Partial Proxy Simulation Schemes for Generic and Robust Monte-Carlo Greeks

Christian P. Fries
email@christian-fries.de

Mark S. Joshi
mark@markjoshi.com

September 30, 2006

(Version 1.1.3)

Abstract

We consider a generic framework which allows to calculate robust Monte-Carlo sensitivities seamlessly through simple finite difference approximation. The method proposed is a generalization and improvement of the proxy simulation scheme method (Fries and Kampen, 2005).

As a benchmark we apply the method to the pricing of digital caplets and target redemption notes using LIBOR and CMS indices under a LIBOR Market Model. We calculate stable deltas, gammas and vegas by applying direct finite difference to the proxy simulation scheme pricing.

The framework is generic in the sense that it is model and almost product independent. The only product dependent part is the specification of the proxy constraint. This allows for an elegant implementation, where new products may be included at small additional costs.

Contents

Version 1.1.3 (20070506)
http://www.christian-fries.de/finmath/proxyscheme

1 Introduction

1.1 Proxy Simulation Scheme

We propose an improvement and extension to the proxy simulation scheme method proposed in [5]. Proxy simulation schemes are a method of simplifying the development of prices and sensitivities in Monte Carlo simulations by allowing perturbations of inputted data to be shifted from the evolution to a likelihood ratio weight. This can both greatly simplify numerical implementations and lead to much reduced variances for Monte Carlo simulations. Our fundamental innovation in this paper is that our proxy is partial in that we arrange for some rates to have the same evolution whilst allowing others to vary. This enables us to achieve stable Greeks in low-factor LIBOR models.

The proxy simulation scheme method considers two numerical schemes, the proxy scheme K^0 and the target scheme K^*, where in the usual applications K^* is a perturbation of K^0. The method replaces a direct simulation using K^* by a weighted simulation using the original paths of K^0. The weights are given by a measure change (the Radon-Nykodym derivative a.k.a. likelihood ratio). Since the measure change is considered on the level of a numerical scheme one may derive analytic formulas for it. However, there are situations when such a measure change does not exist.

In this paper we introduce the *k-dimensional partial proxy simulation scheme* K^1. The scheme K^1 coincides with K^0 only on a k-dimensional sub-manifold given by a rank k projection operator Π. If Π is the identity our definition of K^1 still differs from K^0 such that the measure change from K^* to K^1 is always well defined. In the case where the measure change from K^* to K^0 is well defined, the scheme K^0 and K^1 coincide, otherwise K^1 is the (in some sense) closest approximation to K^0.

This approach has the following advantages

- **Equivalence with target scheme:** Using the weighted paths of K^1 will give the same results as using the paths of K^*.
- **Proxy property:** The projection operator Π may be chosen such that certain the path-wise values of a k-dimensional sub-manifold of K^1 coincides with K^0. This will result in robust finite differences with respect to these quantities.

Again, note that for a full proxy simulation scheme (K^0), as originally proposed in [5], the first property may fail.

1.2 Application: Monte-Carlo Sensitivities (Partial Derivatives) of Financial Derivatives with Trigger Feature

An application of the method proposed is given by the approximation of partial derivatives[1] of some expectation (e.g. the price of a financial product), calculated by applying finite-differences to the expectation calculated by Monte-Carlo simulation. The finite difference applied to a Monte-Carlo expectation is a Monte-Carlo expectation of the finite difference approximation of a path-wise derivative. It is usually of very poor quality if the path-wise values are discontinuous functions of the initial data. Discontinuous path-wise values are a property of financial derivatives with a trigger feature, where a (large) payment is *triggered* when some random variable crosses a threshold value. Different solutions to this problem have been considered by numerous authors, see, e.g. [2, 3, 6, 13, 14] and references therein.

Here we propose the use of the partial proxy simulation scheme, defining a scheme K^1 such that *the path-wise values of the trigger variable under K^1 coincide with the path-wise values of the trigger variable under K^0*. Thus the sensitivity with respect to a trigger change

[1] In the context of mathematical finance these partial derivatives are called *sensitivties* or *Greeks*.

is calculated by the differentiation of the likelihood ratio. Once the proxy constraint has been defined, the sensitivity is calculated by applying ordinary finite differences to the given Monte-Carlo pricing algorithm.

Our method can be explained intuitively by saying that we use weighted Monte-Carolo or importance sampling to fix quantities that give rise to discontinuities thereby avoiding the problems normally associated with computing Greeks of digital, trigger and barrier options, it therefore has some similarities to methods in [7, 10, 11].

As a benchmark example we consider the Greeks of LIBOR- and CMS-based target redemption notes (TARNs). Their Monte-Carlo calculation is know to be a very hard problem, see [14]. A target redemption note is similar to a digital payout, however the trigger criteria is highly non-trivial, especially if the TARN is based on index that is itself a non-linear function of the underlyings (like a CMS in a LIBOR market model). Our method solves the problem. Yet it is generic and not restricted to TARNs. The method works seamlessly for low-factor LIBOR market models, a situation where the (full) proxy scheme method proposed in [5] has only a limited scope of application or fails.

1.3 Plan of Paper

We start in Section 2 with a short review of the full proxy simulation scheme as proposed in [5]. In Section 3 we will define the k-dimensional partial proxy scheme, point out the differences to the full proxy scheme and derive an analytic formula for its Monte-Carlo weight when the scheme is an Euler scheme. In Section 4 we will state our benchmark application: the calculation of Greeks of a target redemption note under a Monte-Carlo simulation of a LIBOR Market Model. Section 5 will give the corresponding numerical results.

The paper assumes some basic knowledge of Monte-Carlo simulation and - for the chosen application - of risk neutral pricing. For details on this and our chosen benchmark model, the LIBOR Market Model, see [1, 6, 8, 12, 16].

Version 1.1.3 (20070506)
http://www.christian-fries.de/finmath/proxyscheme

2 Proxy Simulation Scheme Revisited

We consider two schemes K° and K^{*} and assume that we know the conditional transition probability densities $\phi^{K^{\circ}}$ and $\phi^{K^{*}}$ for K° and K^{*} respectively. Under the assumption that

$$\phi^{K^{\circ}}(t_{i+1}, y, t_i, x) = 0 \quad \Longrightarrow \quad \phi^{K^{*}}(t_{i+1}, y, t_i, x) = 0 \qquad \forall\, i, x, y \tag{1}$$

– i.e. the space spanned by the scheme K^{*} is a subspace of the space spanned by the scheme K°

$$\{K^{*}(t_i, \omega)\,|\omega \in \Omega\} \subset \{K^{\circ}(t_i, \omega)\,|\omega \in \Omega\} \quad \forall i$$

– we may move from realizations K° to K^{*} through a change of measure: Instead of the simulation scheme K^{*} we use the simulation scheme K° and perform a change of measure by $\frac{\phi^{K^{*}}}{\phi^{K^{\circ}}}$. For the expectation operator we have

$$\begin{aligned}
&\mathrm{E}^{\mathbb{Q}}\big(f(K^{*}(t_0), K^{*}(t_1), \ldots, K^{*}(t_n))|\mathcal{F}_{t_k}\big) \\
&= \mathrm{E}^{\mathbb{Q}}\left(f(K^{\circ}(t_0), K^{\circ}(t_1), \ldots, K^{\circ}(t_n)) \cdot \prod_{i=k}^{n-1} \frac{\phi^{K^{*}}(t_i, K^{\circ}(t_i); t_{i+1}, K^{\circ}(t_{i+1}))}{\phi^{K^{\circ}}(t_i, K^{\circ}(t_i); t_{i+1}, K^{\circ}(t_{i+1}))}|\mathcal{F}_{t_k}\right)
\end{aligned} \tag{2}$$

This is immediately clear using the integral representation of $\mathrm{E}^{\mathbb{Q}}$ with the above densities.

This enables us to price a derivative under the scheme K^{*} by (re-)using the realizations of the scheme K°.

2.1 Transition Probabilities

If our numerical schemes are Euler-Schemes

$$\begin{aligned}
K^{0}(t_{i+1}) &= K^{0}(t_i) + \mu^{0}(t_i)\Delta t_i + \Gamma^{0}(t_i)\Delta W(t_i) \\
K^{*}(t_{i+1}) &= K^{*}(t_i) + \mu^{*}(t_i)\Delta t_i + \Gamma^{*}(t_i)\Delta W(t_i)
\end{aligned}$$

then, using the known transition probability of the Brownian increment

$$\phi^{W}(t_i, x, t_{i+1}, y) \;=\; \frac{1}{(2\pi\Delta t_i)^{m/2}} \exp\big(-\frac{\|y-x\|^2}{\Delta t_i}\big)$$

we trivially have for the transition probability of the scheme realizations that

$$\begin{aligned}
\phi^{0}(t_i, K^{0}(t_i), t_{i+1}, K^{0}(t_{i+1})) &= \phi^{W}(t_i, W(t_i), t_{i+1}, W(t_{i+1})) \\
\phi^{*}(t_i, K^{*}(t_i), t_{i+1}, K^{*}(t_{i+1})) &= \phi^{W}(t_i, W(t_i), t_{i+1}, W(t_{i+1})).
\end{aligned}$$

From this we derive an analytic formula for $\phi^{*}(t_i, K^{0}(t_i), t_{i+1}, K^{0}(t_{i+1}))$ and thus for the Monte-Carlo weight in (2).

