Proxy simulation schemes using likelihood ratio weighted Monte Carlo

for generic robust Monte-Carlo sensitivities and high accuracy drift approximation with applications to the LIBOR Market Model

Christian P. Fries
email@christian-fries.de

Jörg Kampen
joerg.kampen@iwr.uni-heidelberg.de

April 12, 2005

Abstract

We consider a generic framework for generating likelihood ratio weighted Monte Carlo simulation paths, where we use one simulation scheme K° (proxy scheme) to generate realizations and then reinterpret them as realizations of another scheme K^* (target scheme) by adjusting measure (via likelihood ratio) to match the distribution of K^* such that

$$\mathrm{E}^{\mathbb{Q}}(f(K^*) \mid \mathcal{F}_t) \;=\; \mathrm{E}^{\mathbb{Q}}(f(K^\circ) \cdot w \mid \mathcal{F}_t). \tag{1}$$

This is done numerically in every time step, on every path.

This makes the approach independent of the product (the function f in (1)) and even of the model, it only depends on the numerical scheme.

The approach is essentially a numerical version of the likelihood ratio method [6] and Malliavin's Calculus [8, 14] reconsidered on the level of the discrete numerical simulation scheme. Since the numerical scheme represents a time discrete stochastic process sampled on a discrete probability space the essence of the method may be motivated without a deeper mathematical understanding of the time continuous theory (e.g. Malliavin's Calculus).

The framework is completely generic and may be used for high accuracy drift approximations and the robust calculation of partial derivatives of expectations w.r.t. model parameters (i.e. sensitivities, aka. Greeks) by applying finite differences by reevaluating the expectation with a model with shifted parameters. We present numerical results using a Monte-Carlo simulation of the LIBOR Market Model for benchmarking.

Acknowledgment

We are very grateful to Benedikt Wilbertz for his support. We would like to thank Marius Rott, Markus Meister and Benedikt Wilbertz for stimulating discussions.

MSC-class: 65C05 (Primary), 68U20, 60H35 (Secondary). ACM-class: G.3; I.6.8. JEL-class: C15, G13.
Keywords: Monte-Carlo, Likelihood Ratio, Malliavin Calculus, Sensitivities, Greeks

Contents

1 Introduction

Due to the high dimensionality of stochastic differential equations used in the pricing of financial derivatives, Monte Carlo methods are still one of the most important numerical tools for the calculation of financial derivative prices (which disguise as expectations) and risk parameters (Greeks, which are partial derivatives of expectations with respect to model parameters).

A standard Monte-Carlo simulation of a stochastic differential equation, e.g. an Itô process

$$\mathrm{d}K = \mu^K \mathrm{d}t + \Sigma \cdot \Gamma \cdot \mathrm{d}U, \quad K(0) = K_0 \tag{2}$$

defined over a filtered probability space $(\Omega, \mathbb{Q}, \mathcal{F}, \{\mathcal{F}_t\})$ fulfilling the usual conditions, is given by generating sample paths $\omega_1, \ldots, \omega_n$ of time-discrete realizations

$$K(t+\Delta t) = K(t) + \int_t^{t+\Delta t} \mu^K(\tau)\mathrm{d}\tau + \int_t^{t+\Delta t} \Sigma(\tau) \cdot \Gamma(\tau) \cdot \mathrm{d}U(\tau) \tag{3}$$

of (2). Since the integrals in (3) are usually not available in closed form the time-discrete process is approximated - e.g. by an Euler scheme -

$$K^*(t+\Delta t) = K^*(t) + \mu^{K^*}\Delta t + \Sigma \cdot \Gamma \cdot \Delta U$$

The Monte-Carlo approximation of the expectation $\mathrm{E}(f(K(T))\,|\mathcal{F}_0)$ of a function f of a realization $K(T)$ is then given by

$$\begin{aligned} \mathrm{E}(f(K(T))|\mathcal{F}_0) &= \int f(\kappa)\phi^K(\kappa - K_0)\mathrm{d}\kappa \\ &\overset{(5)}{\approx} \int f(\kappa)\phi^{K^*}(\kappa - K_0)\mathrm{d}\kappa \approx \frac{1}{n}\sum_{i=1}^{n} f(K^*(T,\omega_i)) \end{aligned} \tag{4}$$

where ϕ^K and ϕ^{K^*} denote the probability density of $K(T)$ and $K^*(T)$ respectively. To shorten notation we will drop the conditioning on $\mathcal{F}_0$ in the expectation and the $K(0)$, implicitly viewing the probabilities as transition probabilities depending on $K(0)$ as a parameter. We assume that the time discretization error is small, i.e. that the densities ϕ^K and ϕ^{K^*} are close

$$||\phi^K - \phi^{K^*}||_{C^0} < \epsilon. \tag{5}$$

The whole procedure involves two approximation errors: The first is the time discretization error, i.e. the distance of the two densities ϕ^K and ϕ^{K^*}, the second is the Monte-Carlo error, i.e. the error introduced by the approximation of the last integral in (4) through a sum.

1.1 Sensitivities

The simplest approach to calculate an approximation of the partial derivative of the expectation is to apply finite differences to the Monte-Carlo price (the sum in (4)) (bumping the simulation).

Let X denote any model parameter (e.g K_0, Σ, Γ) and let us assume that the densities ϕ^K and ϕ^{K^*} depend smoothly on X and are C^1 close to each other (close also in first derivative)

$$||\phi^K - \phi^{K^*}||_{C^1} < \epsilon. \tag{6}$$

Then one might differentiate the above approximation to get partial derivatives of the expectation (price) with respect to a model parameter (giving the risk measure):

$$\begin{aligned} \frac{\partial}{\partial X}\mathrm{E}(f(K(T))) &= \int f(\kappa)\frac{\partial \phi^K}{\partial X}(\kappa)\mathrm{d}\kappa \\ &\overset{(6)}{\approx} \int f(\kappa)\frac{\partial \phi^{K^*}}{\partial X}(\kappa)\mathrm{d}\kappa \overset{?}{\approx} \frac{1}{n}\sum_{i=1}^{n} f'(K^*(T,\omega_i)) \cdot \frac{\partial K^*}{\partial X}(T,\omega_i). \end{aligned} \tag{7}$$

In applications the partial derivative is numerically replaced by finite differences.

The last step in (7) holds only in a week sense and might not even be an "approximation" (thus we put a question mark here). E.g. if f is not smooth, say even discontinuous, the last term in (4) is discontinuous too, thus not differentiable.[1] Applying finite differences results in poor Monte-Carlo convergence rates since the Monte-Carlo integral has to accurately resolve a region of the size of the shift. This is the reason why finite differences applied to Monte-Carlo (bumping the simulation) has poor convergence rates for non-smooth functions f

This problem may be solved by using the Monte-Carlo approximation of the differentiated integral rather than differentiating the Monte-Carlo approximation, i.e. we consider

$$\begin{aligned} \frac{\partial}{\partial X}\mathrm{E}(f(K(T))) &= \int f(\kappa)\frac{\partial \phi^K}{\partial X}(\kappa)\mathrm{d}\kappa \\ &\overset{(6)}{\approx} \int f(\kappa)\frac{\partial \phi^{K^*}}{\partial X}(\kappa)\mathrm{d}\kappa = \int f(\kappa)\frac{\frac{\partial}{\partial X}\phi^{K^*}(\kappa)}{\phi^{K^*}(\kappa)}\phi^{K^*}(\kappa)\mathrm{d}\kappa \\ &\overset{!}{\approx} \frac{1}{n}\sum_{i=1}^{n} f(K^*(T,\omega_i))\cdot\frac{\frac{\partial}{\partial X}\phi^{K^*}(K^*(T,\omega_i))}{\phi^{K^*}(K^*(T,\omega_i))}. \end{aligned} \tag{8}$$

If we compare the last term in (8) with the one in (4) we see that the partial derivative is just the expectation of a weighted payoff function $f \cdot w$ where the weight is given by

$$w = \frac{\frac{\partial}{\partial X}\phi^{K^*}}{\phi^{K^*}} = \frac{\partial}{\partial X}\log\left(\phi^{K^*}\right)$$

Thus the sensitivity has a similar approximation error than the price. This is essentially the likelihood ratio approach of Broadie and Glasserman or the application of a Malliavin weight, see [2, 6, 8, 14]. It should be noted that (8) already exhibits a slight difference to the way the likelihood ratio or Malliavin weight is usually considered, namely that we consider the weight to be derived from the scheme K^e and not from the original scheme K, i.e. we first apply a time discretization of the scheme and then apply the likelihood ratio and not reverse. This tiny modification will become important in the following.

