

It appears as if 3(b) is superfluous, because no path will hit the trigger. However, for the most general case we will work with an approximation which may introduce numerical errors such that some paths cross the trigger boundary. In this case, the

[1]The digital option is a degenerate example, because the trigger is observed only once and there is only one (future) payment to be canceled. The digital option is an example where the essence of the conditional-analytic simulation will become very transparent.

trigger criteria has to be ignored and it is 3(b) that makes the method work even better when using an approximation in (1).

Our work is somewhat related to the direct integration of the discontinuity of a discontinuous product as it was discussed for nth to default credit derivatives by Joshi & Kainth [13], and for CDOs by Fries & Rott [20].[2]

To some extent our method may be interpreted as a variant of a partial proxy simulation scheme, see [4], but in contrast to a partial proxy simulation scheme we modify the numerical scheme such that it samples only a subset of the whole domain of possible realizations of the original numerical scheme. In addition we add a correction term to the product payoff compensating for the part which was left out (if required). We require that, conditionally, there is an analytic formula (or approximation) for this region (hence we call our method "conditional-analytical"). As for the proxy simulation scheme: both parts of the payoff are multiplied with a weight compensating for the measure change applied to the numerical scheme. The method also bears some similarities to the approach used for pricing barrier options by Joshi and Leung [14]. Indeed, one could regard the method we present here as being a hybridization of these two approaches.

All this is done on a per time-step and per path basis within the numerical simulation; this means that the location of the trigger as well as the payout in case of a trigger hit can be stochastic, since analyticity of the payout is only required conditionally.

Although the interplay between product pricing code and the model's numerical scheme seems to be complex, both are tied by two simple pieces of information:

(1) The location of the trigger in the next simulation time-step (to be provided by the product) and
(2) The probability to trigger (to be provided by the model).

In most settings it is easy to calculate these on the level of the numerical scheme.

The simple use of analytic formulas within a Monte-Carlo simulation to improve convergence is very common. For example: for the valuation of a range accrual, one approach it is common to simulate only larger time steps and approximate the accrual periods using analytic formulas; for barrier options there are adjustments which allow larger time-steps for the Monte-Carlo simulation and adjust for the in-between barrier hit probability (see, e.g., [14]).

Also for barrier options, Glasserman and Staum [9] proposed an algorithm which performs an importance sampling and allows to add a possible redemption. Indeed, for the case of a linear boundary functional f (using the notation of Sec. 3.2) the simulation method proposed in Sec. 3.4 is similar to that proposed by Glasserman and Staum and briefly discussed in the context of TARNS in the LMM in [19].

[2]This is the motivation to denote the method as *conditional-analytic*.

Glasserman and Staum were, however, principally focussed on pricing rather than sensitivities. In particular, when the constraint is a non-linear function of the state variables, their methodology introduces a discontinuity which will increase the variance for deltas and have a large effect for gammas. Our contributions are therefore that

- We consider a broader class of products/models, introducing the glass of generalized trigger products, comprising target redemption notes in a multi-factor model;
- Our simulation method explicitly generalizes to the case of non-linear boundary functionals;
- We explicitly consider the stability of sensitivities.

In particular, we propose a linearization of the barrier distance function f and a modified valuation algorithm, which removes the discontinuity induced by the barrier to ensure lower noise in the finite differences used to calculate sensitivities.

Whilst the study of Greeks in Monte-Carlo simulations has been extensive, see for example [1, 2, 8, 5], very little of the extent work is applicable to discontinuous products in, e.g., low-factor LIBOR market models.[3] The reason being that likelihood ratio or Malliavin calculus techniques [7] require the density to be smooth which fails for natural discretizations of the LIBOR market model, and path-wise techniques require either a smooth pay-off or explicit evaluation of delta function terms. The two main papers that address this case are the precursors to this paper [6] and [4]. In particular, before those papers the calculation of Greeks for TARNS was regarded as a very hard problem [19].

2. Product and Model Definition

We now define a class of auto-callable products and an exemplary pricing model for which we will then develop the conditional-analytic numerical scheme. Both, the product and the model definition can easily be generalized.

2.1. *Generalized trigger product*

Given a tenor structure $T_1 < T_2 < \cdots T_{n+1}$ we consider a (generalized) trigger product paying

$$X(T_{j+1}) = \begin{cases} C_j & \text{if } I_j < H_j \text{ and } \forall\, k < j : I_k < H_k, \\ R_j & \text{if } I_j \geq H_j \text{ and } \forall\, k < j : I_k < H_k, \\ 0 & \text{else} \end{cases}$$

[3] A reduction of the number of factors of a LIBOR market model is desirable, because it allows a much faster generation of the Monte-Carlo paths, see [12].

in T_{j+1} for $j = 1, 2, \ldots, n$.[4] Here, I_j is the trigger index with fixing in T_j, i.e. it is an $\mathcal{F}_{T_j}$-measurable random variable and H_j is an $\mathcal{F}_{T_{j-1}}$-measurable random variable, the trigger level. We assume that payments C_j (coupon) and R_j (coupon plus redemption) are $\mathcal{F}_{T_{j+1}}$-measurable and paid in T_{j+1}.[5] There is actually no restriction on the fixing and payment of the coupon, but we have to impose an additional assumption on the redemption payment R_j, which we will formulate in Sec. 2.1.1 below.

If

$$A_j := \{I_j < H_j \text{ and } \forall\, k < j : I_k < H_k\}$$
$$B_j := \{I_j \geq H_j \text{ and } \forall\, k < j : I_k < H_k\}$$

denote the survival and the trigger hit regime, respectively, then the payout can be written as

$$X(T_{j+1}) = C_j \mathbf{1}_{A_j} + \ R_j \mathbf{1}_{B_j}.$$

2.1.1. *Conditional analyticity of the redemption payment*

We assume that conditional to $\mathcal{F}_{T_{i-1}}$ we have an analytic pricing formula (or approximation) for the next period's redemption payment, i.e., we analytically have

$$\tilde{R}_j(T_{j-1}) := N(T_{j-1})\mathrm{E}^{\mathbb{Q}}\left(\frac{R_j}{N(T_{j+1})}\mathbf{1}_{B_j}\,\middle|\,\mathcal{F}_{T_{j-1}}\right).$$

This allows us to equivalently reformulate the payoff in the following sense:

Lemma. *Define*

$$\tilde{X}(T_{j+1}) = \frac{\tilde{R}_j(T_{j-1})}{P(T_{i+1}; T_{i-1})} + \begin{cases} C_j & \text{if } I_j < H_j \text{ and } \forall\, k < j : I_k < H_k, \\ 0 & \text{otherwise,} \end{cases}$$

then at $T_k \leq T_{j-1}$, the valuation of the payoffs $\tilde{X}(T_{j+1})$ and $X(T_{j+1})$ coincide.

Proof. Let A_j and B_j as above. Then

$$X(T_{j+1}) = C_j \mathbf{1}_{A_j} + R_j \mathbf{1}_{B_j}. \qquad \square$$

Let $\mathbb{Q}$ denote the pricing measure corresponding to the numéraire N.

$$\mathrm{E}^{\mathbb{Q}}\left(\frac{X(T_{j+1})}{N(T_{j+1})}\,\middle|\,\mathcal{F}_{T_{j-1}}\right)$$
$$= \mathrm{E}^{\mathbb{Q}}\left(\frac{C_j}{N(T_{j+1})}\mathbf{1}_{A_j} + \frac{R_j}{N(T_{j+1})}\mathbf{1}_{B_j}\,\middle|\,\mathcal{F}_{T_{j-1}}\right)$$

[4] We consider payment in T_{j+1}. This is no restriction, because other payment times (e.g. for in-arrears fixing indices) can be reflected by multiplying or dividing the payout with the corresponding discount factor.