3 Partial Proxy Simulation Scheme

Let K^0 denote the unperturbed scheme and K^* some perturbation of K^0, e.g. a scheme with different initial data. We will call K^0 the reference scheme and K^* the target scheme.

The usual procedure of bump-and-revalue for computing Greeks would simulate paths of K^* having Monte-Carlo weight $\frac{1}{n}$. The proxy simulation schemes would simulate paths of K^0 using Monte-Carlo weights $\frac{1}{n} \cdot \frac{\phi^*}{\phi^0}$. Instead, here we consider a third scheme K^1, the *(partial) proxy simulation scheme* where paths are such that the path-wise values of some (but not all) components of K^1 (or a function thereof) agree with the corresponding path-wise quantities under K^0.

3.1 Linear Proxy Constraint

Let $\Pi(t_i)$ denote a projection operator of rank k. Let $v(t_i)$ be defined as

$$v(t_i) \ := \ (\Pi \cdot \Gamma(t_i))^{-1} \cdot (\Pi \cdot K^*(t_{i+1}) - \Pi \cdot K^0(t_{i+1})), \tag{3}$$

where $(\Pi \cdot \Gamma(t_i))^{-1}$ is the quasi inverse of $\Pi \cdot \Gamma(t_i)$, i.e. v is the solution of

$$\|\Pi \cdot K^0(t_{i+1}) - \Pi \cdot (K^*(t_{i+1}) - \Pi\Gamma(t_i) v(t_i))\|_{L_2} \ \rightarrow \ \min\,. \tag{4}$$

We define the *k-dimensional partial proxy scheme* K^1 as:

$$\begin{aligned} K^1(t_0) \ &:= \ K^*(t_0) \\ K^1(t_{i+1}) \ &:= \ K^*(t_{i+1}) - \Gamma(t_i) \cdot v(t_i). \end{aligned} \tag{5}$$

The scheme K^1 has the following properties:

- It coincides with K^0 on the k-dimensional sub-manifold defined by Π, i.e. $\Pi \cdot K^1(t_i) = \Pi \cdot K^0(t_i)$.
- It is given through a mean shift $v(t_i)$ on the Brownian increment $\Delta W(t_i)$ of the target scheme K^*.

Consequently, the Monte-Carlo weight of the partial proxy scheme is given by

$$w(t_i) \ = \ \frac{\phi^{K^*}(t_i, K^1(t_i); t_{i+1}, K^1(t_{i+1}))}{\phi^{K^1}(t_i, K^1(t_i); t_{i+1}, K^1(t_{i+1}))}.$$

In the case of a linear proxy constraint, the mean shift $v(t_i)$ is $\mathcal{F}_{t_i}$ measureable.[2] Then, using simple Euler schemes we have for the transition probabilities that

$$\begin{aligned} \phi^{K^1}(t_i, K^1(t_i); t_{i+1}, K^1(t_{i+1})) \ &= \ \phi^W(t_i, W(t_i), t_{i+1}, W(t_{i+1})) \\ \phi^{K^*}(t_i, K^1(t_i); t_{i+1}, K^1(t_{i+1})) \ &= \ \phi^W(t_i, W(t_i), t_{i+1}, W(t_{i+1}) - v(t_i)). \end{aligned} \tag{6}$$

From this we may derive $w(t_i)$ as a simple analytic formula, see Section 3.4.2.

We would like to note that in (4) we may replace the projection operator by a general nonlinear function, if necessary. We will discuss this case in Section 3.3 and we will consider this case in our benchmark application in Section 4.[3]

[2] We will later consider the general case of non-linear proxy constraints and $\mathcal{F}_{t_i}$ measureable mean shifts, see Section 3.3 and 3.4.

[3] For the case of the LIBOR Market Model a possible application would be a constraint depending on some swap rate.

3.2 Comparison to Full Proxy Scheme Method

The proxy simulation scheme proposed in [5] corresponds to $K^1 = K^0$. Thus, it is a special case of (3), (5) if Π is the identity and if

$$\Gamma(t_i)v(t_i) \ := \ K^*(t_{i+1}) - K^0(t_{i+1}) \tag{7}$$

has a solution $v(t_i)$ (not only in the sense of a closest approximation). If however (7) has no solution, $v(t_i)$ from (3) still defines a valid mean shift for the scheme K^*. The scheme K^1 will be the closest approximation to K^0 fulfilling the measure continuity condition with respect to K^*.

A major advantage of the partial proxy scheme is that the projection Π may be chosen such that (3) has an exact solution with respect to the sub manifold defined by Π, so K^1 and K^0 coincide on a k-dimensional sub manifold. We will make use of this in our benchmark example, see Section 4.3.

3.3 Non-Linear Proxy Constraint

An obvious (and commonly required) generalization is to replace the linear projection operator Π by a general, possibly non-linear function $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$ and define $v(t_i)$ as the solution of

$$f(t_{i+1}, K^0(t_{i+1})) = f(t_{i+1}, K^*(t_{i+1}) - \Gamma(t_i) \cdot v(t_i)). \tag{8}$$

Thus we have $f(t_{i+1}, K^0(t_{i+1})) = f(t_{i+1}, K^1(t_{i+1}))$. An application of this generalization is, e.g., a LIBOR Market Model, where f represents a certain swap rate or function of swap rates (e.g. a CMS spread). The condition will then ensure that the path value of the swap rate(s) are the same under K^0 and K^1.

3.3.1 Linearization of the Proxy Constraint

While a constraint like (8) will be the general application, its numerical implementation may be expensive, since one has to solve the non-linear equation on every path in every time-step. However, if $K^*(t_{i+1})$ is a small perturbation of $K^0(t_{i+1})$, we may linearize Equation 8. In other words we would set

$$\Pi \ := \ f'(K^0(t_{i+1})). \tag{9}$$

Note that the proxy simulation method is constructed such that finite-difference using small perturbation will remain stable, i.e. $K^*(t_{i+1})$ may be chosen to be arbitrarily close to $K^0(t_{i+1})$.

3.3.2 Finite Difference Approximation of the Non-Linear Proxy Constraint

The linearization (9) of f may still result in relatively large computational costs, because the projection operator has to be calculated on every paths. Note that we linearize around $K^0(t_{i+1}, \omega)$. Thus the quasi-inverse of $\Pi\Gamma$ has to be calculated on every path in every time-step. If one wishes to implement a faster calculation of the mean shift $v(t_i, \omega_j)$ one may calculate an approximate solution of (8) using a guessed directional shift $\tilde{v}(t_i)$ and finite differences to determine the shift size:

Assume we knew that the directional shift $\tilde{v}(t_i)$ does not lie in $\mathrm{Kern} f'\Gamma$. Then for some $\epsilon > 0$ calculate

$$\Delta_{-\epsilon\Gamma\tilde{v}(t_i)} f \ := \ \frac{1}{\epsilon}(f(t_{i+1}, K^*(t_{i+1}) - \epsilon \cdot \Gamma(t_i) \cdot \tilde{v}(t_i)) - f(t_{i+1}, K^0(t_{i+1}))) \tag{10}$$

and set

$$\Gamma \cdot v(t_i) := (\Delta_{-\epsilon\tilde{v}(t_i)})^{-1} \cdot \tilde{v}(t_i) \tag{11}$$

in the definition of the partial proxy scheme K^1, (5).

This solution has the desirable property, that its implementation allows the constraint function f to be specified exogenously by the user; this constraint function may vary with the application.

Example: If K is the $\log$ of the forward rates under a LIBOR Market Model and f is some swap rate, i.e. we would like to keep a swap rate rigid, then we may achieve this by modifying the first factor. This corresponds to $\tilde{v}(t_i) = (1, 0, \ldots, 0)$. From (10) we calculate the impact of a shift of the first factor on the swap rate, from (11) we calculate the required magnitude of this shift (it is scalar equation with a scalar unknown $v_1(t_i)$).

We will consider a constraint like (8) next, in our benchmark application, a trigger option on an index like a CMS swap rate considered under the LIBOR Market Model.