However - the approach of a Malliavin weight loses the comfort of taking finite differences by "bumping the model". Shifting the model parameter has the charming advantage that it may be applied to any model parameter without modification of the model implementation. It is a scenario analysis of the model implementation. Instead, for (8) to work one has to know the densities and derive new weights for any partial derivative operator.

1.2 Proxy Scheme

We modify the approach in (8) towards a more generic framework to which we may apply finite differences by shifting input parameters while retaining the smoothness and convergence properties of a likelihood ratio method:

In addition to K^* consider another scheme K° with probability density ϕ°. The probability density ϕ° should be close to ϕ but need not to be a very accurate approximation. Then consider the following Monte Carlo approximation:

$$\begin{aligned} \mathrm{E}(f(K(T))) &= \int f(\kappa)\phi^K(\kappa)\mathrm{d}\kappa \\ &\overset{(5)}{\approx} \int f(\kappa)\phi^{K^*}(\kappa)\mathrm{d}\kappa = \int f(\kappa)\frac{\phi^{K^*}(\kappa)}{\phi^{K^\circ}(\kappa)}\phi^{K^\circ}(\kappa)\mathrm{d}\kappa \\ &\approx \frac{1}{n}\sum_{i=1}^{n} f(K^\circ(T,\omega_i))\cdot\frac{\phi^{K^*}(K^\circ(T,\omega_i))}{\phi^{K^\circ}(K^\circ(T,\omega_i))} \end{aligned} \tag{9}$$

[1] It is in general a bad idea to differentiate an approximation if the terms are not close in C^1. The approximation property may be completely lost then.

1.2.1 Sensitivities

For deriving a partial derivative with respect to a model parameter X take the proxy scheme K° and its density ϕ^{K° fixed, i.e it does not depend on X. Since ϕ^{K^*} is C^2 close to ϕ^K we may differentiate this approximation and arrive at the likelihood ratio weighted Monte-Carlo

$$\begin{aligned}\frac{\partial}{\partial X}\mathrm{E}(f(K(T))) &= \int f(\kappa)\frac{\partial}{\partial X}\phi^{K}(\kappa)\mathrm{d}\kappa \\ &\overset{(6)}{\approx} \int f(\kappa)\frac{\partial}{\partial X}\phi^{K^*}(\kappa)\mathrm{d}\kappa = \int f(\kappa)\frac{\frac{\partial}{\partial X}\phi^{K^*}(\kappa)}{\phi^{K^\circ}(\kappa)}\phi^{K^\circ}(\kappa)\mathrm{d}\kappa \\ &\approx \frac{1}{n}\sum_{i=1}^{n} f(K^\circ(T,\omega_i))\cdot\frac{\frac{\partial\phi^{K^*}}{\partial X}(K^\circ)}{\phi^{K^\circ}(K^\circ)} \\ &= \frac{\partial}{\partial X}\Big(\frac{1}{n}\sum_{i=1}^{n} f(K^\circ(T,\omega_i))\cdot\frac{\phi^{K^*}(K^\circ)}{\phi^{K^\circ}(K^\circ)}\Big)\end{aligned} \tag{10}$$

Note that the differential operator only acts on ϕ^{K^*} since ϕ^{K° is assumed fixed. Also note that the Monte-Carlo weight is the same in (9) and (10) and that the differentiation is applied to the original Monte-Carlo approximation. For the implementation this means that the realizations are generated by one scheme. The model parameter enter to the Monte-Carlo weights $\frac{\phi^{K^*}(K^\circ)}{\phi^{K^\circ}(K^\circ)}$ only. Thus the sensitivities may be calculated generically by applying finite differences to the numerical implementation of the model (*bumping the model*).

1.2.2 High accuracy scheme

Beside the Monte-Carlo error, which may be reduced by choosing n larger, the approximations (9) and (10) still exhibit the time discretization error: The time discrete approximation scheme K^* differs from K (ie. ϕ^{K^*} differs from ϕ). In financial models (at least the LIBOR Market Model which we will consider) this difference often stems from the poor approximation of the drift integral and may be of similar order as the Monte-Carlo error. This error disappears for $\Delta t \rightarrow 0$, however small time steps will increase the consumption of computation resources (cpu time and memory).

Using a likelihood ratio weighted proxy scheme the accuracy of the scheme may be improved just through the explicit knowledge of a more accurate transition density ϕ^{K^*}. This can be obtained by analytic expansions of the fundamental solutions. In [12] the convergence of expansions with Varadhan metric ([18]) as leading term is shown. In [7] an even faster convergent expansion of the fundamental solution will be applied to the LIBOR Market Model.

1.2.3 Generic framework

Since ϕ^{K^*} and ϕ^{K° are densities of time-discrete numerical schemes it is in general possible to give them in closed form. We will in fact work with the transition probabilities for a single time-step (for an Euler scheme this is just a normal distribution with known parameter). Using a simple discrete simulation scheme K° we generate sample paths. Theses are reinterpreted as realizations of K^* and combined by a *path and time-step wise correction of the transition probability* density by the *likelihood ratio*.

We see the novelty and added value of this paper in the following points:

- We provide a generic framework for generating a likelihood ratio weighted discrete simulation scheme which may be used for more accurate pricing (through the possibility of much better drift approximation) and much more accurate sensitivity calculation.

- The approach considered here is applied to the numerical scheme, and thus essentially model and product independent. This contrasts to most applications of the likelihood ratio method and/or Malliavin calculus as they are being considered on the level of the time continuous model and applied to a specific product.
- We calculate the likelihood ratios (Malliavin weights) numerically in on the fly. The only ingredients is a formula for the transition probabilities.
- We apply the approach to the LIBOR Market Model with numerical results.
- Our reference implementation shows that the approach is very efficient in term of computational resources (cpu time and memory).

1.3 Layout of the paper

We will start in Section 2 by presenting the basics of the LIBOR Market Model, an interest rate model to which we will apply the method and which we will use to conduct out benchmark calculations. In Section 3 we will present some direct simulation schemes, among them the Euler scheme and the predictor corrector scheme. The latter was introduced for the LIBOR Market Model in [11] in order to reduce the time discretization error. Section 4 will briefly discuss the problem of sensitivities in Monte-Carlo. Section 5 then presents the proxy scheme with likelihood ratio weighted Monte-Carlo. Section 6 gives the data used in our benchmark calculations which we will present in Section 7 and 8.

2 The LIBOR Market Model

As application and to conduct some benchmark calculations we will consider option price and sensitivity calculation 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.[2]
- 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 [4, 17].

2.1 SDE

The LIBOR Market Model, see [4, 17], models the forward rate curve $L = (L_1, \ldots, L_n)$ by the SDE

$$\mathrm{d}L_i = L_i \mu_i^L \mathrm{d}t + L_i \sigma_i \mathrm{d}W_i, \qquad i = 1, \ldots n \tag{11}$$

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 [3, 16].

2.2 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 $R(t)$.[3] Then we may write

$$\mathrm{d}W = F \cdot \sqrt{\Lambda} \cdot \mathrm{d}U, \tag{12}$$

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.[4] Writing $\Gamma = F \cdot \sqrt{\Lambda}$ we have $\mathrm{d}W_i = (\Gamma \cdot \mathrm{d}U)_i$. Note that

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

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

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

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

2.3 Drift

We consider (11) under the terminal measure, i.e. $\mathbb{Q}$ denotes the equivalent martingale measure corresponding to the numéraire $N(t) := P(T_{n+1}; t)$, where $P(T_{n+1})$ denotes the zero coupon bond with maturity T_{n+1}. Then the drift μ_i^L is given by

$$\mu_i^L = \sum_{i<j\leq n} \frac{L_j \delta_j}{1 + L_j \delta_j} \sigma_i \sigma_j \rho_{i,j}. \tag{13}$$

2.4 Log-Coordinates

Using $\log$ coordinates $K := \log(L)$, where $L = (L_1, \ldots, L_n)$, $K = (K_1, \ldots, K_n)$ we rewrite (11) (using vector notation) as

$$\mathrm{d}K = \mu^K \mathrm{d}t + \Sigma \cdot \Gamma \cdot \mathrm{d}U, \tag{14}$$

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, [3, 16].

2.5 Factor reduction

It is common to use a rather low dimensional Brownian U motion in (12). 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 driving factors usually may be associated with an intuitive interpretation of interest rate curve movements (parallel shift, tilt, etc.) and, in addition, improves Monte-Carlo convergence. To obtain Γ one may extract a few prominent factors via principal component analysis.