[5] In the regular case C_j and R_j are even $\mathcal{F}_{T_j}$-measurable.

$$= \mathrm{E}^{\mathbb{Q}}\left(\frac{C_j}{N(T_{j+1})}\mathbf{1}_{A_j}\middle|\mathcal{F}_{T_{j-1}}\right) + \mathrm{E}^{\mathbb{Q}}\left(\frac{R_j}{N(T_{j+1})}\mathbf{1}_{B_j}\middle|\mathcal{F}_{T_{j-1}}\right)$$

$$= \mathrm{E}^{\mathbb{Q}}\left(\frac{C_j}{N(T_{j+1})}\mathbf{1}_{A_j}\middle|\mathcal{F}_{T_{j-1}}\right) + \frac{\tilde{R}_j(T_{j-1})}{N(T_{j-1})} = \mathrm{E}^{\mathbb{Q}}\left(\frac{\tilde{X}(T_{j+1})}{N(T_{j+1})}\middle|\mathcal{F}_{T_{j-1}}\right).$$

2.1.2. *Example: Target redemption note*

For a target redemption note the trigger criteria is

$$\sum_{k=1}^{j} C_k \geq C^*,$$

where C^* is the target coupon. The redemption usually consists of a notional payment (assumed to be 1) and a coupon filling the gap for the target coupon. Within the notation above, the target redemption note has

$$I_j = C_j, \quad H_j = C^* - \sum_{k=1}^{j-1} C_k, \quad R_j = 1 + H_j.$$

For the case where the redemption is paid at T_{j+1} then $\tilde{R}_j(T_{j-1})$ is the value of a digital option with the underlying index I_j (fixing in T_j, payment in T_{j+1}).

2.2. *Pricing algorithm*

The equivalent[6] reformulation of the payout allows us to develop a new pricing algorithm. We generate a (Monte-Carlo) simulation restricted to the domain $\cup_i A_i$. This allows the numerical evaluation of the complex coupon part $C_i \mathbf{1}_{A_i}$, as usual. The conditional analytic part $R_i \mathbf{1}_{B_i}$ will be treated in every time step using the conditional analytic formula $\tilde{R}_i$. With this reformulation, the Monte-Carlo simulation will not suffer from the Monte-Carlo error induced by the discontinuity at the border of $\cup_i A_i$. If C_i is smooth, then the Monte-Carlo simulation will effectively be applied to a smooth product. The discontinuous part is handled analytically. The result is a sizeable reduction of Monte-Carlo variance for price and particularly sensitivities.

3. Modification for a Monte-Carlo Euler Scheme

3.1. *Pricing model*

We consider a pricing model given by a discretized stochastic process K. For illustrative purposes we consider a model given by an Itô stochastic process [3, 11, 16]

$$\mathrm{d}K = \mu(t)\mathrm{d}t + \Sigma(t)\cdot\Gamma(t)\cdot\mathrm{d}W(t),$$

[6]The term "equivalent" here refers to "equivalent with respect to valuation using the chosen numeraire-measure pair $(N, \mathbb{Q})$".

where $W = (W_1, \ldots, W_m)$ and W_i are Brownian motions with

$$\mathrm{d}W_i \mathrm{d}W_j = \begin{cases} \mathrm{d}t & \text{for } i = j, \\ 0 & \text{otherwise,} \end{cases}$$

and Σ and Γ denote the volatility and the factor matrix, respectively, determining the instantaneous covariance of the model. We focus on this model for illustrative purposes mainly. We allow the possibility that the coefficients are state-dependent.[7]

Our methodology may be generalized to others models. In fact, the method presented makes only a few basic assumptions on the numerical discretization scheme of the model. E.g., the following derivation does not assume that $\Delta W(t_i) := \int_{t_i}^{t_{i+1}} \mathrm{d}W(t)$ is a Brownian increment. We just require that there is an sufficiently efficient and accurate algorithm to calculate the conditional cumulative distribution function of $\Delta K(t_i) = K(t_{i+1}) - K(t_i)$ and its inverse. If μ, Σ and Γ are previsible then this is equivalent that an there is an sufficiently efficient and accurate algorithm to calculate the conditional cumulative distribution function of $\Delta W(t_i)$.

3.2. *Assumption on the trigger*

We assume that the *trigger index* I_j of the trigger product is a function of the model's state variables $K(T_j)$, i.e.,

$$I_j = f(T_j, K(T_j)).$$

In other words, we assume that the trigger index I_j itself is not path-dependent in terms of the model primitives. However, since we allow that the trigger level H_j is an $\mathcal{F}_{T_{j-1}}$-measurable random variable, most products with path-dependent triggers can be rewritten in the above form, e.g., as for the target redemption note in the previous example.

3.3. *Original numerical scheme*

Let $\tilde{K}(t_i)$ be an approximation of $K(t_i)$ given by a numerical scheme, e.g., an Euler-like discretization of our model given by

$$\Delta\tilde{K}(t_i) = \tilde{\mu}(t_i)\Delta t_i + \Sigma(t_i) \cdot \Gamma(t_i) \cdot \Delta W(t_i), \quad \tilde{K}(0) = K(0).$$

Let $\Delta W_k(t_i)$ be generated by drawings from independent uniform random variables $Z_{i,k}$ using

$$\Delta W_k(t_i) = \Phi^{-1}(Z_{i,k})\sqrt{\Delta t_i},$$

where Φ^{-1} denotes the inverse of the cumulative standard normal distribution function.[8] The fact that post-discretization the model increments are Gaussian means that the method still applies despite the state-dependence.

[7]The LIBOR market model and swap market models are of this type. For the LIBOR market model we have $K_i = \log(L_i)$, where L_i is the forward rate for the period $[T_i, T_{i+1}]$.
[8]For an overview on Monte-Carlo methods see, e.g., [10].

Most models in derivative pricing are based on Itô processes and an implementation using an Euler scheme is often sufficient. However, it is straight-forward to extend our approach to more general processes and other discretization schemes, as long as there are formulas for the distribution functions of the transition probability of $\Delta K(t_i)$.

3.4. *Numerical scheme adapted to the trigger product*

We will define a Monte-Carlo pricing scheme that allows to price our trigger product without its pathwise discontinuity. The idea is to generate only those paths that do not hit the trigger, calculate the corresponding probability measure, and semi-analytically calculate the value given by a trigger hit. To do so, we define the gradient of the trigger criteria (i.e. $f - H$) and calculate the location of the trigger.

We will assume that the trigger function is linear with respect to the one-step Brownian increment. We will comment on the general case of non-linear triggers in Sec. 4.1.

Induction Start: Let $K^*(t_0) := K(t_0)$.

Induction Step: Given $K^*(t_i)$ let

$$g(x) = f(K^*(t_i) + \tilde{\mu}(t_i, K^*)\Delta t_i + x).$$

Define

$$v = \nabla g(0)/\|\nabla g(0)\|$$

and let $q \in \mathbb{R}$ be the solution[9] of

$$g(qv) = H_{i+1}.$$

We will assume now, that g is linear in x, the discussion of the general case will follow below. If g is linear in x we have

$$g(qv) = g(0) + \nabla g(0) \cdot qv, \quad \text{i.e., define } q := \frac{g(0) - H_{i+1}}{\|\nabla g(0)\|}. \tag{3.1}$$

Then

$$I_{i+1} < H_{i+1} \Leftrightarrow g(\Sigma\Gamma\Delta W) < H_{i+1} \Leftrightarrow \Sigma\Gamma\Delta W < qv \Leftrightarrow \langle v, \Sigma\Gamma\Delta W\rangle < q$$

Let

$$X := \langle v, \Gamma\Delta W\rangle.$$

We wish to replace the sampling of X with a sampling Y such that $Y < q$. Clearly, X is a normal distributed random variable with mean 0. Let σ_X denote the standard

[9]To be precise, we consider $q \in \mathbb{R}$ conditional $\mathcal{F}_{T_i}$, i.e., q is a $\mathcal{F}_{T_i}$-measurable random variable.

deviation of X. Then $x = \Phi(X/\sigma_X)$ is uniform distributed. Let $b := \Phi(q)$ and $Y := \Phi^{-1}(bx)$. Then we have that $bx < b$, thus $Y < q$. Furthermore,

$$P(X < K) = bP(Y < K)$$

for all $K < q$. i.e., the distribution function of Y and X differ on $(-\infty, q)$ only by the constant factor b.