3.4 Transition Probability from a Nonlinear Proxy Constraint

3.4.1 The Proxy Constraint Revistied

There is subtle but crucial detail in the definition of the mean shift $v(t_i)$: It is defined by comparing $K^*(t_{i+1})$ to $K^0(t_{i+1})$

$$f(t_{i+1}, K^0(t_{i+1})) = f(t_{i+1}, K^*(t_{i+1}) - \Gamma(t_i) \cdot v(t_i)), \tag{12}$$

not by comparing $K^*(t_i)$ to $K^0(t_i)$. Thus, in general, $v(t_i)$ is a $\mathcal{F}_{t_{i+1}}$-measurable random variable, but not $\mathcal{F}_{t_{i+1}}$-measurable.[4] If we would define $v(t_i)$ through

$$f(t_{i+1}, K^0(t_i)) = f(t_{i+1}, K^*(t_i) - \Gamma(t_i) \cdot v(t_i)),$$

then it is not guaranteed that

$$f(t_{i+1}, K^0(t_{i+1})) = f(t_{i+1}, K^*(t_{i+1}) - \Gamma(t_i) \cdot v(t_i)),$$

holds, after the drift and the diffusion from t_i to t_{i+1} has been applied. To account for the drift we could define $v(t_i)$ through

$$f(t_{i+1}, K^0(t_i) + \mu^0(t_i)\Delta t_i) = f(t_{i+1}, K^*(t_i) + \mu^*(t_i)\Delta t_i - \Gamma(t_i) \cdot v(t_i)). \tag{13}$$

By this we would have that $v(t_i)$ is a $\mathcal{F}_{t_i}$-measurable random variable. But still it is not guaranteed that the proxy constraint holds after the diffusion has been applied. However, it will be the case for linear constraints.

From this consideration it becomes obvious that for the linearization of the proxy constraint, we would have to linearize around $K^0(t_{i+1})$ and not around $K^0(t_i)$. As a solution of this linearization $v(t_i)$ will be $\mathcal{F}_{t_{i+1}}$-measurable only.

If the mean shift $v(t_i)$ is defined through (12) as a $\mathcal{F}_{t_{i+1}}$-measurable random variable it mean - using Euler schemes - that $v(t_i)$ depends non-linearly on the increment $\Delta W(t_i)$ and the formula for the corresponding transition probability involve inverting this dependence. We give two examples:

[4] If the following we will call $v(t_i)$ being $\mathcal{F}_{t_{i+1}}$*-measurable only*, if it is $\mathcal{F}_{t_{i+1}}$-measurable, but not $\mathcal{F}_{t_i}$-measurable.

Version 1.1.3 (20070506)
http://www.christian-fries.de/finmath/proxyscheme

3.4.2 Transition Probabilites for General Proxy Constraints

If the proxy constraint on time t_{i+1} is linear, then it may be realized by an $\mathcal{F}_{t_i}$ measurable mean-shift $v(t_i)$. In this case the calculation of the transition probabilities that form the Monte-Carlo weight leads to very simple formulas. From (6) we find that for a $\mathcal{F}_{t_i}$ measurable mean-shift that

$$w(t_i) = \prod_{k=1}^{m} \exp\left(- \frac{(x_k - v_k(t_i))^2 + x_k^2}{2\Delta t_i} \right) \qquad \text{where } x_k := \Delta W_k(t_i) \tag{14}$$

If the mean shift $v(t_i)$ is only $\mathcal{F}_{t_{i+1}}$ measurable, then it is still possible to obtain simple analytic formula for the transition probability, however, this formula requires the differentiation of the functional dependence of $v(t_i)$ on the increment $\Delta W(t_i)$.

Consider the general case where the mean shift $v(t_i)$ depends on the Brownian increment $\Delta W(t_i)$, i.e.

$$v(t_i) = v(t_i, \Delta W(t_i)).$$

Define $\tilde{x} = g(x) := x - v(t_i, x)$. Obviously we have

$$\phi(\tilde{x})\mathrm{d}\tilde{x} \stackrel{\tilde{x}=g(x)}{=} \phi(g(x))\det\left(\frac{\partial v(t_i,x)}{\partial x}\right)\mathrm{d}x = \frac{\phi(g(x))\det\left(\frac{\partial g(x)}{\partial x}\right)}{\phi(x)}\phi(x)\mathrm{d}x. \tag{15}$$

Here x denotes the (realization of the) Brownian increment ΔW and ϕ denotes its probability density. Evaluating functions of $\tilde{x} = g(x)$ corresponds to pricing under the partial proxy scheme K^1, evaluating functions of x corresponds to the pricing under the target scheme K^*. From (15) we can read off the Monte-Carlo weights for the pricing under the scheme K^1 as

$$w(t_i) = \det\left(I - \frac{\partial v(t_i,x)}{\partial x}\right) \cdot \prod_{k=1}^{m} \exp\left(- \frac{(x_k - v_k(t_i))^2 + x_k^2}{2\Delta t_i} \right) \quad \text{where } x_k := \Delta W_k(t_i). \tag{16}$$

Obviously this result is not limited to the case of Euler schemes. The only requirement with respect to the scheme, it that it is generated by the Brownian increments $\Delta W(t_i)$ (e.g. as for a Milstein scheme). We summarise our result in a theorem:

Lemma (partial proxy): Let $K^*(t_i)$, $i = 0, 1, 2, \ldots$, denote a numerical scheme generated from the Brownian increments $\Delta W(t_i)$, $i = 0, 1, 2, \ldots$ (target scheme), i.e.

$$K^*(t_{i+1}) = K^*(t_{i+1}, K^*(t_i), \Delta W(t_i) - v(t_i))$$

Let $K^0(t_i)$, $i = 0, 1, 2, \ldots$ denote another numerical scheme, also generated from the Brownian increments $\Delta W(t_i)$ and close to K^*.

For a given function f (the proxy constaint) let $v(t_i)$ denote a solution of

$$f(t_{i+1}, K^*(t_{i+1}, K^*(t_i), \Delta W(t_i) - v(t_i))) = f(t_{i+1}, K^1(t_{i+1}))$$

and - assuming a solution exits - define the scheme K^1 by

$$K^1(t_{i+1}) := K^*(t_{i+1}, K^*(t_i), \Delta W(t_i) - v(t_i)).$$

Then the Monte-Carlo pricing under the scheme K^* is, in the Monte-Carlo limit, equivalent to the pricing under the scheme K^1 using the Monte-Carlo weights $\prod w_i$ with w_i given by (16).

We call the scheme K^1 the (partial) proxy scheme satisfying the proxy constrain $f(t_{i+1}, K^1(t_{i+1})) = f(t_{i+1}, K^0(t_{i+1}))$.

3.4.3 Example

Since we desire a generic and fast implementation we would like to discuss a special case, sufficiently general for all our applications and simple enough to give direct formulas for the transition probabilities:

Consider that $v(t_i)$ depends linear on the increment $\Delta W(t_i)$, i.e.

$$v(t_i) := A(t_i) \cdot \Delta W(t_i) + b(t_i),$$

with A and b being $\mathcal{F}_{t_i}$-measurable. Then we have for the mean-shifted diffusion

$$\Delta W(t_i) - v(t_i) \ = \ (1 - A(t_i)) \cdot \big(\Delta W(t_i) - b(t_i)\big).$$

This the corresponding transition probability is normal distributed with mean $b(t_i)$ and standard deviation $(1 - A(t_i))\sqrt{\Delta t_i}$. Note that if the target scheme is a small perturbation of the reference scheme, then $A(t_i)$ is small and $(1 - A(t_i))$ is non singular.

So here, the $\mathcal{F}_{t_{i+1}}$-measurable mean shift is given by an $\mathcal{F}_{t_i}$-measurable mean shift b and a scaling of the "factor" ΔW. We will make use of this in our next example: A proxy constraint stabilizing the calculation of vega, the sensitivity with respect to a change in the diffusion coefficient.

3.4.4 Approximating an $\mathcal{F}_{t_{i+1}}$-measurable proxy constraint by an $\mathcal{F}_{t_i}$-measurable proxy constraint

To allow for a fast calculation of the transition probability we propose to approximate the proxy constraint (12) by (13). From this we have that $v(t_i)$ is a $\mathcal{F}_{t_i}$ measurable mean-shift and the ratio of the transition probabilities is given by (14).

In addition we propose to linearize this constraint around $K^0(t_i) + \mu^0(t_i)\Delta t_i$, defining the linear proxy constraint by $\Pi := f'(K^0(t_i) + \mu^0(t_i)\Delta t_i)$.

All of our benchmark examples are based on the approximative constraint (13) or its linearization.