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

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

2.6 Pricing

The price of an interest rate derivative[5] 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 $\mathbb{Q}$:

$$V(t_0) = N(0) \cdot \mathrm{E}^{\mathbb{Q}} \left(\frac{V(t_k, L(t_0), L(t_1), \ldots, L(t_k))}{N(t_k)} \,\Big|\, \mathcal{F}_0 \right) \tag{15}$$

where $N(t_k)$ denotes the numéraire which we have chosen to be $N(t_k) = P(t_n; t_k)$ (this implies the expression of μ^L in (13)). Since $N(t_k)$ itself is a function of $L(t_k)$ we have

$$\begin{aligned} V(t_0) &= \mathrm{E}^{\mathbb{Q}}(f(t_k, L(t_0), L(t_1), \ldots, L(t_k))|\mathcal{F}_0) \\ &= \int_{\mathbb{R}^{k\times n}} f(t_k, L(t_0), L(t_1), \ldots, L(t_k)) \cdot \\ &\qquad \cdot \phi(t_k, L(t_0), L(t_1), \ldots, L(t_k)) \, \mathrm{d}(L(t_0), L(t_1), \ldots, L(t_k)), \end{aligned} \tag{16}$$

where ϕ denotes the probability density.[6]

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

[6] The integral in (16) 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 (losely) $\mathbb{R}^{k\times n}$ for the integration domain.

3 (Standard) Monte-Carlo Simulation Schemes

3.1 Simulation Schemes

We consider a discretisation $0 = t_0 < t_1 < t_2 < \ldots$ of simulation time and ask for a simulation scheme for generating samples $K(t_i, \omega_j)$ of the time t_i-realizations $K(t_i)$ of the stochastic process K. This is usually done by generating sample path of a time-discrete process (*the scheme*) $K^\circ(t_i)$, $i = 0, 1, 2, \ldots$, where $K^\circ(t_i)$ is an approximation to $K(t_i)$.

3.1.1 Euler Scheme

One such scheme is the *Euler Scheme* of (14) given by

$$K^{\mathrm{e}}(t_{i+1}) = K^{\mathrm{e}}(t_i) + \mu^K(t_i, K^{\mathrm{e}}(t_i)) \cdot (t_{i+1} - t_i) + \Sigma \cdot \Gamma \cdot (U(t_{i+1}) - U(t_i)) \tag{17}$$

- see [13]. The Euler scheme realizations $K^{\mathrm{e}}(t_i)$ are crude approximation of the "true" realizations $K(t_i)$. If we assume that $\Sigma \cdot \Gamma$ is constant on (t_i, t_{i+1}) the discretisation error of the scheme is inaccurate integration of the drift term:

$$\int_{t_i}^{t_{i+1}} \mu^K(t, K(t))\, \mathrm{d}t \approx \mu^K(t_i, K^{\mathrm{e}}(t_i)) \cdot (t_{i+1} - t_i).$$

This drift approximation may be improved in several ways:

3.1.2 Predictor-Corrector Scheme

Another scheme of (14) is given by

$$\begin{aligned} K^{\mathrm{pc}}(t_{i+1}) = K^{\mathrm{pc}}(t_i) + \frac{1}{2}\big(\mu^K(t_i, K^{\mathrm{pc}}(t_i)) + \mu^K(t_{i+1}^-, K^{\mathrm{p}}(t_{i+1}))\big) \cdot (t_{i+1} - t_i) \\ + \Sigma \cdot \Gamma \cdot (U(t_{i+1}) - U(t_i)), \end{aligned} \tag{18}$$

where the *predictor* step is and (e.g.) an Euler scheme step

$$K^{\mathrm{p}}(t_{i+1}) = K^{\mathrm{pc}}(t_i) + \mu^K(t_i, K^{\mathrm{pc}}(t_i)) \cdot (t_{i+1} - t_i) + \Sigma \cdot \Gamma \cdot (U(t_{i+1}) - U(t_i))$$

and $\mu^K(t_{i+1}^-) := \lim_{t \nearrow t_{i+1}} \mu^K(t)$. Note: This obscurity in notation stems from the application to the LIBOR Market Model where it is common to use volatility functions which are defined piecewise on $[t_i, t_{i+1})$, in general piecewise constant, yet discontinuous. This will lead to a jump in drift at t_{i+1}. For piecewise constant Σ and Γ in (14) the term $\mu^K(t_{i+1}^-, \tilde{K}^{\mathrm{pc}}(t_{i+1}))$ may be replaced by $\mu^K(t_i, \tilde{K}^{\mathrm{pc}}(t_{i+1}))$.

For the LIBOR Market Model the predictor corrector scheme was considered in [11].

3.1.3 Trapezoidal Average Drift Scheme

In some applications the drift has a special dependency structure which allows to obtain the accuracy of the predictor-corrector scheme without the need to calculate the predictor step. This will make the scheme almost as fast as the simple Euler scheme.

Assume for example that the j-th component $\mu_j^K(t_i)$ of the drift is a function of $K_l(t_i)$ with $l > j$ only (in other words: the derivative of $\mu^K(t_i)$ with respect to $K(t_i)$ is an upper triangular matrix). This is the case for the LIBOR Market Model in terminal measure, see Section 2. Then it is trivial to solve the (implicit Euler scheme) equation

$$\begin{aligned} K^{\mathrm{ta}}(t_{i+1}) = K^{\mathrm{ta}}(t_i) + \frac{1}{2}\big(\mu^K(t_i, K^{\mathrm{ta}}(t_i)) + \mu^K(t_{i+1}^-, K^{\mathrm{ta}}(t_{i+1}))\big) \cdot (t_{i+1} - t_i) \\ + \Sigma \cdot \Gamma \cdot (U(t_{i+1}) - U(t_i)) \end{aligned} \tag{19}$$

for $K^{\mathrm{ta}}(t_{i+1})$ to obtain an explicit scheme. In an implementation this can be done effectively by looping backward over the component index j:
For $j = n, n-1, \ldots, 1$:

$$\begin{aligned} K^{\mathrm{ta}}_j(t_{i+1}) = K^{\mathrm{ta}}_j(t_i) &+ \frac{1}{2}\mu^K_j(t_i, K^{\mathrm{ta}}_{j+1}(t_i), \ldots, K^{\mathrm{ta}}_n(t_i)) \cdot (t_{i+1} - t_i) \\ &+ \frac{1}{2}\mu^K(t^-_{i+1}, K^{\mathrm{ta}}_{j+1}(t_{i+1}), \ldots, K^{\mathrm{ta}}_n(t_{i+1})) \cdot (t_{i+1} - t_i) \\ &+ \Sigma \cdot \Gamma \cdot (U(t_{i+1}) - U(t_i)). \end{aligned} \tag{20}$$

The trapezoidal average drift scheme uses the discrete drift

$$\mu^{K^{\mathrm{ta}}}(t_i) := \frac{1}{2}\big(\mu^K(t_i, K^{\mathrm{ta}}(t_i)) + \mu^K(t^-_{i+1}, K^{\mathrm{ta}}(t_{i+1}))\big)$$

which is just a trapezoidal integration rule for the true average drift

$$\int_{t_i}^{t_{i+1}} \mu^K(\tau)\mathrm{d}\tau,$$

hence its name.[7]

3.2 Transition probability of an Euler-type Monte-Carlo scheme

For a given path ω let $x := U(T_i)(\omega)$ and $y := U(T_{i+1})(\omega)$. “Solving” (17) for $\Delta U(T_i) := U(T_{i+1}) - U(T_i)$ we have

$$\Delta U(T_i) = \Lambda^{-1/2} F^{\mathrm{T}} \Sigma^{-1} \big(\Delta K^{\mathrm{e}} - \bar{\mu}^K(T_i)\Delta T_i\big). \tag{21}$$

and using the transition probability of $\Delta U(T_i)(\omega)$

$$\phi(T_i, x; T_{i+1}, y) = \frac{1}{(2\Pi\Delta T_i)^{n/2}} \exp(-\frac{(y-x)^2}{2\Delta T_i})$$

we get from $y - x = \Delta U(T_i)$ and (21)

$$\phi^{K^{\mathrm{e}}}(T_i, K_i; T_{i+1}, K_{i+1}) = \\ \frac{1}{(2\Pi\Delta T_i)^{n/2}} \exp\Big(-\frac{1}{2\Delta T_i}\ \big(\Lambda^{-1/2} F^{\mathrm{T}} \Sigma^{-1} \big(K_{i+1} - K_i - \bar{\mu}^K(T_i)\Delta T_i\big)\big)^2\Big).$$

[7] Note: A nice feature of the proxy scheme method which we introduce in Section 5 is that one may use implicit schemes (as target scheme).