In other words: sampling Y is equivalent to sampling X on the restricted domain $(-\infty, q)$, with a Monte-Carlo weight b. For $\Gamma\Delta W + (Y - X)v$ we have

$$\langle v, \Gamma\Delta W + (Y - X)v\rangle = X + Y - X = Y \leq q$$

In place of $\tilde{K}$ we consider the numerical scheme K^* defined by

$$K^*(t_{i+1}) := K^*(t_i) + \tilde{\mu}(t_i)\Delta t_i + \Sigma \cdot \Gamma(\Delta W + (Y - X)v)$$

This scheme has the property that

$$\begin{aligned} f(K^*(t_{i+1})) &= f(K^*(t_i) + \tilde{\mu}(t_i)\Delta t_i) + \nabla g \cdot (\Delta W + (\Phi^{-1}(qZ) - \Phi^{-1}(Z))v_1) \\ &= g(0) + \nabla g \cdot qv \leq H_{i+1} \end{aligned} \tag{3.2}$$

Thus, for linear triggers we have that this scheme generates realizations that sample the non-trigger hit region. For the original increment we had

$$\mathbb{Q}(f(K(t_{i+1})) \leq H_{i+1} \,|\, K^*(t_i)) = b,$$

for the adapted scheme we have

$$\mathbb{Q}(f(K^*(t_{i+1})) \leq H_{i+1} \,|\, K^*(t_i)) = 1,$$

i.e., the Monte-Carlo weight of the corresponding sample path will be multiplied with a factor of b.

Note that this is applied conditionally to t_i in each time step.[10]

3.5. *Reformulation of the pricing*

If our numerical scheme samples only the survival region, then we may rewrite the product such that it may be evaluated purely on the paths of K^*. On each path ω_k we calculate the value

$$X(T_{j+1}, \omega_k) = C_j(\omega_k) \cdot Q_j(\omega_k) + \frac{\tilde{R}_j(T_{j-1})}{P(T_{i+1}; T_{i-1})}, \tag{3.3}$$

where $Q_j(\omega_k)$ is the likelihood ratio given by the importance sampling K^* versus K. The probability Q_j may be calculated directly from the conditional probabilities of not hitting the barrier, provided by the model K^*:

$$Q_j = \prod_{i: T_j \leq t_i \leq T_{j+1}} b_i.$$

[10] Effectively x, y, b are processes.

In other words, we have introduced an importance sampling which additionally allows to rewrite the previously discontinuous payout as a smooth one. Rewriting the product in this way is also important if numerical errors or the approximations in (4.1) and (4.2) result in K^* hitting the trigger. In the payout (3.3) the discontinuity of the trigger has been removed.

4. Generalizations

4.1. *Non-linear triggers*

In the previous section we derived our numerical scheme under the assumption that the function g (i.e., the trigger) is linear in the Brownian increment. This assumption seems to be strong, but it is actually fairly weak: in many cases, a change of the model primitives and/or and equivalent transformation of the trigger criteria can be applied so that the trigger criteria becomes a linear function of the Brownian increment.

Let us illustrate this for simple examples: consider a trigger criteria $L > H$ where L follows a lognormal process. We then transform the trigger criteria to $\log(L) > \log(H)$, and define an Euler scheme for $K := \log(L)$ as in Sec. 3.4. The trigger function $f(L) = \log(L) - \log(H)$ is linear in the diffusion of K. If the trigger is a CMS swap-rate this can be achieved by using a swap-rate market model instead of a LIBOR market model. See, for example, [15] or [17].

Effectively, this procedure represents a subtle linearization of the trigger, because the underlying state variable K is linearized within the time-step Δt through the numerical scheme.

If this is not possible, we may linearize g. If g is smooth, the linearization error will tend to 0 as $\Delta t \to 0$. We will then work with a linearization of (3.1):

$$g(qv_1) \approx g(0) + \nabla g \cdot qv_1, \tag{4.1}$$

and Eq. (3.2) will hold only as an approximation, i.e., the scheme then has the property that

$$\begin{aligned} f(K^*(t_{i+1})) &\approx f(K^*(t_i) + \tilde{\mu}(t_i)\Delta t_i) + \nabla g \cdot \left(qv_1 z_1 + \sum_{j=2}^{n} v_j z_j \right) \\ &= g(0) + \nabla g \cdot (qv_1) z_1 \leq H_{i+1} \end{aligned} \tag{4.2}$$

So in first order we have that the scheme generates realizations that do not hit the trigger. In the limit we have obviously

$$P(f(K^*(t_{i+1})) \leq H_{i+1}) \to 1 \quad \text{as } \Delta t \to 0.$$

In addition we have

$$\mathbb{Q}(f(K(t_{i+1})) \leq H_{i+1} \,|\, K^*(t_i)) \approx q.$$

Due to the time discretization error it is not guaranteed that the scheme does not generate paths for which the trigger is hit. However, in the limit $\Delta t \to 0$ this is the case. We can cope with this by modifying the payout in such a way that the product priced under the scheme is no longer a trigger.

4.2. *Other transition probabilites*

The conditional analytic numerical scheme may be generalized to other transition probabilites. The idea is the same as in [20]: Our scheme relied on a modification of the diffusion term $\Gamma\Delta W$. The modification required the knowledge of the cumulative distribution function Φ and its inverse Φ^{-1} only. In our example Φ was the distribution function of the standard normal. Obviously the method can be applied more general increment as long as we have a cumulative distribution function from which we can transform the increment and the trigger criteria to equidistributed random variables on $[0, 1]$. Since many numerical scheme generate their increment by means of equidistributed random variables transformed by the inverse cumulative distribution function, it is usually generally the case that code for Φ and Φ^{-1} exists, or can be constructed.

5. Numerical Results

The use of the conditional-analytic Monte-Carlo simulation reduces the Monte-Carlo error for pricing and for sensitivities calculated from from finite difference of pricings.

For the pricing, the size of the variance reduction depends on how large the product's discontinuity contributes to the Monte-Carlo error. We consider a simple digital option, the variance reduction will be more significant for options having short maturities. Otherwise the Monte-Carlo error contributed by the continuous part, the coupon and discount factor, will be much larger than the Monte-Carlo error contributed by the discontinuous part.

For the calculation of sensitivities from finite differences of prices the use of small shift sizes will magnify the Monte-Carlo error contributed from the discontinuity of the product. It is here where the power of method will show. This is similar to results in [14] where the fact that discontinuities do not arise from barrier crossing is crucial when computing Greeks.

5.1. *Comparison with other methods*

We compare the pricing of a direct simulation and the conditional analytic simulation and delta, gamma and vega between direct simulation (i.e. standard Monte-Carlo simulation), the partial proxy simulation scheme and the conditional analytic simulation.

In all cases, the conditional analytic simulation gives the best result. Similar to the localized partial proxy simulation scheme, see [6], the conditional analytic simulation does not suffer if the shift size used in the finite difference becomes large; this was not the case for partial proxy simulations.

5.1.1. *Role of finite difference shift size*

Using finite difference of prices to calculate sensitivities, the choice of the shift size is crucial. While direct simulation gives extremely noisy results for small shifts, the partial proxy method gives extremely noisy results for large shifts. Although one would in general prefer small shifts in order to reduce the error from higher order effect, and thus prefer the partial proxy method, whether a shift is "small" or "large" depends on the product considered: products for which the reset date of the trigger index is close are much more sensitive to the shift size than others. Thus, as a product is approaching its trigger reset date during its lifecycle, the partial proxy simulation scheme will give extremely noise results (given that the shift size is not adapted).