3.5 Sensitivity with respect to the Diffusion Coefficients – Vega

If we consider only an $\mathcal{F}_{t_i}$-measurable mean shift applied to the Brownian increment $\Delta W(t_i)$, then the method is not applicable to the calculation of a sensitivity with respect to the diffusion coefficient $\Gamma(t_i)$ - a.k.a. *vega*. The reason is simple: There is no $\mathcal{F}_{t_i}$-measurable mean shift that will ensure that the proxy constraint holds at t_{i+1} after a *different* ($\mathcal{F}_{t_{i+1}}$-measurable) diffusion has been applied – not even if the proxy constraint is a linear equation. Neglecting the Brownian increment, as suggested in 3.4.4, is a step in the wrong direction, since we are interested in the sensitivity with respect to the diffusion coefficient.

Of course, in our general formulation (12), a $\mathcal{F}_{t_{i+1}}$-measurable mean shift applied to the diffusion $\Delta W(t_i)$ will ensure that the proxy constraint holds at time t_{i+1}, even if the diffusion coefficient has changed. However, to obtain a simple formula for the transition probability and thus the Monte-Carlo weight $w(t_i)$ it is helpful to take an alternative view to the problem: The idea is similar to what is done for the case of a full proxy scheme (see [5]): We modify the diffusion of the proxy scheme to match the diffusion of the reference scheme and calculate the corresponding change of measure. In other words, we use the unperturbed diffusion coefficient for the (partial) proxy scheme. This adjustment is done prior to the calculation of the mean shift $v(t_i)$ for the corresponding proxy constraint, which will correct additional differences in the drift, if any.

From the consideration of the previous section it is clear, that this is equivalent to specifying a $\mathcal{F}_{t_{i+1}}$-measurable mean shift, being linear in the Brownian increment $\Delta W(t_i)$.

We give a stunning example of a vega calculation for a digital caplet in Section 5.1.2.

4 Benchmark Model and Benchmark Product: LIBOR Market Pricing of Target Redemption Notes

4.1 LIBOR Market Model

As application and to conduct some benchmark calculations we will consider option price and sensitivity calculations within a LIBOR Market Model (LMM). Some challenging properties of the LIBOR Market Model motivated its choice as our benchmark framework for numerical calculations:

- The LMM is in general high dimensional and due to its non-linear drift it is not possible to represent the state variable as a function of a low-dimensional Markovian process. This makes Monte-Carlo simulation the natural choice.[5]
- The LMM features a non-linear state dependent drift such that standard direct simulation schemes exhibit approximations errors in the drift part of the SDE.
- The LMM is one of the most popular, yet powerful interest rate models in practice.

We consider a tenor structure $T_0 < T_1 < \ldots < T_{n+1}$ with forward rates $L_i := L(T_i, T_{i+1}) := \frac{P(T_i)-P(T_{i+1})}{P(T_i)\cdot(T_{i+1}-T_i)}$, where $P(T_i)$ denote the zero coupon bond maturing in T_i - see [1, 16].

4.1.1 SDE

The LIBOR Market Model, see [1, 16], models the forward rate curve $L = (L_1, \ldots, L_n)$ by the SDE

$$\mathrm{d}L_i = L_i \mu_i^L \mathrm{d}t + L_i \sigma_i \mathrm{d}W_i, \qquad i = 1, \ldots n \tag{17}$$

where $W = (W_1, \ldots, W_n)$ is a n-dimensional $\mathbb{Q}$-Brownian motion with instantaneous correlation matrix $t \mapsto R(t)$, i.e.

$$R(t) = (\rho_{i,j}(t))_{i,j=1,\ldots,n}, \qquad \mathrm{d}W_i(t)\mathrm{d}W_j(t) = \rho_{i,j}(t)\mathrm{d}t.$$

We denote the filtration generated by W by $\{\mathcal{F}_t\}_{t\in[0,\infty)}$, the corresponding filtered probability space by $(\Omega, \mathbb{Q}, \mathcal{F}, \{\mathcal{F}_t\}_{t\in[0,\infty)})$, with the usual assumptions on filtration, see [4, 15].

4.1.2 Driving Factors

Let $f_1(t), \ldots, f_m(t) \in \mathbb{R}^n$ denote the orthonormal eigenvectors (*factors*) corresponding to the non-zero eigenvalues (*factor loadings*) $\lambda_1(t) \geq \ldots \geq \lambda_m(t) \geq 0$ of $R(t)$.[6] Then we may write

$$\mathrm{d}W = F \cdot \sqrt{\Lambda} \cdot \mathrm{d}U, \tag{18}$$

where $t \mapsto F(t)$ is the $n \times m$-matrix of the factors $F = (f_1, \ldots, f_m)$, $\sqrt{\Lambda} = \mathrm{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_m})$ and $U = (U_1, \ldots, U_m)$ is an m-dimensional $\mathbb{Q}$-Brownian motion with mutually uncorrelated components U_i.[7] Writing $\Gamma = F \cdot \sqrt{\Lambda}$ we have $\mathrm{d}W_i = (\Gamma \cdot \mathrm{d}U)_i$. Note that

$$\Gamma \cdot \Gamma^{\mathrm{T}} = R.$$
$$\Gamma^{\mathrm{T}} \cdot \Gamma = \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_m).$$

[5] In common applications it is 20 to 40 dimensional SDE.

[6] Note that $R(t)$ is symmetric and thus all eigenvalues are real. We assume that the f_i are chosen as an orthonormal basis of the $\mathbb{R}^n$.

[7] Then $\mathrm{d}W \cdot \mathrm{d}W^{\mathrm{T}} = F \cdot \sqrt{\Lambda} \cdot \mathrm{d}U\mathrm{d}U^{\mathrm{T}} \cdot \sqrt{\Lambda} \cdot F = F\Lambda F^{\mathrm{T}}\mathrm{d}t = R\mathrm{d}t$.

4.1.3 Drift

We consider (17) under the terminal measure, i.e. $\mathbb{Q}$ denotes the equivalent martingale measure corresponding to the numeraire $N(t) := P(T_{n+1}; t)$, where $P(T_{n+1})$ denotes the zero coupon bond with maturity T_{n+1}. The drift, μ_i^L, is given by

$$\mu_i^L = -\sum_{i<j\leq n} \frac{L_j \delta_j}{1 + L_j \delta_j} \sigma_i \sigma_j \rho_{i,j}. \tag{19}$$

4.1.4 Log-Coordinates

Using log coordinates $K := \log(L)$, where $L = (L_1, \ldots, L_n)$, $K = (K_1, \ldots, K_n)$ we rewrite (17) (using vector notation) as

$$\mathrm{d}K = \mu^K \mathrm{d}t + \Sigma \cdot \Gamma \cdot \mathrm{d}U, \tag{20}$$

where $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$, $\mu^K = (\mu_1^K, \ldots, \mu_n^K)$ and $\mu_i^K = \mu_i^L - \frac{1}{2}\sigma_i^2$ by Itô's Lemma, [8, 15].

4.1.5 Factor reduction

It is common to use a rather low-dimensional Brownian U motion in (18). Common values for the dimension of the Brownian motion are $m \leq 5$, whereas common applications require $n \geq 20$. Using a low-dimensional driving Brownian motion has the advantage that the computational complexity of a time step is proportional to the number of factors, see [9], and, in addition, the Monte-Carlo convergence is improved. To obtain Γ one may extract a few prominent factors via principal component analysis. The driving factors of a low-dimensional model may be associated with an intuitive interpretation of interest rate curve movements (parallel shift, tilt, etc.).

Thus R is singular in general and Γ^{T} acts as an projection onto an m-dimensional subspace (namely $D := (\mathrm{kern}(\Gamma))^{\perp}$). (Restricted to that subspace $\Lambda^{-1} \cdot \Gamma^{\mathrm{T}}$ is the inverse of Γ).