4 Sensitivities

Sensitivities[8] are partial derivatives of $V(t_0)$ with respect to model parameters like the initial condition $L(t_0)$, the volatility Σ and correlation structure Γ. If we denote any of these model parameters by X we have (formally)

$$\begin{aligned}
&\frac{\partial V(t_0)}{\partial X} = \\
&= \int_{\mathbb{R}^{k\times n}} f(t_k, L(t_0), \ldots, L(t_k)) \cdot \frac{\partial}{\partial X} \phi(t_k, L(t_0), \ldots, L(t_k)) \, \mathrm{d}(L(t_0), \ldots, L(t_k)) \\
&= \int_{\mathbb{R}^{k\times n}} f(t_k, L(t_0), \ldots, L(t_k)) \cdot \frac{\frac{\partial}{\partial X} \phi(t_k, L(t_0), \ldots, L(t_k))}{\phi(t_k, L(t_0), \ldots, L(t_k)} \\
&\qquad\qquad \cdot \phi(t_k, L(t_0), \ldots, L(t_k) \, \mathrm{d}(L(t_0), \ldots, L(t_k)) \\
&= \mathrm{E}^{\mathbb{Q}}\Big(f(t_k, L(t_0), \ldots, L(t_k)) \cdot \frac{\partial}{\partial X} \log \big(\phi(t_k, L(t_0), \ldots, L(t_k))\big) \Bigm| \mathcal{F}_0 \Big),
\end{aligned}$$

which shows that the partial derivative is a weighted average of the payoff.[9]

4.1 Sensitivities in Monte-Carlo

Since the probability density of $(L(t_1), \ldots, L(t_k))$ is a C^∞ function of the model parameters $L(t_0)$, Σ and Γ and the expectation operator is a convolution of f with the corresponding probability density, the derivative price $V(t_0)$ is also C^∞ w.r.t. $L(t_0)$, Σ and Γ.

For a standard Monte-Carlo method *this property is lost* and the Monte-Carlo approximation of a price of a financial derivative with a discontinuous payout function (e.g. Digital Options, Trigger Products, see Section 8) is a discontinuous of the model parameters $L(t_0)$, Σ and Γ. This results in a poor convergence of finite difference approximation of partial derivatives w.r.t. these model parameters.

The method which we consider in Section 5 will retain the C^∞ property of the derivative price $V(t_0)$.

4.2 The Likelihood Ratio and Malliavin's Calculus

Sensitivities in Mone-Carlo has been considered under generalized assumptions and mathematically more rigorously in numerous papers, see [1, 2, 5, 6, 8, 9, 10, 15] and references therein.

Broadie and Glasserman considered sensitivities in Monte-Carlo and gave the interpretation of the weight w as *likelihood ratio*. Fournié et al. proved under a much more generalized setting that sensitivities are the weighted expectation of the payoff by applying Malliavin's calculus (*stochastic calculus of variations*) (the weight is referred to as *Malliavin weight*), [8, 14]. Benhamou showed that the *likelihood ratio* is the *Malliavin weight* with minimal variance, [2].

For fast calculation of sensitivities of LIBOR derivatives with smoother (continuous) payoffs or the class of callable LIBOR exotics see, e.g., [10, 15].

[8] Aka- *Greeks*.

[9] We abuse notation again and write $\mathrm{d}L(t_i)$ for $\mathrm{d}x$ to show the link of the corresponding component of f to the density. The integrals are Lebesgue integrals.

5 Proxy scheme with likelihood ratio weighted Monte-Carlo

5.1 The proxy simulation scheme method

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

$$\phi^{K^\circ}(T_{i+1}, y, T_i, x) = 0 \quad \Longrightarrow \quad \phi^{K^*}(T_{i+1}, y, T_i, x) = 0 \qquad \forall\, i, x, y \tag{22}$$

– 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

$$\begin{aligned}
&\mathrm{E}^{\mathbb{Q}}\big(f(K^*(t_0), K^*(t_1), \ldots, K^*(t_n) | \mathcal{F}_{t_k}\big) \\
&= \mathrm{E}^{\mathbb{Q}}\big(f(K^\circ(t_0), K^\circ(t_1), \ldots, K^\circ(t_n) \cdot \prod_{i=k}^{n-1} \frac{\phi^{K^*}(T_i, K^\circ(t_i); T_{i+1}, K^\circ(t_{i+1}))}{\phi^{K^\circ}(T_i, K^\circ(t_i); T_{i+1}, K^\circ(t_{i+1}))} | \mathcal{F}_{t_k}\big)
\end{aligned} \tag{23}$$

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°.

5.2 Note on the likelihood ratio method

Equation (23) is nothing more than a likelihood ratio Monte-Carlo method for a conditional expectation over one time step. The likelihood ratio is given by the distribution ratios of the two scheme. Likelihood ratio weighted Monte-Carlo methods are common in mathematical finance, see [9], and the method of choice for calculation of sensitivities of credit event linked products.

The additional step we take here is to consider the likelihood ratio at simulation scheme level and use the method to generate realizations of one scheme (K°) and change measure (via likelihood ration) towards realizations of another simulation scheme (K^*). The likelihood ratio method [6] is sometimes criticized to rely on the explicit knowledge of the corresponding probability density, which seems to limit the scope of application. Indeed, considering the continuous time SDE the large time step transition probability densities are usually not available in closed form. However, considering the likelihood ratio method at the level of the numerical scheme / numerical implementation (e.g. with any of the schemes of section 3) the transition probability density easy to derive. In [12] it is shown that certain expansion of the transition density corresponding to continuous time SDE (so called WKB expansion) is strongly convergent.

5.3 Scope of application

The advantage of this approach is

- The method may be used for weak schemes, where one does have an analytic formula ϕ^{K^*} but does not have efficient method for drawing realizations of K^*.
- The method may be used to calculate the expectation with respect to a perturbed version of the SDE while avoiding the generation of new paths. This will lead to perfectly smooth calculation of partial derivatives (sensitivities) even when f is discontinuous - Section 5.5 and 8.

- The method is very efficient in terms of memory consumption, thus allowing the precalculation of serval scenarios K^* of perturbations of K°.
- Existing simulation methods and implementations may be reused. The improvement of the simulation scheme may be obtained through the adjustment of the change of measure $\frac{\phi^{K^*}}{\phi^{K^\circ}}$, which is a known function of the realizations.
- The implementation of a weighted Monte-Carlo framework may be reused for importance sampling.

A disadvantage of this approach is

- The method will fail to correct the transition density ϕ° if condition (22) does not hold. Special care has to be taken to ensure that condition (22). Note that this is possible since one is free to choose or modify the underlying proxy scheme K°.

5.4 Singularity of Γ:

By condition (22) we require that the transition probability measure/density corresponding to the schemes K^* is absolutely continuous with respect to the transition probability measure/density corresponding to the scheme K°. This requirement might not be fulfilled if $\mathrm{rank}(\Gamma) = m < n$.

E.g. for a single Euler-Scheme simulation step we have from $\mathrm{rank}(\Gamma) = m < n$ that $K^e(t_{i+1})$ lies in an m-dimensional affine subspace, namely

$$K^e(t_{i+1}) \in K^e(t_i) + \mu^K(t_i, K^e(t_i)) \cdot \Delta t_i + D,$$

where $D := \mathrm{span}(\Gamma)$.

However, due to the nonlinearity of the drift $\mu^K(T_i)$ we have for the solution K of the SDE (14)

$$K(t_{i+1}) = K(t_i) + \int_{t_i}^{t_{i+1}} \mu^K(t, K(t_i))\mathrm{d}t + B \cdot \Gamma \cdot \Delta U(t_i).$$

that

$$\{K(t_{i+1}, \omega) \mid \omega \in \Omega\} = \mathbb{R}^n \not\subset K^e(t_i) + \mu^K(t_i, K^e(t_i)) \cdot \Delta t_i + D.$$

Thus the weighted Monte-Carlo method will not work if one uses an Euler-Scheme with $\mathrm{rank}(\Gamma) = m < n$, see Figures 1 and 2.

To cope with this problem one may use on of the following methods:

- Use a full rank matrix $\tilde{\Gamma}$ for the primary simulation scheme K°.
- Use a more accurate drift approximation in the primary simulation scheme, such that $\{K^\circ(t_{i+1}, \omega) - K^\circ(t_i, \omega) \mid \omega \in \Omega\} = \mathbb{R}^n$, e.g. the predictor-corrector scheme, see Section 3.1.2.