The calculation of sensitivities from the conditional analytic simulation does not exhibit this defect. A striking example is given in Sec. 5.7.

5.2. *Benchmark model*

As a benchmark model we take the LIBOR market model of semi-annual rates, with simplified model parameters: the initial forward curve is flat at 0.1, all rates have flat volatility 0.2 the Brownian driver had an exponentially decaying correlation $\rho_{i,j} = \exp(-0.15|T_i - T_j|)$, reduced to the first 5 factors.

We do a standard Euler scheme for $\log(L_i)$.

5.3. *Benchmark products*

As benchmark products we consider digital caplets and target redemption notes. The trigger index I_i of the digital caplet is the forward LIBOR L_i. The trigger index I_i of the target redemption note is the coupon C_i. We consider a target redemption note with a floored reverse-floating index $C_i = \max(0.10 - 2L_i, 0)$.

In both cases it is straight-forward to transform the trigger criteria to be linear in $\log(L_i)$. For the tarn, e.g., the trigger criteria is

$$\max(0.10 - 2L_i, 0) < H_i.$$

Since in our log-normal model $L_i \geq 0$, the trigger criteria for the tarn is only effective for trigger levels $H_i \leq 0.10$. In this case we have.

$$\begin{aligned}\max(0.10 - 2L_i, 0) < H_i \Leftrightarrow 0.10 - 2L_i < H_i \Leftrightarrow L_i < (0.10 - H_i)/2 \\ \Leftrightarrow \log(L_i) < \log((0.10 - H_i)/2)\end{aligned}$$

We will consider two TaRNs: the first one being a 6Y LIBOR TaRN with semi-annual fixings and target coupon $C^* = 0.10$, i.e., first fixing in 0.5, last fixing in 5.5; the second one being a 6.05Y LIBOR TaRN with short first period and target coupon $C^* = 0.0575$, i.e., first fixing in 0.05, then semi-annual fixings, last fixing in 5.55.

Table 1. Prices and standard deviation of a Monte-Carlo pricing using direct simulation and conditional analytic simulation, both with 5000 paths. The LIBOR TaRN Swap 2 has a short first period of length 0.05.

Product	Direct Simulation	Conditional Analytic
Digital Caplet/Maturity $t = 0.5$	$21.40\% \pm 0.31\%$	$21.40\% \pm 0.00\%$
Digital Caplet/Maturity $t = 2.0$	$17.38\% \pm 0.27\%$	$17.39\% \pm 0.19\%$
Digital Caplet/Maturity $t = 5.0$	$12.04\% \pm 0.19\%$	$12.03\% \pm 0.15\%$
LIBOR TaRN Swap 1/Maturity $t = 6.0$	$3.56\% \pm 0.07\%$	$3.56\% \pm 0.06\%$
LIBOR TaRN Swap 2/Maturity $t = 6.05$	$2.511\% \pm 0.012\%$	$2.511\% \pm 0.005\%$

5.4. *Pricing of digitals and TaRNs*

Compared to direct simulation, the conditional analytic simulation reduces the Monte-Carlo error. The reduction is small for products with long maturity, because here the Monte-Carlo error induced by the discontinuity is not the prominent part. For short maturities the reduction gets significant. The digital caplet with maturity $t = 0.5$ is a limit case, where the pricing under a conditional analytic simulation becomes completely analytic.

5.5. *Sensitivities of digital caplet*

In the following we present delta, gamma and vega calculated by finite differences applied to the respective pricing algorithm. In the figures we draw mean (line) and standard deviation (transparent corridor) for direct simulation (light gray), partial proxy simulation scheme (medium gray) and the conditional analytic scheme (dark gray).

The scaling of the sensitivities is as follows: delta and gamma are normalized as price change per 100 bp shift. Vega is normalized as price change per 1% volatility change times 100.

5.5.1. *Digital caplet: delta*

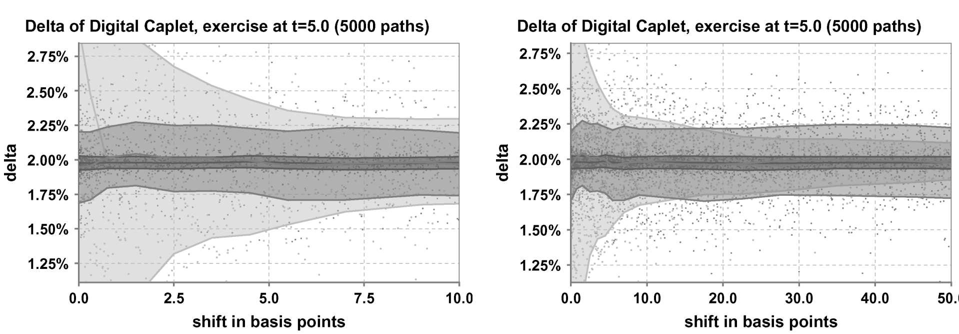


Fig. 1. Delta of a 5Y Digital Caplet: Finite difference is applied to a direct simulation (light gray), to a partial proxy simulation scheme keeping constraining cumulated coupons (medium gray) and to a conditional analytic scheme (dark gray).

Table 2. Delta of a 5Y-digital caplet. Data corresponding to Fig. 1.

Shift	Direct Simulation	Partial Proxy	Conditional Analytic
In bp	Mean ± Std. Dev. (%)	Mean ± Std. Dev. (%)	Mean ± Std. Dev. (%)
0.1–0.5	2.31 ± 2.67	1.96 ± 0.24	1.98 ± 0.04
0.5–1.0	1.91 ± 1.10	2.07 ± 0.21	1.98 ± 0.04
1.0–2.0	1.98 ± 0.78	2.02 ± 0.25	1.98 ± 0.05
2.0–3.0	2.02 ± 0.58	2.00 ± 0.23	1.98 ± 0.04
3.0–4.0	1.95 ± 0.52	$2.02 \pm 0{,}25$	1.98 ± 0.04
4.0–5.0	1.94 ± 0.46	1.96 ± 0.22	1.99 ± 0.05
5.0–6.0	1.95 ± 0.37	1.95 ± 0.28	1.97 ± 0.04
6.0–8.0	1.98 ± 0.31	1.99 ± 0.24	1.97 ± 0.04
8.0–10.0	1.99 ± 0.31	1.97 ± 0.23	1.98 ± 0.04
10–15	1.98 ± 0.25	1.97 ± 0.26	1.98 ± 0.04
15–20	1.97 ± 0.21	1.95 ± 0.26	1.97 ± 0.05
20–25	1.94 ± 0.20	1.99 ± 0.24	1.97 ± 0.05
25–30	1.98 ± 0.17	1.99 ± 0.25	1.98 ± 0.04
30–40	1.98 ± 0.16	2.00 ± 0.26	1.98 ± 0.04
40–50	1.99 ± 0.13	1.97 ± 0.25	1.97 ± 0.04

5.5.2. *Digital caplet: gamma*

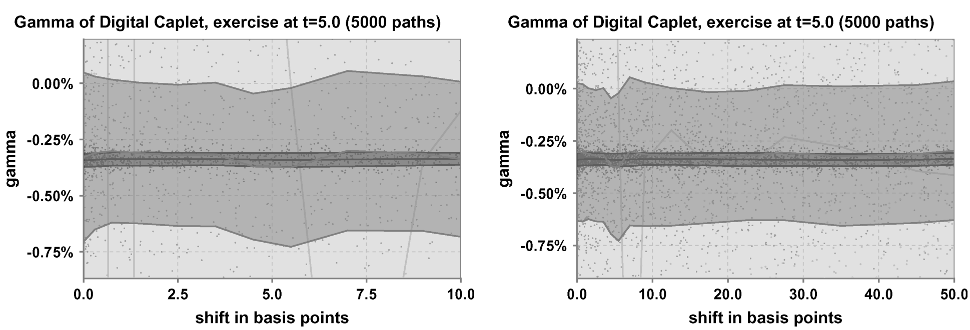


Fig. 2. Gamma of a 5Y Digital Caplet: Finite difference is applied to a direct simulation (light gray), to a partial proxy simulation scheme keeping constraining cumulated coupons (medium gray) and to a conditional analytic scheme (dark gray).