To simplify notation we assume that Σ is a non-singular $n \times n$ matrix. This enables us to recover the stochastic increments $\mathrm{d}U$ from the increment of the realizations $\mathrm{d}K$ via

$$\mathrm{d}U = \Lambda^{-1}\Gamma^{\mathrm{T}}\Sigma^{-1}\big(\mathrm{d}K - \mu^K \mathrm{d}t\big).$$

4.1.6 Pricing

The price of an interest rate derivative[8] with a time t_k value (e.g. payout) $V(t_k, L(t_0), L(t_1), \ldots, L(t_k))$, depending on the interest rate realizations L at $t_0, \ldots, t_k$, is given as an expectation with respect to the measure $\mathbb{Q}$:

$$V(t_0) = N(0) \cdot \mathrm{E}^{\mathbb{Q}}\left(\frac{V(t_k, L(t_0), L(t_1), \ldots, L(t_k))}{N(t_k)} \,\Big|\, \mathcal{F}_0\right) \tag{21}$$

where $N(t_k)$ denotes the numeraire which we have chosen to be $N(t_k) = P(t_n; t_k)$ (this implies the expression of μ^L in (19)). Since $N(t_k)$ itself is a function of $L(t_k)$ we have

$$\begin{aligned} V(t_0) &= \mathrm{E}^{\mathbb{Q}}(f(t_k, L(t_0), L(t_1), \ldots, L(t_k))|\mathcal{F}_0) \\ &= \int_{\mathbb{R}^{k\times n}} f(t_k, L(t_0), L(t_1), \ldots, L(t_k)) \cdot \\ &\qquad \cdot \phi(t_k, L(t_0), L(t_1), \ldots, L(t_k))\, \mathrm{d}(L(t_0), L(t_1), \ldots, L(t_k)), \end{aligned} \tag{22}$$

[8] To be precise: the price of the corresponding replication portfolio.

where ϕ denotes the probability density.[9]

4.2 Target Redemption Notes

As benchmark product we will consider target redemption notes. We will consider a reverse LIBOR floater and a reverse CMS floater. The latter requires a proxy constraint on the CMS rate, which is non-linear in the LIBOR rates. The Monte-Carlo calculation of delta and gamma of a target redemption note (TARN) has been known to be a difficult problem. For other methods to treat this see, e.g., [14]. We give a short definition of the TARN:

Let $0 = T_0 < T_1 < T_2 < \ldots < T_n$ denote a given tenor structure. For $i = 1, \ldots, n-1$ let C_i denote a (generalized) "interest rate" (the coupon) for the periods $[T_i, T_{i+1}]$, respectively. We assume that C_i is a $\mathcal{F}_{T_i}$-measurable random variable (natural fixing). Furthermore let N_i denote a constant value (notional). A *target redemption note* pays

$$N_i \cdot X_i \qquad \text{at} \qquad T_{i+1},$$

where

$$\begin{aligned} X_i := \quad & \begin{cases} C_1 & \text{for } i = 1, \\ \min(C_i \,, K - \sum_{k=1}^{i-1} C_k) & \text{for } i > 1 \end{cases} && \text{(structured coupon)} \\ + & \begin{cases} 1 \quad \text{for } \sum_{k=1}^{i-1} C_k < K <= \sum_{k=1}^{i} C_k \text{ or } i = n, \\ 0 \quad \text{else.} \end{cases} && \text{(redemption)} \\ + & \begin{cases} \max(0 \,,\ K - \sum_{k=1}^{i} C_k) & \text{for } i = n \\ 0 & \text{else.} \end{cases} && \text{(target coupon guarantee).} \end{aligned}$$

The payoff of the target redemption note contains the discontinuous (digital) part

$$\begin{cases} 1 \quad \text{for } \sum_{k=1}^{i-1} C_k < K <= \sum_{k=1}^{i} C_k, \\ 0 \quad \text{else.} \end{cases}$$

The product can either be structured as a swap or as a note. In the case of a swap, the floating leg is paid until the fixed target coupon for the other leg is reached so a small change in the total coupon paid can make a large difference to the value. Similarly for a note, the principal is repaid when the target coupon is reached, and so a small change in the value of the coupon can result in a large change in the deflated value of the principal repayment. These effects are particularly marked in the case of an increasing yield curve and a TARN with an inverse floater coupon, since in that case, the coupons rapidly become zero. So if the target coupon is not reached early in the deal, it will not be reached until the final maturity; a small shift in rates will mean the difference between two years and ten years of floating payments.

In order to calculate robust Monte-Carlo sensitivities a special treatment of this part is needed. We apply a partial proxy simulation scheme such that the path-wise coupon values $C_k(\omega)$ stay unchanged in order to surmount this digital effect – this will be enough to ensure that the digital effects disappear but will not guarantee the same deflated coupon value for every path since the numeraire is not required to be fixed.

[9] The integral in (22) is a Lebesgue integral. Abusing notation we write $\mathrm{d}L(t_i)$ for $\mathrm{d}x$ to show the link of the corresponding component of f and the density. Note that $L(t_0)$ and the 0-th component of the vector process L are non stochastic, thus we write (loosely) $\mathbb{R}^{k \times n}$ for the integration domain.

4.3 Partial Proxy Scheme

4.3.1 Choice of the Proxy Constraint

For our example, we want to have the path-wise values of the structured coupon C_k to be invariant under a change from K^0 to K^1, i.e.

$$C_k(K^1(t_k)) = C_k(K^0(t_k)).$$

If the coupon C_k depends on a single rate, say K_k, only, then we may take the projection operator $\Pi(t_i)$ for $T_{k-1} < t_i \leq T_k$ to be the one dimensional projection on the k-th component, i.e.

$$\Pi(t_i) \cdot K \ = \ (0, \ldots, 0, K_k, 0, \ldots, 0)^{\mathrm{T}}.$$

In other words, we require that $K^1(t_i)$ coincides with $K^0(t_i)$ in its k's component $T_{k-1} < t_i \leq T_k$.

If we consider the SDE under the spot measure numeraire, then this choice will also ensure that the value of the numeraire and thus the path wise value of any fixed cash flow agrees under K^1 and K^0.

4.4 Benchmark Model and Product Specifications

To allow for an easy reproduction of our benchmark results we used a simplified, but not totally unrealistic, model:

The LIBOR Market Model used in Section 5 modeled semi-annual forward rates, i.e. the tenor was $T_j := 0.5 \cdot j$, using a simple log-Euler scheme with (large) time steps $t_i = 0.5 \cdot i$. We used 5 factors extracted from the correlation model $\rho_{i,j} = \exp(-0.2|T_i - T_j|)$. Volatility was 0.2 flat for all rates.

For the digital caplets we used an initially flat curve $L_i(0) = 0.1$ and the digital caplets considered had strike $K = 0.1$. For the LIBOR and CMS tarn we used an upward sloping curve, linear from $L_0(0) = 0.02$ up to $L_0(10) = 0.10$.

5 Numerical Results

5.1 Digital Option on a LIBOR (Digital Caplet)

5.1.1 Delta and Gamma of a Digital Caplet

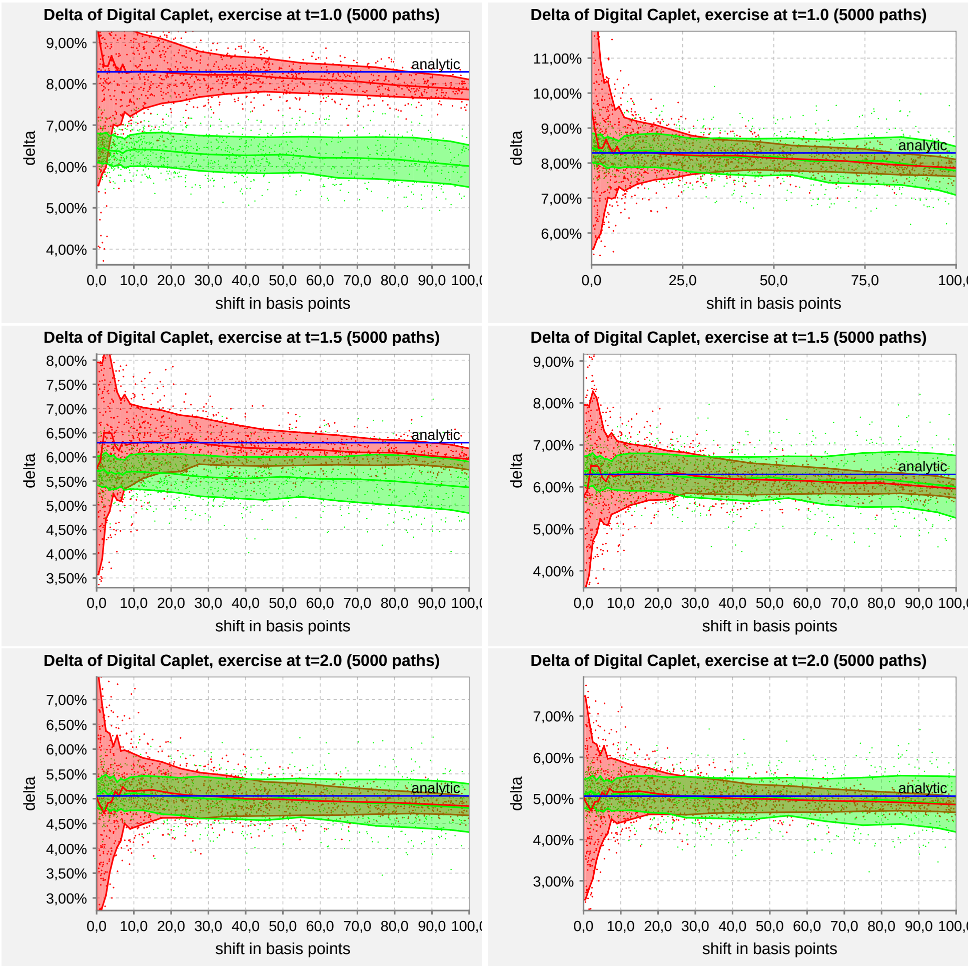

Figure 1: *Dependence of the Digital Caplet delta. Finite difference is applied to a direct simulation (red) and to an full (left) or partial (right) proxy scheme simulation (green). Each dot corresponds to one Monte-Carlo simulation with the stated number of paths. The red and green corridors represent the corresponding standard deviation.*
The proxy scheme simulation shows no dependence on the shift size while giving similar expected values for the sensitivity. The simulation is a low factor simulation and the finite difference shift is not in the span of the factors. Thus, for short maturities the proxy simulation scheme method will miss a portion of the sensitivity. The partial proxy scheme works for short maturities too.