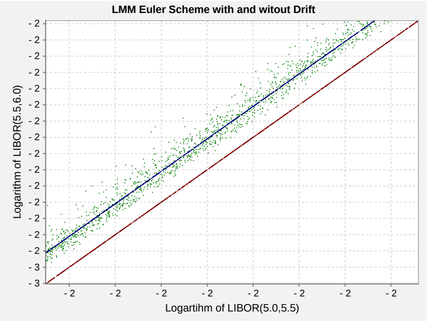


Figure 1: *The space* $\{(L_{10}(t_1, \omega), L_{11}(t_1, \omega)) \mid \omega \in \Omega\}$ *spanned by the two forward rates* L_{10}, L_{11} *at time* $t_1 = 1.0$ *for different simulation schemes using correlation* 1.0. *Shown are Euler scheme without drift (red), Euler scheme with drift and a single time step* ($\Delta t = 1.0$) *(blue) and the (almost) true realizations of the SDE (generated by an Euler scheme with an very fine time step). Obviously a change of measure may not carry over one scheme into another.*

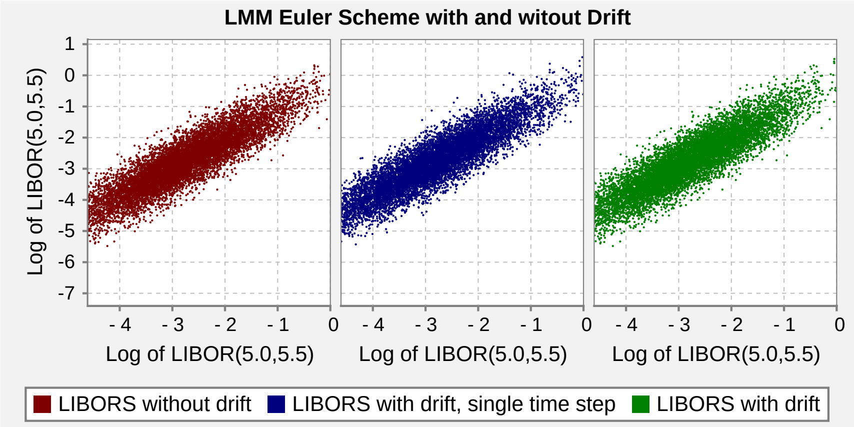


Figure 2: *The space* $\{(L_{10}(t_1, \omega), L_{11}(t_1, \omega)) \mid \omega \in \Omega\}$ *spanned by the two forward rates* L_{10}, L_{11} *at time* $t_1 = 1.0$ *for different simulation schemes and for a slight decorrelation of the processes. The schemes shown are as in Figure 1. A change of measure may carry over one scheme into another.*

5.5 Proxy scheme method for a target scheme with perturbed initial conditions

The method may be applied to reuse the sample-paths of a scheme as realizations of the same scheme with perturbed initial conditions. This will provide us with a full generic and robust methodology for calculating partial derivatives of option prices[10] in a Monte Carlo framework. See Section 8.

We consider three simulation schemes

K° The proxy scheme, i.e. the scheme that generates the realizations.

K^* A target scheme which complies with condition (22).

K^{**} A version of K^* with perturbed initial conditions, i.e. the SDE of K^{**} is that of K^*, but the initial conditions are perturbed.

In general condition (22) will not hold for the pair (K°, K^{**}). The problem is two fold:

First, the initial condition is (assumed to be) non-stochastic and since K° and K^{**} do not have identical initial conditions the transition probability of K° is zero if that of K^{**} is not:

$$\phi(t_0, K_0^{**}; t_1, K_1^{**}) > 0 \Rightarrow \phi(t_0, K_0^\circ; t_1, K_1^\circ) = 0.$$

Thus (22) is violates and we may not correct the realizations of K° by a likelihood ratio such that its distribution matches that of K^{**}.

Second, if the realizations of $K^\circ(t_i)$ and $K^*(t_i)$ for $i > 0$ span a low dimensional subspace of $\mathbb{R}^n$ it might happen that $K^{**}(t_i)$ has realizations outside that subspace. This is essentially the problem discussed in Section (5.4) and it may be avoided as discussed there (e.g. modify K° such that $K^\circ(t_i)$ spans the whole $\mathbb{R}^n$ for $i > 0$. However it should be noted that if a perturbation K^{**} of K^* is such that the span of $K^{**}(t_i)$ is not a subset of the span of $K^*(t_i)$, then this perturbation shouldn't even be considered, since one is measuring the movement with respect to a parameter that is assumed to be constant in the original model.

For the application of calculating option prices under the perturbed scheme K^{**} the first problem may be usually avoided: Since $K^{**}(t_0)$ is non-stochastic the option payoff in general depends only on the future realizations $K^{**}(t_i)$. Thus, considering the pricing formula (15), $K^{**}(t_0)$ enters only in the numéraire, and there in the deterministic part of the discounting. We will thus consider the adjusted scheme $K^{**\circ}$ defined by

$$K^{**\circ}(t_0) := K^\circ(t_0), \qquad K^{**\circ}(t_i) := K^{**}(t_i), \quad i > 0,$$

calculate option prices under the scheme $K^{**\circ}$ (to be precise, under the proxy scheme pair $(K^\circ, K^{**\circ})$) and adjust the price for the false discounting, i.e. by the factor $\frac{N^{**}(t_0)}{N^\circ(t_0)}$, where $N^{**}(t_0)$ is the numéraire calculate from $K^{**}(t_0)$ and $N^\circ(t_0)$ is the numéraire calculate from $K^\circ(t_0)$.

[10] Also known as sensitivities.

6 Benchmark Model and Product Specification

For numerical investigation of the proxy scheme method we will consider the following setting:

Proxy Scheme

As proxy scheme we will use the more than inaccurate *zero drift* scheme

$$K^{\mathrm{zd}}(t_{i+1}) = K^{\mathrm{zd}}(t_i) + \Sigma \cdot \Gamma \cdot (U(t_{i+1}) - U(t_i)),$$

in other word we completely neglect the drift and let the likelihood ratio correct for it - a rather strong test.

LIBOR Market Model

We generate Monte-Carlo path for a LIBOR Market Model for semiannual rates $L(T_i, T_{i+1})$ with $T_{i+j} - T_j = 0.5$. We will simulate 20 rates over 10 years ($t \in [0, 10]$) and use large simulation time steps of $t_{i+1} - t_i = 0.5$.

The instantaneous volatility used is $\sigma = 50\%$, the initial curve is $L_i = 10\%$ and the correlation is exponentially decaying with a rate of 0.8. The rather high volatility will generate larger Monte-Carlo errors.

Benchmark Products

We benchmark drift convergence though the pricing of the 19 Zero-Bonds $P(T_i)$, $i = 1, \ldots, 19$. To benchmark sensitivity convergence we calculate digital caplets (strike 10%) and two one-out-of-three Auto Caps with semiannual periods, the first one with fixings $\{2.0, 3.0, 4.0\}$ and strikes $\{12\%, 10\%, 8\%\}$, the second one with fixings $\{4.0, 6.0, 8.0\}$ and strikes $\{10\%, 10\%, 10\%\}$. The trigger effect of the two Auto Caps is strong enough to challenge the calculation of Monte-Carlo sensitivities. The digital caplets are the prototypical benchmark products for sensitivities in Monte-Carlo due to their payoff being a Heaviside function.

Numerics

Beside this we use rather simple tools and techniques: The random number generator is a Mersenne Twister with Box-Muller transform. We conducted hundreds of Monte-Carlos experiments each with 10,000 to 1,000,000 paths. Calculations were performed on 64 bit processors using 64 bit operating systems[11] hosting a Java 1.5 Virtual Machine.

Our numerical examples illustrate the difference between classical direct simulation and proxy scheme with likelihood ration simulation. Both methods exhibit residual Monte-Carlo errors with are larger than usual due to the slightly exaggerated date used (high volatility, higher decorrelation). Improving the numerics (e.g. random numbers, choice of proxy scheme, variance reduction, etc.) will greatly improve the overall level of convergence.

[11] LinuX and Mac OS X

7 Benchmark application (I): High accuracy drift approximation for the LIBOR Market Model

7.1 Numerical Results

Pricing of Zero Bonds (measuring the drift error)

The price of the zero bond directly measures the drift error (the price is invariant of Σ and Γ). To assess the drift error we compare the Monte-Carlo pricing of zero bonds with analytical prices.

To show that the proxy scheme method actually works we compare direct simulations (non-weighted Monte-Carlo) with proxy scheme simulations with likelihood ratio weighted Monte-Carlo.

As proxy scheme we use the more than inaccurate *zero drift* scheme

$$K^{\text{zd}}(t_{i+1}) = K^{\text{zd}}(t_i) + \Sigma \cdot \Gamma \cdot (U(t_{i+1}) - U(t_i)), \tag{24}$$

in other word we completely neglect the drift and let the likelihood ratio correct for it - a rather strong test.