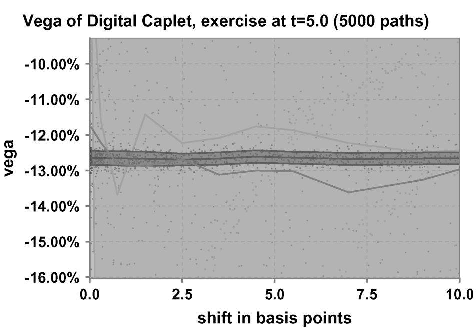
Table 3. Gamma of a 5Y-digital caplet. Data corresponding to Fig. 2.

Shift	Direct Simulation	Partial Proxy	Conditional Analytic
In bp	Mean ± Std. Dev. (%)	Mean ± Std. Dev. (%)	Mean ± Std. Dev. (%)
0.1–0.5	1.9E3 ± 1.3E4	-0.30 ± 0.33	-0.34 ± 0.03
0.5–1.0	35.14 ± 265.9	-0.31 ± 0.33	-0.34 ± 0.03
1.0–2.0	1.30 ± 112.2	-0.31 ± 0.30	-0.34 ± 0.03
2.0–3.0	5.43 ± 43.43	-0.33 ± 0.33	-0.34 ± 0.03
3.0–4.0	-0.49 ± 30.21	-0.30 ± 0.31	-0.34 ± 0.03
4.0–5.0	4.08 ± 18.84	-0.44 ± 0.34	-0.34 ± 0.03
5.0–6.0	-4.09 ± 14.11	-0.31 ± 0.37	-0.34 ± 0.03
6.0–8.0	-0.57 ± 9.23	-0.29 ± 0.34	-0.34 ± 0.03

Table 3. (*Continued*)

Shift	Direct Simulation	Partial Proxy	Conditional Analytic
In bp	Mean ± Std. Dev. (%)	Mean ± Std. Dev. (%)	Mean ± Std. Dev. (%)
8.0–10.0	-0.12 ± 5.99	-0.34 ± 0.35	-0.34 ± 0.03
10–15	-0.27 ± 4.26	-0.31 ± 0.31	-0.34 ± 0.03
15–20	-0.46 ± 2.60	-0.34 ± 0.31	-0.34 ± 0.03
20–25	-0.28 ± 1.68	-0.30 ± 0.31	-0.34 ± 0.03
25–30	-0.18 ± 1.26	-0.32 ± 0.34	-0.34 ± 0.03
30–40	-0.37 ± 0.91	-0.33 ± 0.33	-0.34 ± 0.03
40–50	-0.41 ± 0.65	-0.30 ± 0.33	-0.34 ± 0.03

5.5.3. *Digital caplet: vega*



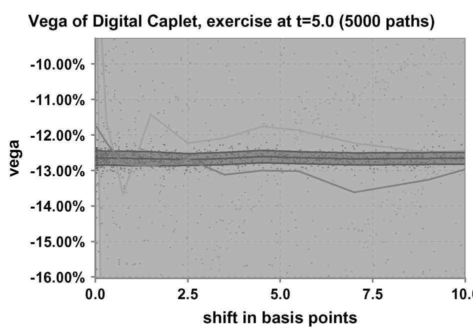


Fig. 3. Vega of a 5Y Digital Caplet: Finite difference is applied to a direct simulation (light gray), to a partial proxy simulation scheme keeping constraining cumulated coupons (medium gray) and to a conditional analytic scheme (dark gray).

Table 4. Vega of a 5Y-digital caplet. Data corresponding to Fig. 3.

Shift	Direct Simulation	Partial Proxy	Conditional Analytic
In bp	Mean ± Std. Dev. (%)	Mean ± Std. Dev. (%)	Mean ± Std. Dev. (%)
0.0–0.1	-6.70 ± 0.20	-11.77 ± 5.43	-12.65 ± 0.21
0.1–0.5	-16.49 ± 36.87	-12.66 ± 5.87	-12.66 ± 0.18
0.5–1.0	-10.86 ± 13.42	-12.99 ± 6.68	-12.66 ± 0.22
1.0–2.0	-11.99 ± 11.29	-12.77 ± 5.21	-12.68 ± 0.18
2.0–3.0	-12.47 ± 7.51	-12.60 ± 6.17	-12.71 ± 0.16
3.0–4.0	-11.71 ± 5.73	-13.64 ± 5.55	-12.62 ± 0.19
4.0–5.0	-11.82 ± 5.88	-12.37 ± 5.57	-12.60 ± 0.16
5.0–6.0	-11.92 ± 5.09	-13.68 ± 5.75	-12.68 ± 0.17
6.0–8.0	-12.54 ± 5.27	-13.56 ± 5.91	-12.67 ± 0.16
8.0–10.0	-12.44 ± 4.47	-12.97 ± 5.89	-12.66 ± 0.17

5.6. *Sensitivities of target redemption note*

5.6.1. *Target redemption note: delta*

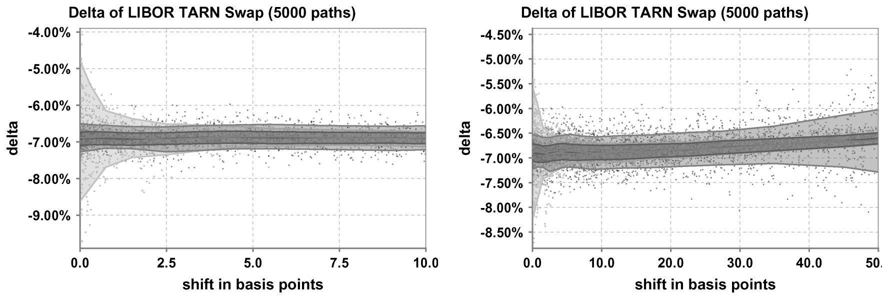


Fig. 4. Delta of a LIBOR TARN: Finite difference is applied to a direct simulation (light gray), to a partial proxy simulation scheme keeping constraining cumulated coupons (medium gray) and to a conditional analytic scheme (dark gray).

Table 5. Delta of a LIBOR TARN. Data corresponding to Fig. 4.

Shift	Direct Simulation	Partial Proxy	Conditional Analytic
In bp	Mean ± Std. Dev. (%)	Mean ± Std. Dev. (%)	Mean ± Std. Dev. (%)
0.1–0.5	-6.90 ± 1.30	-6.88 ± 0.34	-6.89 ± 0.18
0.5–1.0	-6.85 ± 0.60	-6.84 ± 0.29	-6.93 ± 0.18
1.0–2.0	-6.94 ± 0.46	-6.93 ± 0.31	-6.92 ± 0.16
2.0–3.0	-6.92 ± 0.38	-6.93 ± 0.37	-6.90 ± 0.16
3.0–4.0	-6.86 ± 0.32	-6.86 ± 0.30	-6.87 ± 0.17
4.0–5.0	-6.87 ± 0.26	-6.87 ± 0.35	-6.87 ± 0.16
5.0–6.0	-6.89 ± 0.23	-6.84 ± 0.32	-6.90 ± 0.17
6.0–8.0	-6.87 ± 0.20	-6.92 ± 0.33	-6.88 ± 0.14
8.0–10.0	-6.93 ± 0.20	-6.89 ± 0.33	-6.90 ± 0.15
10–15	-6.87 ± 0.17	-6.87 ± 0.33	-6.87 ± 0.14
15–20	-6.84 ± 0.15	-6.86 ± 0.33	-6.84 ± 0.13
20–25	-6.82 ± 0.15	-6.79 ± 0.35	-6.82 ± 0.13
25–30	-6.75 ± 0.14	-6.81 ± 0.34	-6.75 ± 0.13
30–40	-6.72 ± 0.12	-6.66 ± 0.42	-6.71 ± 0.11
40–50	-6.61 ± 0.12	-6.66 ± 0.63	-6.60 ± 0.11