We consider the calculation of delta and gamma of digital caplet, i.e. a digital option on a model primitive, a forward rate. Figure 1 compare full proxy simulation to partial proxy simulation. The LIBOR Market Model is a five factor model with correlation structure of $\mathrm{d}W_i \mathrm{d}W_j = \exp(-0.2|T_i - T_j|)$. The shift scenario is not in the span of the factors, thus, for short maturities, the proxy simulation scheme method will leave out a significant portion of the sensitivity. As was pointed out in [5], a possible solution to this problem is to calculate the sensitivity with respect to the orthogonal of the span of the factors via direct simulation (i.e. apply a decomposition of the scenario).

Instead, here we use the partial proxy simulation scheme. A possible proxy constraint is

$$L^1(T_j, T_{j+1}; t) \;=\; L^0(T_j, T_{j+1}; t) \qquad \forall\, t,$$

where T_j is the fixing of the digital option. For the calculation of the figures shown we used the same proxy constraint as for the TARN, which works too.

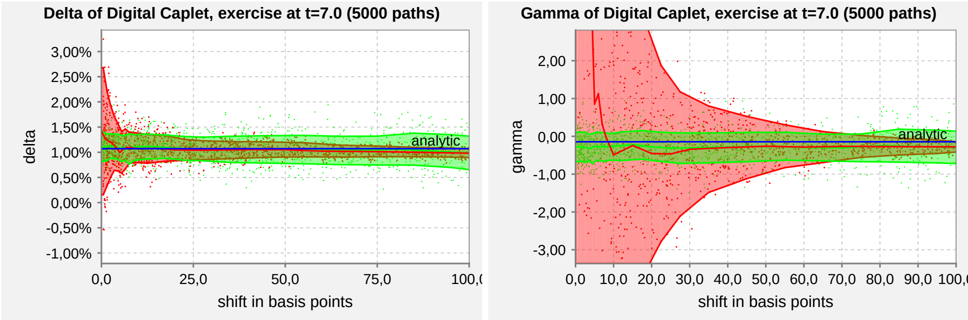

Figure 2: *Dependence of the Digital Caplet delta (left) and gamma (right) on the shift size of the finite difference approximation. Finite difference is applied to a direct simulation (red) and to a (partial) proxy scheme simulation (green). Each dot corresponds to one Monte-Carlo simulation with the stated number of paths. The red and green corridors represent the corresponding standard deviation.*
The proxy scheme simulation shows no dependence on the shift size while given similar expected values for the sensitivity.

5.1.2 Vega of a Digital Caplet

As mentioned in Section 3.5 the method may be applied to the calculation of the senstivity with respect to a diffusion coefficient (vega). We consider a digital caplet under a LIBOR Market Model, and there its sensitivity with respect to a change in the instantaneous volatility of the forward rate $L(3.0, 3.5)$. All other rates are simulated with unchanged diffusions. Thus we see a non-zero vega of the digital caplet with maturity $T = 3.0$ and the vega of all other digital caplets is zero.

Figure 3 shows the corresponding calculation with different shift sizes and Monte-Carlo seeds. As expected, direct finite differences develop an increase in Monte-Carlo variance while the partial proxy simulation scheme remains stable.

Even more stunning is the result depicted in Figure 4 showing the digital caplet with maturity $T = 2.0$. Analytically the vega is zero and the partial proxy simulation schemes reproduces this. However, applying finite differences to a direct simulation shows some Monte-Carlo variance for intermediate shift size. The reason for this is that - for the LIBOR Market Model - we have that a change of the diffusion of one rate enters into the drift of some other rates. This is a lower order effect. Analytically the pricing measure and numeraire ensure that the price of the corresponding digital caplet is not sensitive to a change of the diffusion coefficient of the other rate. But an individual paths of the rate is sensitive to a change of the diffusion coefficient of the other rate, hence we see a Monte-Carlo variance in the sensitivity. However, the proxy simulation scheme used a proxy constraint that kept all rates rigid upon their reset dates.

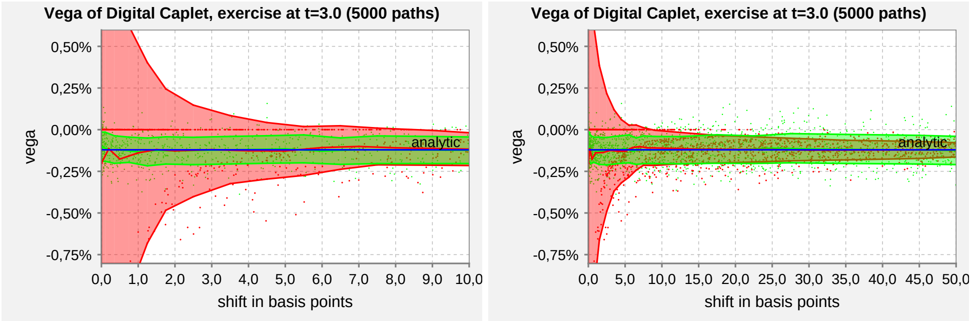

Figure 3: *Dependence of the Digital Caplet vega on the shift size of the finite difference approximation. The shift is applied to the diffusion coefficient for the underlying forward rate $L(3.0, 3.5)$. Finite difference is applied to a direct simulation (red) and to a (partial) proxy scheme simulation (green).*

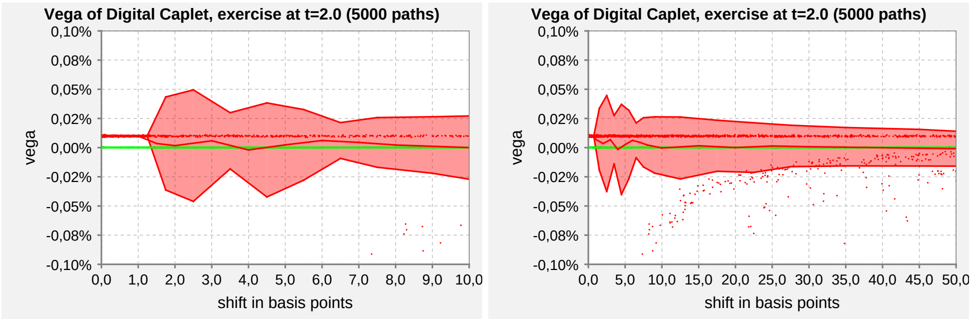

Figure 4: *Dependence of the Digital Caplet vega on the shift size of the finite difference approximation. The shift is applied to the diffusion coefficient for the forward rate $L(3.0, 3.5)$. Analytically the digital caplet has not sensitivity to this diffusion coefficient. Finite difference is applied to a direct simulation (red) and to a (partial) proxy scheme simulation (green).*

5.2 LIBOR Target Redemption Note

We calculate delta and gamma for a TARN swap. The coupon for the period $[T_i, T_{i+1}]$ is an inverse floater $\max(K - 2 \cdot L(T_i, T_{i+1}), 0)$ and it is swapped against floating rate $L(T_i, T_{i+1})$ until the accumulated coupon reaches a given target coupon. If the accumulated coupon does not reach the target coupon, then the difference to the target coupon is paid at maturity.

Thus the coupon of the tarn is linked to a trigger feature, similar to the digital caplet. However, here, the trigger depends on more than one rate, so it is not sufficient to setup a proxy constraint for a single forward rate, unlike for the digital caplet.

Our unperturbed scheme in this case is the LIBOR market model with the initial yield curve and an Euler approximation for the drift, whilst evolving the log. The natural perturbed scheme is then the same except for a different initial condition. We use the following proxy constraint:

$$L^1(T_j, T_{j+1}; t) \;=\; L^0(T_j, T_{j+1}; t) \qquad \forall\, t \in (T_{j-1}, T_j],$$

for all periods of the model to obtain the preferred proxy scheme. The constraint is realized by a mean shift of the diffusion of the first factor, since the forward rate follows a log-normal process we explicitly have $v = (v_1, 0, \ldots, 0)$ with

$$v_1(t) \;=\; \frac{\log(L^0(T_j, T_{j+1}; t)) - \log(L^1(T_j, T_{j+1}; t))}{f_{1,j}},$$

where $f_{1,j}$ denote the j-th component of the first factor. We assume here that $f_{1,j} \neq 0$. A non-zero factor loading exists as long as the forward rate $L(T_j, T_{j+1}$ has a non-zero volatility. The results can be improved, if the factor having the largest absolute factor loading is chosen (factor-pivoting).