We calculate Bond prices $P(T_i)$ for $T_i \in \{0.5, 1.0, \ldots, 9.5\}$ and measure the L_1-distance of the Bond price curve to the analytic solution. In other words out error measure is

$$\text{error} = \sum_{i=1}^{19} |P^{\text{simulation}}(T_i) - P^{\text{analytic}}(T_i)|. \tag{25}$$

We perform several Monte-Carlo simulation an calculate the mean and the standard deviation of the error measure (25). These are shown in Table 1. Figure 3 graphically compares the distributions of the error measure of our Monte-Carlo tests.

The results clearly show that not only the proxy scheme method works to generate similar Bond prices that the corresponding direct simulation, it even outperforms the direct simulation.

7.2 Higher accuracy drift approximations: Expansion for the transition probability of the LIBOR Market Model

Since the likelihood ratio-weighted Monte-Carlo method uses only the transition probability ϕ^{K^*} of the secondary scheme K^* we may use high accurate approximations of the transition probability ϕ^K of K to correct the primary simulation scheme K°. The work of [12] gives an expansion of the transition probability for general Itô processes. We apply this to the LIBOR Market Model process in [7].

Direct Simulation			Proxy Scheme Method		
Scheme	Bond Error (in bp)		Scheme	Bond Error (in bp)	
K^{zd}	175.69347	±6.29436			
K^{e}	6.4343	±2.74023	$(K^{\mathrm{zd}}, K^{\mathrm{e}})$	5.78276	±2.56599
K^{pc}	5.68393	±2.2745			
K^{ta}	5.68384	±2.27453	$(K^{\mathrm{zd}}, K^{\mathrm{ta}})$	4.83355	±1.96074

Table 1: *Bond Errors.*

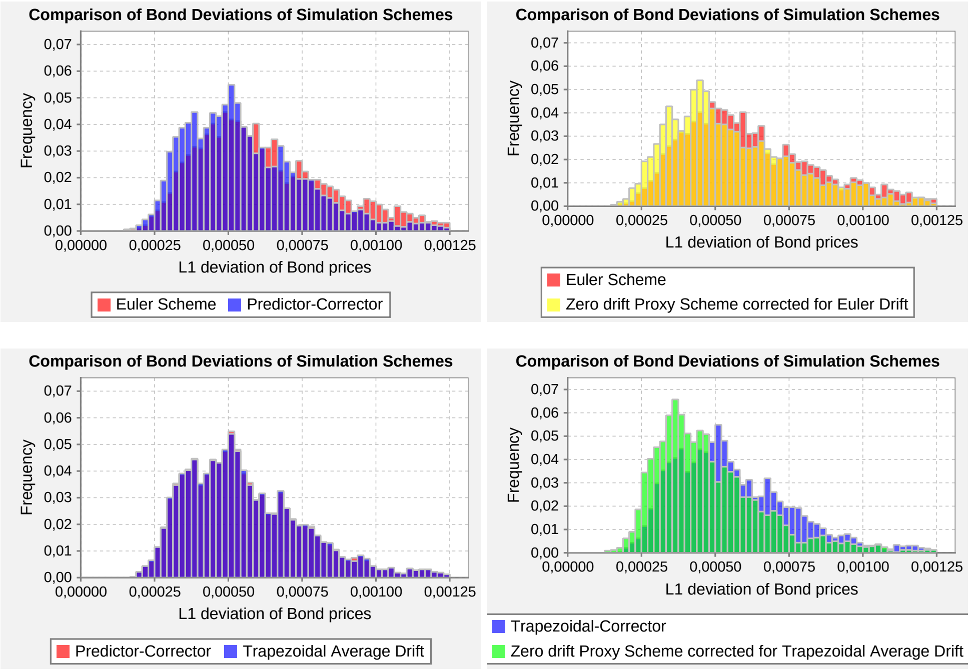


Figure 3: *Distribution of Monte-Carlo bond price errors for direct simulation schemes and proxy simulation scheme with likelihood ratio. A simulation scheme performs better than the other if the distribution is further to the left than the other schemes distribution.*

8 Benchmark application (II): Robust sensitivities (derivatives) of product with discontinuous payouts

8.1 Sensitivities in Monte-Carlo

Trigger type products feature a payout function f which is a discontinuous function of the realizations $L(t_0,\omega), L(t_1,\omega), \ldots, L(t_k,\omega)$ on every fixed sample path ω. Thus (standard) Monte-Carlo approximation of the expectation (26) of f

$$V(t_0) = N(0) \cdot \frac{1}{m} \sum_{i=1}^{m} f(t_k, L(t_0,\omega_i), L(t_1,\omega_i), \ldots, L(t_k,\omega_i)) \tag{26}$$

for sample path $\omega_1, \ldots, \omega_n$ drawn according to (11) is the sum of remains a discontinuous function of the model parameters (as a sum of discontinuous functions). This dramatically reduces the convergence rate of sensitives when calculated as finite differences

$$\frac{V(t_0; X+h) - V(t_0; X)}{h} \tag{27}$$

where X denotes any model parameter, $V(t_0; X)$ the Monte-Carlo price with respect to the simulation scheme with parameter X.

Instead we will consider three simulations scheme: K° (the proxy scheme), $K^{*,X}$ (the target scheme, which may be identical to the proxy scheme in this application) and $K^{*,X+h}$ (the target scheme with perturbed model parameters). We will then calculate the option price under $(K^\circ, K^{*,X})$ and $(K^\circ, K^{*,X+h})$ to calculate the finite difference (27). This corresponds to numerical differentiation of the likelihood ratio, which is a C^∞ function of the model parameter X.

8.2 Numerical Results

Sensitivities of Digital Caplet and Auto Cap

We benchmark the method through the calculation of delta and gamma for digital caplets and auto caps. An digital caplet with fixing date T_i and payment date T_{i+1} pays

$$\left.\begin{array}{ll} 1 & \text{if} \quad L(T_i, T_{i+1}; T_i) > S_i \\ 0 & \text{else} \end{array}\right\} \cdot (T_{i+1} - T_i \qquad \text{at time } T_{i+1}.$$

It is the simplest product with a discontinuous payoff. An auto cap with fixing dates $T_1, ..., T_k$, payment dates $T_2, ..., T_{k+1}$, strikes $S_1, ..., S_k$ and maximum number of exercised $n_{\max}$ pays

$$\max(L(T_i, T_{i+1}; T_i) - S_i \ , \ 0) \cdot (T_{i+1} - T_i) \qquad \text{at time } T_{i+1}$$

for the first $n_{\max}$ times T_i for which expression (8.2) is positive. This payout is a discontinuous function of the model parameters since a slight change in model parameters may suppress a later payment (which might be large) in favor for a small earlier payment (which was non-positive before the change).

Benchmark results

We consider the benchmark products above and calculate price, delta and gamma using finite differences. We compare

- K^e - direct simulation with Euler scheme - with
- (K^{zd}, K^e) - proxy simulation with zero-drift scheme as proxy scheme and Euler scheme as target scheme.

Price convergence

The proxy simulation (K^{zd}, K^e) (using zero drift in the generation of the realizations) shows the same Monte-Carlo convergence for pricing of the two Auto Caps considered as the direct simulation of K^e. The prices generated by the proxy scheme with likelihood ratio are only slightly different from the prices generated by direct simulation - the difference is negligible compared to the Monte-Carlo error. See Figure 4.

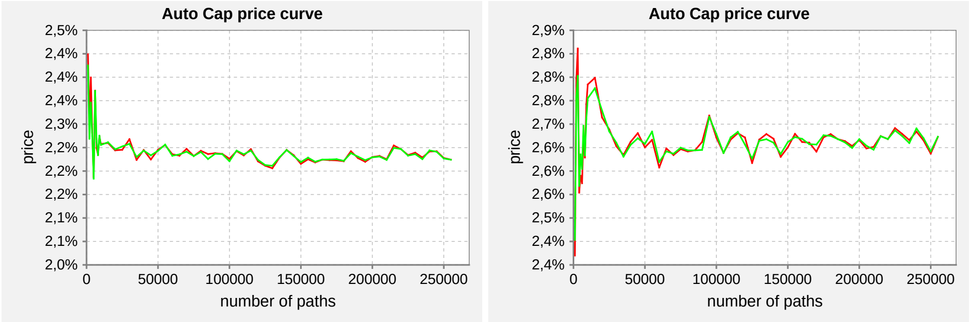


Figure 4: *Convergence of Auto Cap Monte-Carlo price using direct simulation (red) and proxy scheme simulation with likelihood ratio (green).*

Sensitivity convergence and shift dependence

The finite difference calculation of first and second order partial derivatives (delta and gamma) shows a strong dependence on the shift size when the direct simulation scheme is used. The results for proxy scheme simulation are independent on the shift size. For small shifts the proxy scheme sensitivities calculated with a proxy scheme method clearly outperform the direct simulation scheme sensitivities (Table 2, first row). The remaining variance in the proxy scheme sensitivities is the Monte-Carlo error. An increase of the number of paths by a factor of 10 reduces the error by a factor of $\frac{1}{\sqrt{10}} \approx \frac{1}{3}$ (Table 2). For large shifts the sensitivities calculated by shifting the direct simulation stabilize and the Monte-Carlo error of the proxy scheme simulation turns out to be larger by a factor of 2. However large shifts are undesirable, because higher order effects lead to deviation of the mean - theses are likely to be larger than the Monte-Carlo error. This effect is clearly seen for the digital caplet with maturity 0.5, when compared to the analytic value, see Figure 5. For shifts below 0.5% the delta obtained by classical simulation exhibits unacceptable large variances and for shifts above 0.5% the mean of the delta (obtained from repeated Monte-Carlo simulation) exhibits an error larger than one standard deviation. See also Figure 6 and Table 2.