5.6.2. *Target redemption note: gamma*

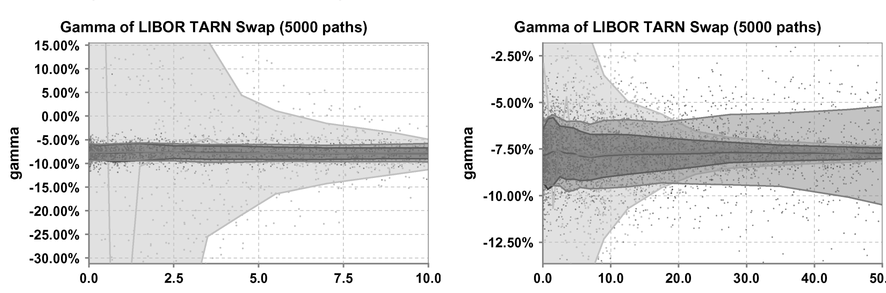


Fig. 5. Gamma of a LIBOR TARN: Finite difference is applied to a direct simulation (light gray), to a partial proxy simulation scheme keeping constraining cumulated coupons (medium gray) and to a conditional analytic scheme (dark gray).

Table 6. Gamma of a LIBOR TARN. Data corresponding to Fig. 5.

Shift	Direct Simulation	Partial Proxy	Conditional Analytic
In bp	Mean ± Std. Dev. (%)	Mean ± Std. Dev. (%)	Mean ± Std. Dev. (%)
0.1–0.5	-2.92 ± 4969.2	-7.71 ± 1.63	-7.84 ± 1.46
0.5–1.0	-7.21 ± 149.4	-7.51 ± 1.98	-7.69 ± 2.32
1.0–2.0	-14.13 ± 70.14	-7.62 ± 1.81	-7.58 ± 1.32
2.0–3.0	-0.22 ± 26.60	-7.69 ± 1.81	-7.68 ± 1.40
3.0–4.0	-9.20 ± 14.89	-8.10 ± 1.93	-7.77 ± 1.41
4.0–5.0	-7.29 ± 10.41	-7.63 ± 1.79	-7.71 ± 1.33
5.0–6.0	-8.06 ± 7.17	-7.96 ± 1.76	-8.03 ± 1.32
6.0–8.0	-7.75 ± 5.57	-7.84 ± 1.70	-7.85 ± 1.14
8.0–10.0	-8.11 ± 3.20	-7.73 ± 1.94	-7.87 ± 1.17
10–15	-7.53 ± 2.56	-7.72 ± 1.66	-7.72 ± 0.96
15–20	-7.67 ± 1.39	-7.65 ± 1.60	-7.84 ± 0.81
20–25	-7.73 ± 0.92	-7.59 ± 1.91	-7.65 ± 0.62
25–30	-7.69 ± 0.73	-7.47 ± 1.86	-7.71 ± 0.48
30–40	-7.75 ± 0.55	-7.61 ± 2.06	-7.73 ± 0.37
40–50	-7.74 ± 0.37	-7.85 ± 2.64	-7.73 ± 0.30

5.6.3. *Target redemption note: vega*

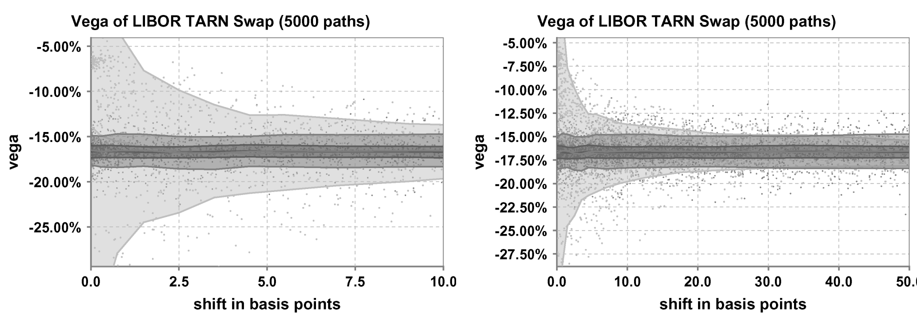


Fig. 6. Vega of a LIBOR TARN: Finite difference is applied to a direct simulation (light gray), to a partial proxy simulation scheme keeping constraining cumulated coupons (medium gray) and to a conditional analytic scheme (dark gray).

Table 7. Vega of a LIBOR TARN. Data corresponding to Fig. 6.

Shift	Direct Simulation	Partial Proxy	Conditional Analytic
In bp	Mean ± Std. Dev. (%)	Mean ± Std. Dev. (%)	Mean ± Std. Dev. (%)
0.1–0.5	-15.32 ± 17.95	-16.71 ± 1.74	-16.64 ± 0.67
0.5–1.0	-15.21 ± 9.19	-16.34 ± 1.92	-16.70 ± 0.64
1.0–2.0	-16.97 ± 7.61	-16.69 ± 1.62	-16.79 ± 0.59
2.0–3.0	-16.32 ± 5.96	-16.89 ± 1.95	-16.70 ± 0.62
3.0–4.0	-16.90 ± 4.33	-16.78 ± 1.65	-16.62 ± 0.69
4.0–5.0	-16.98 ± 4.32	-16.49 ± 1.68	-16.53 ± 0.64
5.0–6.0	-16.55 ± 4.01	-16.65 ± 1.87	-16.72 ± 0.68
6.0–8.0	-16.88 ± 3.40	-16.60 ± 1.76	-16.64 ± 0.56

Table 7. (*Continued*).

Shift In bp	Direct Simulation Mean ± Std. Dev. (%)	Partial Proxy Mean ± Std. Dev. (%)	Conditional Analytic Mean ± Std. Dev. (%)
8.0–10.0	-16.67 ± 2.99	-16.52 ± 1.80	-16.72 ± 0.63
10–15	-16.60 ± 2.49	-16.51 ± 1.73	-16.66 ± 0.61
15–20	-16.46 ± 2.06	-16.66 ± 1.56	-16.66 ± 0.59
20–25	-16.82 ± 1.85	-16.46 ± 1.70	-16.70 ± 0.62
25–30	-16.43 ± 1.78	-16.75 ± 1.67	-16.58 ± 0.69
30–40	-16.59 ± 1.62	-16.57 ± 1.78	-16.63 ± 0.63
40–50	-16.77 ± 1.36	-16.55 ± 1.83	-16.65 ± 0.61

5.7. *Sensitivities of target redemption note close to trigger reset*

The following example presents the delta, gamma and vega of a target redemption note with a short period of 0.05 to its next reset. The target coupon is 0.0575, such that under the market date assumed there is approximately a 50 : 50 chance of knock out in the next period.

In other words, we are approaching the discontinuity in time and space. Such a situation may indeed happen during the life-cycle of a target redemption note. In this case sensitivities will blow up.