In Figure 5 the delta and gamma of a TARN swap for different shift sizes of finite differences applied to standard re-simulation and partial proxy scheme simulation. For this example the interest rate curve was upward sloping from 2% to 10% and for the TARN we took $K = 10\%$ and a target coupon of 10%.

For small shifts the variance of the delta and gamma calculated under full re-evaluation blows-up and the mean becomes unstable, while for delta and gamma calculated under partial proxy scheme their mean remains stable and the variance small. For increasing shift size full re-evaluation stabilizes, but higher order effects give a significant bias. Very high shift increase the Monte-Carlo variance of the likelihood ratio and thus increase the variance of the delta and gamma calculated under the partial proxy scheme simulation.

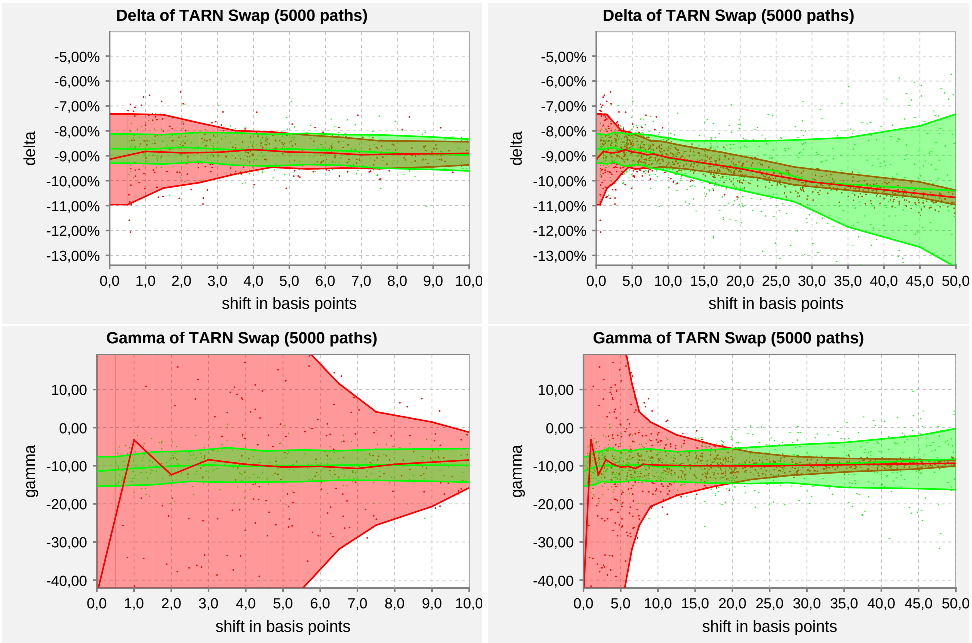

Figure 5: *Dependence of the TARN delta (top row) and gamma (bottom row) on the shift size of the finite difference approximation. Finite difference is applied to a direct simulation (red) and to a (partial) proxy scheme simulation (green). Each dot corresponds to one Monte-Carlo simulation with the stated number of paths. The red and green corridors represent the corresponding standard deviation.*
The proxy scheme simulation shows no variance increase for small shift sizes while giving stable expected values for the sensitivity.

5.3 CMS Target Redemption Note

Next we consider a target redemption note with a coupon $\max(K - 2 \cdot I(T_i), 0)$, where the index $I(T_i)$ is a constant maturity swaprate, i.e. $I(T_i) = S_{i,i+k}(T_i)$ with

$$\begin{aligned} S_{i,i+k} &= \frac{P(T_i) - P(T_{i+k})}{\sum_{j=i}^{k-1}(T_{j+1} - T_j)P(T_{j+1})} = \frac{\frac{P(T_i)}{P(T_{i+k})} - 1}{\sum_{j=i}^{k-1}(T_{j+1} - T_j)\frac{P(T_{j+1})}{P(T_{i+k})}} \\ &= \frac{\prod_{l=i}^{i+k-1}(1 + L(T_l)(T_{l+1} - T_l)) \; - \; 1}{\sum_{j=i}^{k-1}(T_{j+1} - T_j)\prod_{l=j+1}^{i+k-1}(1 + L(T_l)(T_{l+1} - T_l))} \end{aligned}$$

The swaprate $S_{i,i+k}(t)$ is a non-linear function of the forward rate curve $L_j(t)$, $j = i, \ldots, i+k-1$ which we denote by S:

$$S_{i,i+k}(t) = S(L_i(t), \ldots, L_{i+k-1}(t)).$$

From the proxy simulation scheme we require that S under L^1 matches S under the reference scheme L^0. Our proxy constraint is therefore

$$S(L_i^1(t), \ldots, L_{i+k-1}^1(t)) \; = \; S(L_i^0(t), \ldots, L_{i+k-1}^0(t)).$$

We solve this equation by modifying the first factor, i.e. in each time step t_j we solve for a single scalar $v_1(t_j)$ such that

$$\begin{aligned} &S(L_i^*(t_{j+1}) + v_1(t_j) \cdot f_{1,i}, \ldots, L_{i+k-1}^*(t_{j+1}) + v_1(t_j) \cdot f_{1,i+k-1}) \\ &\qquad\qquad = S(L_i^0(t_{j+1}), \ldots, L_{i+k-1}^0(t_{j+1})) \end{aligned} \tag{23}$$

and define $L_i^1(t_{j+1}) := L_i^*(t_{j+1}) + v_1(t_j) \cdot f_{1,i}$.

To simplify and speed up the calculation, we (numerically) linearize equation (23) and get an explicit (first order) formula for v_1, see (11).

5.3.1 Delta and Gamma of a CMS TARN

The result for the calculation of delta and gamma is depicted in Figure 6. Using the simple linearized proxy constraint we see a small increase in Monte-Carlo variance for the gamma with very small shifts.

The linearized constraint remains stable for small shifts. However, using a few Newton iterations on the linearization solves the non-linear constraint and further improves the result for the gamma, see Figure 7.

5.3.2 Vega of a CMS TARN

We calculate the vega of a CMS TARN, here the sensitivity of the CMS TARN with respect to a parallel shift of all instantaneous volatilities. The result is depicted in Figure 8. For medium and large shift size the vega calculated from finite differences applied to a partial proxy is similar to the vega calculated from finite differences applied to direct simulation. However, note that for very small shift sizes (around 1 bp), the vega calculated from finite differences applied to direct simulation converges to a wrong value and that this result occurs with a very small Monte-Carlo variance.

The reason for this effect is that the shifts are too small to trigger a change in the exercise strategy. Hence, the vega calculated is the sensitivity conditional to no change in exercise strategy, which is of course a different thing, see [4].

This effect is also present for delta and gamma and for all trigger products, but it was not visible in the figures so far due to the scale of the shift sizes and the number of paths used there.

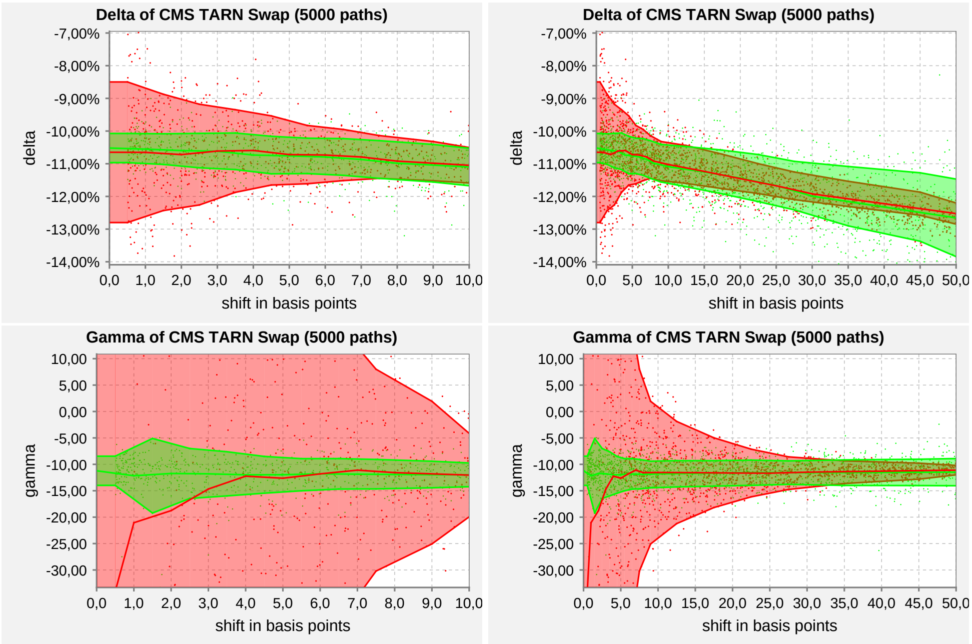

Figure 6: *Dependence of the CMS TARN delta (top row) and gamma (bottom row) on the shift size of the finite difference approximation. Finite difference is applied to a direct simulation (red) and to a (partial) proxy scheme simulation (green). The proxy constraint used was a simple (numerical) linearization of* (23).