Even for very large number of paths the shifts of a direct simulation scheme may produce unexpected large changes. Figure 6 shows the Monte-Carlo convergence of delta and gamma for six different shift size. The deviations of the direct simulation scheme are rare, but unpredictable and large.

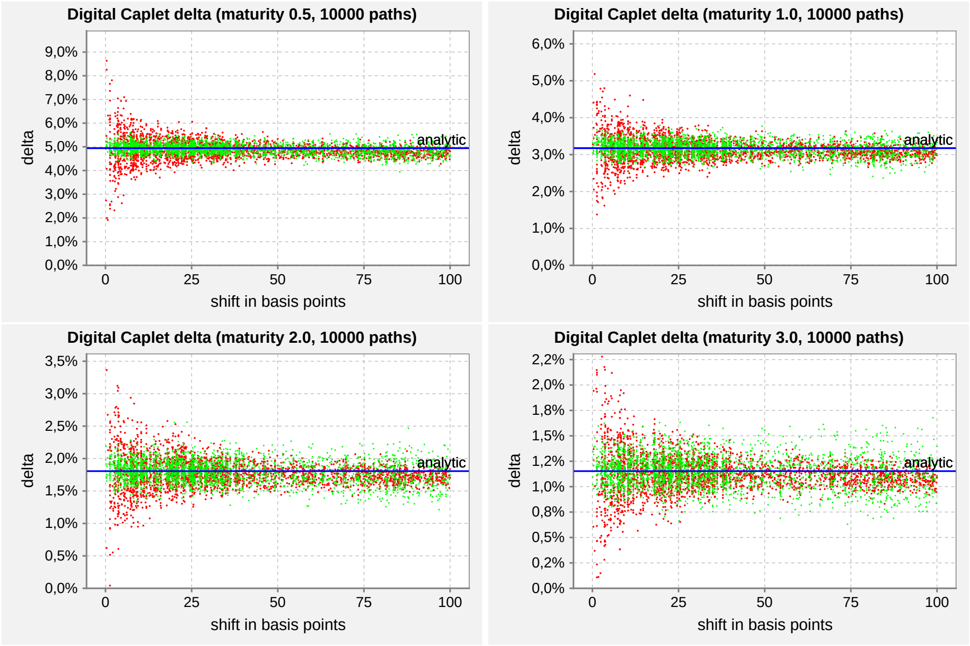
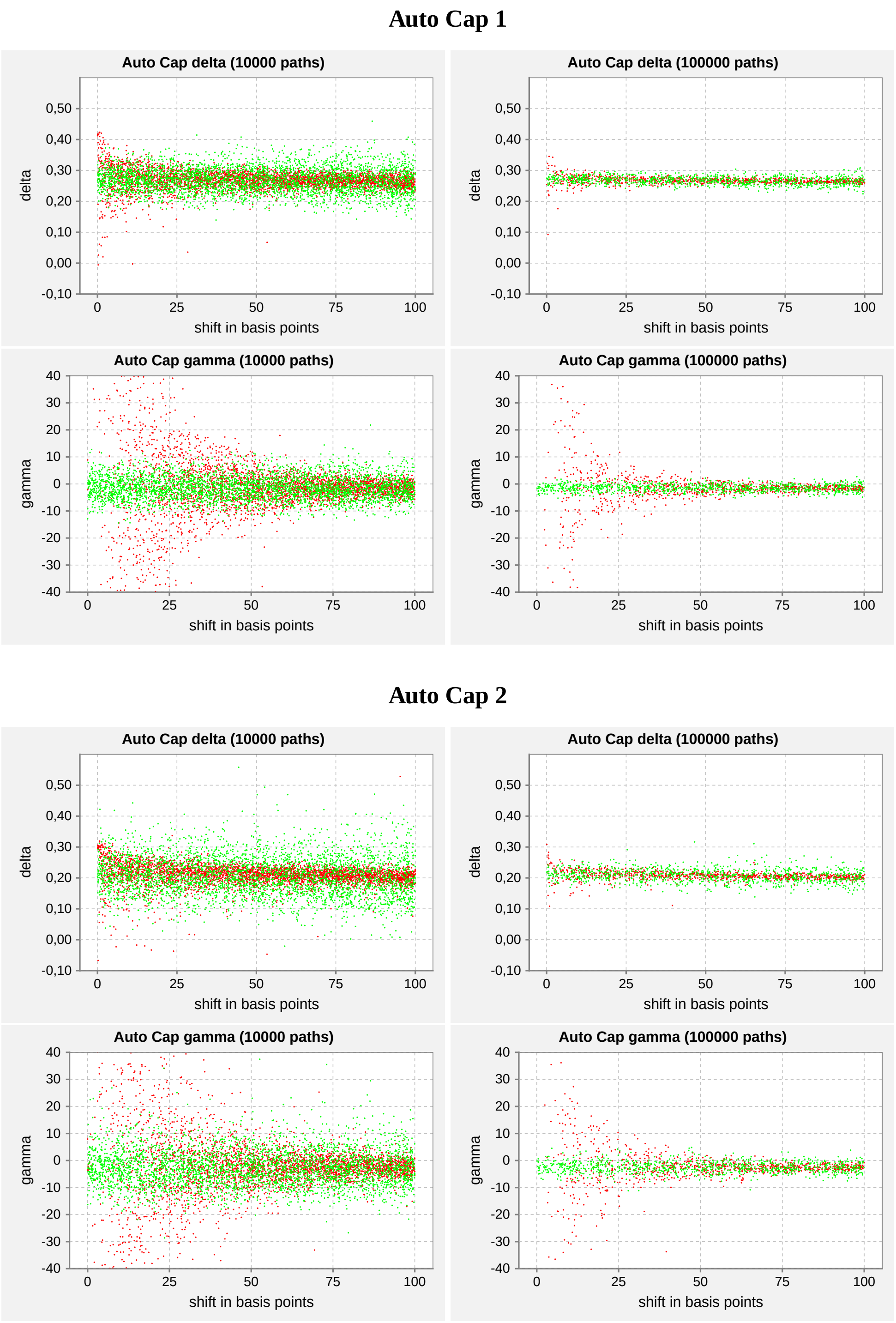


Figure 5: *Dependence of the Digital Caplet delta on the shift size of the finite difference approximation using direct simulation (red) and proxy scheme simulation with likelihood ratio (green). Each dot corresponds to one Monte-Carlo simulation with the stated number of paths.*

Delta and Gamma for Auto Cap 1								
Shift range	Delta (in % per 1% shift)				Gamma (in % per 10% shift)			
(in bp)	Direct Sim		Proxy Scheme		Direct Sim		Proxy Scheme	
10000 Path								
0 − 10	0.271	±0.08	0.272	±0.03	73.8	±3636.6	−1.3	±4.3
10 − 20	0.270	±0.04	0.270	±0.04	−2.9	±43.5	−1.6	±4.4
20 − 30	0.267	±0.03	0.271	±0.04	−1.5	±18.1	−1.3	±4.0
30 − 40	0.270	±0.02	0.268	±0.04	−0.8	±9.3	−1.5	±4.2
40 − 50	0.269	±0.02	0.269	±0.04	−1.2	±6.9	−1.6	±4.3
50 − 60	0.265	±0.02	0.273	±0.04	−2.1	±5.5	−1.3	±4.4
60 − 70	0.267	±0.02	0.267	±0.04	−1.6	±3.7	−1.5	±4.2
70 − 80	0.267	±0.02	0.269	±0.04	−1.5	±3.0	−1.5	±4.5
80 − 90	0.265	±0.02	0.267	±0.04	−1.4	±2.5	−1.5	±4.6
90 − 100	0.264	±0.02	0.262	±0.05	−1.5	±2.2	−1.8	±4.0
100000 Path								
0 − 10	0.269	±0.03	0.272	±0.01	−383.2	±2786.6	−1.4	±1.3
10 − 20	0.271	±0.01	0.271	±0.01	−1.3	±13.5	−1.6	±1.2
20 − 30	0.269	±0.01	0.271	±0.01	−1.7	±5.8	−1.3	±1.3
30 − 40	0.268	±0.01	0.268	±0.01	−2.0	±3.1	−1.5	±1.2
40 − 50	0.267	±0.01	0.267	±0.01	−1.5	±2.2	−1.5	±1.3
50 − 60	0.268	±0.01	0.268	±0.01	−1.4	±1.7	−1.3	±1.3
60 − 70	0.266	±0.01	0.262	±0.01	−1.6	±1.1	−1.8	±1.3
70 − 80	0.266	±0.01	0.267	±0.01	−1.5	±1.0	−1.4	±1.2
80 − 90	0.264	±0.00	0.263	±0.01	−1.6	±0.8	−1.7	±1.4
90 − 100	0.265	±0.00	0.265	±0.02	−1.5	±0.7	−1.4	±1.5