5.7.1. *Target redemption note close to trigger reset: delta*

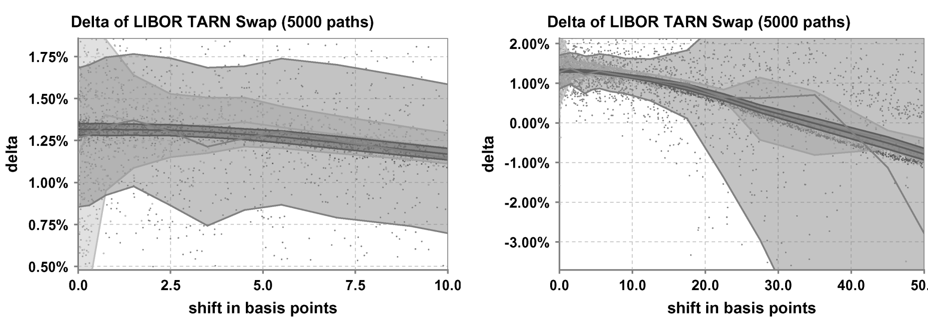


Fig. 7. Delta of a LIBOR TARN with short period to next reset. Finite difference is applied to a direct simulation (light gray), to a partial proxy simulation scheme keeping constraining cumulated coupons (medium gray) and to a conditional analytic scheme (dark gray).

Table 8. Delta of a LIBOR TARN with short period to next reset. Data corresponding to Fig. 7.

Shift In bp	Direct Simulation Mean ± Std. Dev. (%)	Partial Proxy Mean ± Std. Dev. (%)	Conditional Analytic Mean ± Std. Dev. (%)
0.1–0.5	1.36 ± 0.87	1.29 ± 0.42	1.32 ± 0.03
0.5–1.0	1.36 ± 0.34	1.37 ± 0.40	1.32 ± 0.04
2.0–3.0	1.31 ± 0.16	1.24 ± 0.49	1.31 ± 0.03

Table 8. (*Continued*)

Shift	Direct Simulation	Partial Proxy	Conditional Analytic
In bp	Mean ± Std. Dev. (%)	Mean ± Std. Dev. (%)	Mean ± Std. Dev. (%)
3.0–4.0	1.37 ± 0.17	1.19 ± 0.45	1.29 ± 0.03
4.0–5.0	1.35 ± 0.12	1.34 ± 0.40	1.28 ± 0.03
5.0–6.0	1.31 ± 0.12	1.27 ± 0.47	1.26 ± 0.04
6.0–8.0	1.26 ± 0.10	1.22 ± 0.44	1.22 ± 0.04
8.0–10.0	1.20 ± 0.09	1.14 ± 0.44	1.17 ± 0.03
10–15	1.07 ± 0.09	1.03 ± 0.61	1.02 ± 0.07
15–20	0.85 ± 0.11	0.91 ± 1.11	0.78 ± 0.08
20–25	0.57 ± 0.14	0.37 ± 2.92	0.51 ± 0.09
25–30	0.14 ± 1.43	0.87 ± 4.22	0.20 ± 0.10
30–40	-0.16 ± 0.18	0.53 ± 8.74	-0.23 ± 0.16
40–50	-0.69 ± 0.29	-2.78 ± 31.78	-0.79 ± 0.15

5.7.2. *Target redemption note close to trigger reset: gamma*

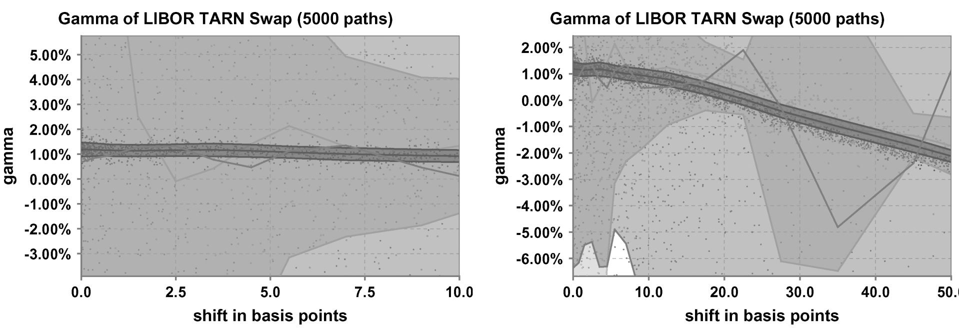


Fig. 8. Gamma of a LIBOR TARN with short period to next reset. Finite difference is applied to a direct simulation (light gray), to a partial proxy simulation scheme keeping constraining cumulated coupons (medium gray) and to a conditional analytic scheme (dark gray).

Table 9. Gamma of a LIBOR TARN with short period to next reset. Data corresponding to Fig. 8.

Shift	Direct Simulation	Partial Proxy	Conditional Analytic
In bp	Mean ± Std. Dev. (%)	Mean ± Std. Dev. (%)	Mean ± Std. Dev. (%)
0.1–0.5	226.5 ± 2799.6	0.89 ± 7.23	1.18 ± 0.26
0.5–1.0	5.79 ± 97.46	0.89 ± 6.88	1.14 ± 0.23
1.0–2.0	-0.76 ± 26.02	1.68 ± 6.68	1.14 ± 0.23
2.0–3.0	0.58 ± 14.87	0.97 ± 6.71	1.19 ± 0.24
3.0–4.0	0.28 ± 8.82	0.59 ± 7.48	1.16 ± 0.28
4.0–5.0	2.55 ± 6.57	0.35 ± 6.08	1.11 ± 0.22
5.0–6.0	1.72 ± 4.00	1.90 ± 5.98	1.05 ± 0.24
6.0–8.0	0.89 ± 3.24	0.80 ± 7.61	0.96 ± 0.25
8.0–10.0	1.33 ± 2.71	0.12 ± 8.28	0.92 ± 0.24
10–15	1.14 ± 1.64	0.87 ± 8.30	0.66 ± 0.26

Table 9. (*Continued*)

Shift	Direct Simulation	Partial Proxy	Conditional Analytic
In bp	Mean ± Std. Dev. (%)	Mean ± Std. Dev. (%)	Mean ± Std. Dev. (%)
15–20	0.67 ± 0.95	0.64 ± 11.21	0.25 ± 0.22
20–25	0.32 ± 1.13	3.16 ± 24.11	-0.17 ± 0.24
25–30	-1.20 ± 10.21	-3.72 ± 32.95	-0.66 ± 0.26
30–40	-0.95 ± 0.60	-5.90 ± 47.20	-1.33 ± 0.28
40–50	-1.73 ± 1.08	1.12 ± 179.2	-2.12 ± 0.23

5.7.3. *Target redemption note close to trigger reset: vega*

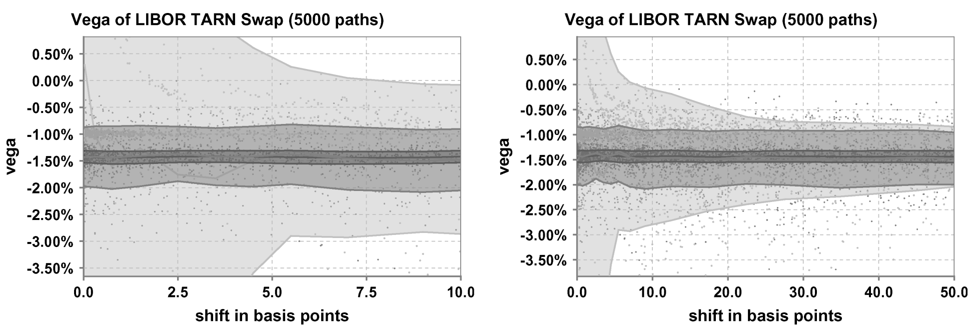


Fig. 9. Vega of a LIBOR TARN with short period to next reset. Finite difference is applied to a direct simulation (light gray), to a partial proxy simulation scheme keeping constraining cumulated coupons (medium gray) and to a conditional analytic scheme (dark gray).

Table 10. Vega of a LIBOR TARN with short period to next reset. Data corresponding to Fig 9.