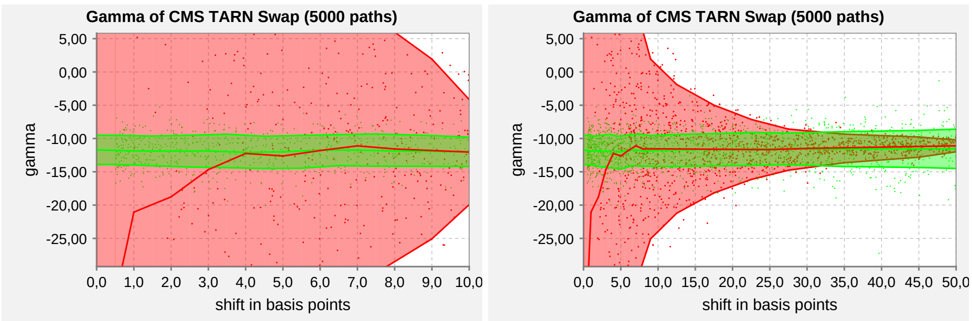

Figure 7: *Dependence of the CMS TARN gamma on the shift size of the finite difference approximation. Finite difference is applied to a direct simulation (red) and to a (partial) proxy scheme simulation (green). The proxy constraint was given by applying a few Newton iterations to the (numerical) linearization of* (23).

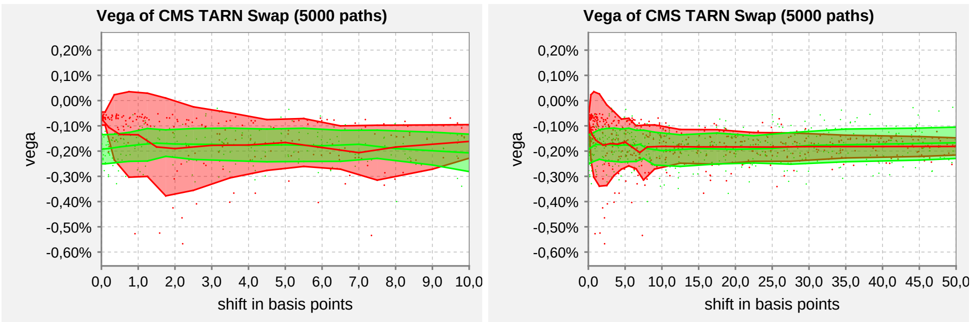

Figure 8: *Dependence of the CMS TARN vega on the shift size of the finite difference approximation. Finite difference is applied to a direct simulation (red) and to a (partial) proxy scheme simulation (green). The proxy constraint was given by applying a few Newton iterations to the (numerical) linearization of* (23).

6 Conclusions

The method presented solves the problem of the calculation of Monte-Carlo sensitivites (Greeks) for trigger products such as digitals, TARNs and barrier options. The method encompasses the notorious case of a TARN in a low-factor LIBOR market model. The framework is generic in the sense that it is model and almost product independent. The only product-dependent part is the specification of the proxy constraint. This allows for an elegant implementation, where new products may be included at small additional costs.

List of Symbols

Symbol	Meaning
$t_i, i = 0, 1, 2, \ldots$	Simulation time discretization, common to all numerical schemes.
$K^0(t_i)$	Realizations of the unperturbed simulation scheme (the reference).
$K^*(t_i)$	Realizations of the perturbed simulation scheme (the target scheme). $K^*(t_i)$ may be either an improved approximation of $K(t_i)$ or an perturbation of $K^\circ(t_i)$.
$K^1(t_i)$	Realizations of the partial proxy scheme.
ϕ^{K°	Transition probability density of K°.
ϕ^{K^*}	Transition probability density of K^*.
ϕ^{K^1}	Transition probability density of K^1.
ϕ^W	Transition probability density of the driving brownian motion W, common to all schemes.
$\mathrm{E}^{\mathbb{Q}}$	Expectation operator with respect to the measure $\mathbb{Q}$.
Π	A (linear) proxy constraint, given as a projection operator.
f	A (non-linear) proxy constraint. The quantity f coincides path-wise under K^1 and K^0.
$v(t_i)$	Shift applied to the Brownian increment $\Delta W(t_i)$ in order to solve the proxy constraint for t_{i+1}.
$T_j, j = 0, 1, 2, \ldots$	Tenor structure (discretisation of interest rate curve into forward rates).
$L = (L_1, \ldots, L_n)$	Vector of processes of forward rates L_i with period $[T_i, T_{i+1}]$, following a LIBOR Market Model.

References

[1] BRIGO, DAMIANO; MERCURIO, FABIO: Interest Rate Models - Theory and Practice. Springer-Verlag, Berlin, 2001. ISBN 3-540-41772-9.

[2] BOYLE, PHELIM; BOADIE, MARK; GLASSERMAN, PAUL: Monte Carlo methods for security pricing. Journal of Economic Dynamics and Control, 21, 1267-1321 (1997).

[3] BROADIE, MARK; GLASSERMAN, PAUL: Estimating Security Price Derivatives using Simulation. Management Science, 1996, Vol. 42, No. 2, 269-285.

[4] FRIES, CHRISTIAN P.: Mathematical Finance. Theory, Modeling, Implementation. John Wiley & Sons, 2007. ISBN 0-470-04722-4. http://www.christian-fries.de/finmath/book.

[5] FRIES, CHRISTIAN P.; KAMPEN, JÖRG: Proxy Simulation Schemes for generic robust Monte-Carlo sensitivities, process oriented importance sampling and high accuracy drift approximation. *Journal of Computational Finance*, 10-2. (2006). http://www.christian-fries.de/finmath/proxyscheme

[6] GLASSERMAN, PAUL: Monte Carlo Methods in Financial Engineering. (Stochastic Modelling and Applied Probability). Springer, 2003. ISBN 0-387-00451-3.

[7] JAECKEL, P.: Monte Carlo Methods in Finance, Wiley 2002, ISBN 0-471-49741-X.

[8] JOSHI, MARK S.: The Concepts and Practice of Mathematical Finance. Cambridge University Press, 2003. ISBN 0-521-82355-2.

[9] JOSHI, MARK S.: Rapid computation of drifts in a reduced factor LIBOR Market Model. Wilmott Magazine, May 2003.

[10] JOSHI, MARK S.; KAINTH, DHERMINDER S.: Rapid computation of prices and deltas of nth to default swaps in the Li Model, Quantitative Finance, volume 4, issue 3, (June 04), pages 266 - 275

[11] JOSHI, MARK S.; LEUNG: Using Monte Carlo simulation and importance sampling to rapidly obtain jump-diffusion prices of continuous barrier options. 2005. http://ssrn.com/abstract=907386

[12] KLOEDEN, PETER E.; PLATEN, ECKHARD: Numerical Solution of Stochastic Differential Equations (Applications of Mathematics. Stochastic Modelling and Applied Probability Vol. 23). Springer Verlag, 1999. ISBN 3-540-54062-8.

[13] PITERBARG, VLADIMIR V.: Computing deltas of callable LIBOR exotics in forward LIBOR models. Journal of Computational Finance. 7 (2003).

[14] PITERBARG, VLADIMIR V.: TARNs: Models, Valuation, Risk Sensitivities. Wilmott Magazine, 2004.

[15] ØKSENDAL, BERNT K.: Stochastic differential equations: an introduction with applications. Springer-Verlag, 2000. ISBN 3-540-63720-6.

[16] REBONATO, RICCARDO: Modern Pricing of Interest-Rate Derivatives: The Libor Market Model and Beyond. Princeton University Press, 2002. ISBN 0-691-08973-6.

Notes

Suggested Citation

FRIES, CHRISTIAN P.; JOSHI, MARK S.: Partial Proxy Simulation Schemes for Generic and Robust Monte-Carlo Greeks. (2006).

Classification

Classification: MSC-class: 65C05 (Primary), 68U20, 60H35 (Secondary).
ACM-class: G.3; I.6.8.
JEL-class: C15, G13.

Keywords: Monte-Carlo Sensitivities, Likelihood Ratio, Importance Sampling, Greeks, Proxy Simulation Scheme, Digital Option, Binary Option, Trigger Product, Target Redemption Note.

26 pages. 8 figures. 0 tables.