Shift	Direct Simulation	Partial Proxy	Conditional Analytic
In bp	Mean ± Std. Dev. (%)	Mean ± Std. Dev. (%)	Mean ± Std. Dev. (%)
0.1–0.5	-1.53 ± 11.53	-1.42 ± 0.58	-1.43 ± 0.11
0.5–1.0	-0.87 ± 1.28	-1.46 ± 0.59	-1.44 ± 0.12
1.0–2.0	-1.66 ± 4.68	-1.36 ± 0.55	-1.43 ± 0.10
2.0–3.0	-1.85 ± 3.19	-1.38 ± 0.47	-1.41 ± 0.11
3.0–4.0	-1.80 ± 2.65	-1.46 ± 0.59	-1.42 ± 0.10
4.0–5.0	-1.19 ± 1.55	-1.37 ± 0.53	-1.43 ± 0.12
5.0–6.0	-1.46 ± 1.61	-1.38 ± 0.59	-1.43 ± 0.12
6.0–8.0	-1.42 ± 1.37	-1.52 ± 0.59	-1.45 ± 0.11
8.0–10.0	-1.47 ± 1.39	-1.48 ± 0.57	-1.42 ± 0.11
10–15	-1.43 ± 1.14	-1.45 ± 0.56	-1.43 ± 0.11
15–20	-1.53 ± 0.95	-1.53 ± 0.56	-1.46 ± 0.10
20–25	-1.51 ± 0.80	-1.37 ± 0.52	-1.42 ± 0.11
25–30	-1.52 ± 0.75	-1.56 ± 0.57	-1.44 ± 0.12
30–40	-1.48 ± 0.72	-1.43 ± 0.57	-1.44 ± 0.11
40–50	-1.44 ± 0.60	-1.48 ± 0.52	-1.44 ± 0.12

6. Conclusions

We have presented a reformulation of the pricing of a family of generalized auto-callable products. For an Euler scheme of an Itô process we showed how to construct a Monte-Carlo scheme sampling only the survival domain of the auto-callable product. The pricing and, especially, the calculation of risk parameters using finite differences of the reformulated payout under this scheme featured a greatly reduced Monte-Carlo variance.

The method presented may easily be generalized to numerical schemes of other processes. It basic requirements are

- The auto-callable value upon trigger hit may be valued analytically.
- The trigger criteria may be formulated such that the trigger index is linear in the increment of the numerical scheme. If not, a linearization may still work in the sense of an approximation, converging for small time steps.
- The cumulative distribution function of the increment of the numerical scheme as well as its inverse is known.[11]

We have seen that this method is effective across a large range of cases where other methods fail; this means that a practitioner can use this method and be confident that it will work consistently.

Appendix: List of Symbols

Symbol	Meaning
t_i	Simulation time discretization, common to all numerical schemes ($i = 0, 1, 2, \ldots$).
T_i	Tenor time discretization ($i = 0, 1, 2, \ldots$). Note: In this paper we assume $t_i = T_i$ for simplicity.
$X(T_{i+1})$	Payment of auto-callable for the period $[T_i, T_{i+1}]$, considered to be payed at T_{i+1}.
C_i	Coupon payment upon survival for the period $[T_i, T_{i+1}]$, considered to be payed at T_{i+1}.
R_i	Redemption payment upon trigger hit (termination) for the period $[T_i, T_{i+1}]$, considered to be payed at T_{i+1}.
I_i	Trigger index for the period $[T_i, T_{i+1}]$.
H_i	Trigger level for the period $[T_i, T_{i+1}]$.
A_i	Domain of survival on $(T_0, T_{i+1}]$.
B_i	Domain of trigger hit in $(T_i, T_{i+1}]$, given survival on $(T_0, T_i]$.

[11] The term analytically of "is known" is used in the sense that there exists a sufficiently accurate and sufficiently fast method of calculating the specific quantity. For example, we do not have an analytic expression for the cumulative normal distribution Φ and its inverse Φ^{-1}, but it may be rapidly calculated up to machine precision.

$\tilde{R}_i(T_{i-1})$	Value of the redemption payment, seen in T_{i-1}, given survival on $(T_0, T_i]$.
$P(T_{i+1}; T_{i-1})$	Value of zero coupon bond with maturity in T_{i+1}, seen in T_{i-1}.
C^*	Target coupon of target redemption note.
K	Model sde, here (exemplary) an Itô process.
$\tilde{K}$	(Unmodified) numerical scheme (Euler) for the exemplary model sde K.
K^*	Conditional analytic scheme, adapted to the trigger product.
Φ	Cumulated distribution function (of the original scheme's transition density).

References

[1] P. Boyle, M. Boadie and P. Glasserman, Monte-Carlo methods for security pricing, *Journal of Economic Dynamics and Control* **21** (1997) 1267–1321.

[2] M. Boadie and P. Glasserman, Estimating security price derivatives using simulation, *Management Science* **42**(2) (1996) 269–285.

[3] C. P. Fries, *Mathematical Finance. Theory, Modeling, Implementation* (John Wiley & Sons, 2007), http://www.christian-fries.de/finmath/book.

[4] C. P. Fries and M. S. Joshi, Partial proxy simulation schemes for generic and robust Monte-Carlo greeks, *Journal of Computational Finance* (2008) 12–1. http://www.christian-fries.de/finmath/proxyscheme.

[5] C. P. Fries and J. Kampen, Proxy simulation schemes for generic robust Monte-Carlo sensitivities, process oriented importance sampling and high accuracy drift approximation, *Journal of Computational Finance* (2006) 10–2, http://www.christian-fries.de/finmath/proxyscheme.

[6] C. P. Fries, Localized proxy simulation schemes for generic and robust Monte-Carlo greeks (2007), http://www.christian-fries.de/finmath/proxyscheme.

[7] E. Fournié, J.-M. Lasry, J. Lebuchoux, P. Lions and N. Touzi, Applications of Malliavin calculus to Monte-Carlo methods in finance, *Finance Stochastics* **3** (1999) 391–412 (Springer-Verlag, 1999).

[8] P. Glasserman, *Monte-Carlo Methods in Financial Engineering* (Stochastic Modelling and Applied Probability), Springer, 2003.

[9] P. Glasserman and J. Staum, Conditioning on one-step survival in barrier option simulations, *Operations Research* **49** (2001) 923–937.

[10] P. Jaeckel, *Monte Carlo Methods in Finance* (Wiley, 2002).

[11] M. S. Joshi, *The Concepts and Practice of Mathematical Finance* (Cambridge University Press, 2003).

[12] M. S. Joshi, *Rapid Computation of Drifts in a Reduced Factor LIBOR Market Model* (Wilmott Magazine, May, 2003).

[13] M. S. Joshi and D. S. Kainth, Rapid computation of prices and deltas of nth to default swaps in the Li Model, *Quantitative Finance* **4**(3) (2004) 266–275.

[14] M. S. Joshi and T. Leung, Using Monte-Carlo simulation and importance sampling to rapidly obtain jump-diffusion prices of continuous barrier options, *Journal of Computational Finance* (2007).

[15] M. S. Joshi and L. Liesch, Effective implementation of generic market models, *ASTIN Bulletin* (2007), pp. 453–473.

[16] P. E. Kloeden and E. Platen, Numerical solution of stochastic differential equations (Applications of Mathematics, Stochastic Modelling and Applied Probability, Vol. 23), Springer-Verlag (1999).

[17] E. Pietersz and M. van Regenmortel, Generic market models, *Finance and Stochastics* **10** (2006) 507–528.

[18] V. V. Piterbarg, Computing deltas of callable LIBOR exotics in forward LIBOR models, *Journal of Computational Finance* **7** (2003).

[19] V. V. Piterbarg, *TARNs: Models, Valuation, Risk Sensitivities* (Wilmott Magazine, 2004).

[20] M. G. Rott and C. P. Fries, Fast and Robust Monte-Carlo CDO Sensitivities and their Efficient Object Oriented Implementation (2005). http://www.christian-fries.de/finmath/cdogreeks.