Delta and Gamma for Auto Cap 2								
Shift range	Delta (in % per 1% shift)				Gamma (in % per 10% shift)			
(in bp)	Direct Sim		Proxy Scheme		Direct Sim		Proxy Scheme	
10000 Path								
0 − 10	0.047	±3.07	0.216	±0.06	−11548.2	±212055.7	−2.3	±6.7
10 − 20	0.210	±0.06	0.213	±0.06	−1.2	±77.0	−2.7	±7.2
20 − 30	0.212	±0.04	0.210	±0.06	−2.1	±19.5	−3.2	±7.0
30 − 40	0.212	±0.03	0.216	±0.06	−1.7	±13.0	−2.7	±6.4
40 − 50	0.207	±0.03	0.207	±0.06	−2.6	±8.7	−3.1	±6.7
50 − 60	0.201	±0.10	0.214	±0.11	−4.1	±16.6	−1.5	±12.9
60 − 70	0.205	±0.02	0.204	±0.07	−2.5	±4.9	−2.9	±7.4
70 − 80	0.204	±0.05	0.199	±0.07	−1.9	±15.3	−3.0	±6.7
80 − 90	0.205	±0.02	0.204	±0.07	−2.2	±3.4	−2.2	±7.1
90 − 100	0.202	±0.03	0.196	±0.08	−2.7	±2.8	−2.8	±6.5
100000 Path								
0 − 10	0.215	±0.03	0.213	±0.02	289.8	±2879.1	−2.4	±2.1
10 − 20	0.213	±0.02	0.213	±0.02	−4.4	±15.0	−2.4	±2.3
20 − 30	0.212	±0.01	0.214	±0.02	−2.9	±7.5	−2.3	±2.6
30 − 40	0.208	±0.01	0.211	±0.02	−2.8	±5.0	−2.9	±2.0
40 − 50	0.208	±0.01	0.207	±0.02	−2.4	±2.6	−2.2	±2.5
50 − 60	0.206	±0.01	0.207	±0.02	−2.6	±2.1	−2.6	±2.0
60 − 70	0.205	±0.01	0.206	±0.03	−2.9	±1.7	−2.4	±2.1
70 − 80	0.204	±0.01	0.205	±0.02	−2.7	±1.2	−2.6	±1.8
80 − 90	0.205	±0.01	0.204	±0.02	−2.5	±1.0	−2.4	±1.9
90 − 100	0.202	±0.01	0.202	±0.02	−2.5	±0.9	−2.7	±2.0

Table 2: *Mean an standard deviation corresponding to the simulations in Figure 6 and 7.*



Auto Cap 1

Auto Cap 2

Figure 6: *Dependence of the Auto Cap Monte-Carlo sensitivity on the shift size of the finite difference approximation using direct simulation (red) and proxy scheme simulation with likelihood ratio (green). Each dot corresponds to one Monte-Carlo simulation with the stated number of paths.*

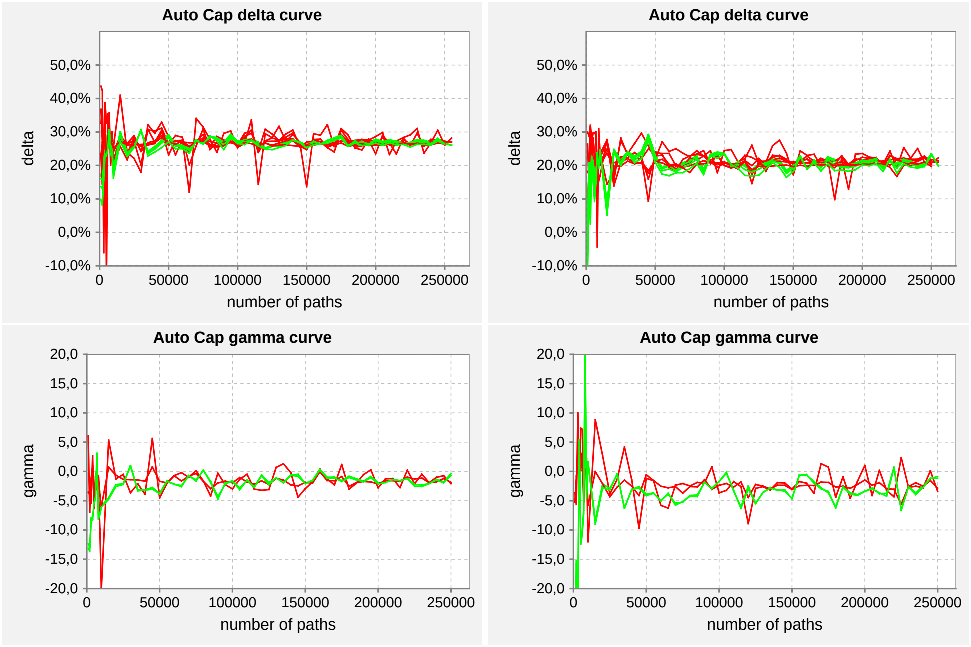


Figure 7: *Convergence of Auto Cap Monte-Carlo delta (upper row) and gamma (lower row) using direct simulation (red) and proxy scheme simulation with likelihood ratio (green). To calculate delta we use finite differences with different shifts h of the initial interest rates, $h \in \{0.0001, 0.0005, 0.0010, 0.0020, 0.0050, 0.0100\}$. Shown are six red and six green curves. Using direct simulation the shift size h leads to different results even for high number of paths. Using proxy scheme simulation the shifts matters only though higher order effects in the likelihood ratios. The graphs on the left correspond to Auto Cap 1, the graphs on the right to Auto Cap 2.*

9 Conclusion

Using Monte-Carlo simulation for the calculation of prices (expectations) and Greeks (derivatives of expectations w.r.t. model parameters) involves three sources of errors:

- The time discretisation of the time-continuous SDE introduces the *time discretiation error*.
- The approximation of the expectation operator by a finite sum introduces the *Monte-Carlo* error.
- The calculation of derivatives of expectations of non-smooth functions (payoffs) by *finite differences* heavily depends on the shift size and the Monte-Carlo error blows up as the shift size tends to zero. Thus finite differences either introduce errors stemming from higher order effects (for large shifts) or are unreliable with extremly large Monte-Carlo errors (for small shifts).

The use of a likelihood ratio weighted proxy scheme simulation enables us to *reduce the time discretiation error* and *make finite differences reliable and independent of the shift size*. The remaining error for pricing and Greeks is a mostly modest Monte-Carlo error. Since the Monte-Carlo error may be measured and thus controlled quite accurately by

$$\frac{1}{n}\sum_{i=1}^{n}(f(K^\circ(T,\omega_i)) - m)^2 \cdot w_i, \quad \text{where} \quad m := \frac{1}{n}\sum_{i=1}^{n} f(K^\circ(T,\omega_i)) \cdot w_i,$$

where

$$w_i := \frac{\phi^{K^*}(K^\circ(T,\omega_i))}{\phi^{K^\circ}(K^\circ(T,\omega_i))} \text{ for pricing} \qquad \text{and} \qquad w_i := \frac{\frac{\partial\phi^{K^*}}{\partial X}(K^\circ)}{\phi^{K^\circ}(K^\circ)} \text{for Greeks.}$$

(The implementation replaces the partial derivative by finite differences).

The use of the proxy scheme may increase the Monte-Carlo error a bit if the likelihood ratios are large. In general it is possible to choose the proxy scheme such that

$$\max\left\{\frac{\phi^*}{\phi^\circ}, \frac{\phi^\circ}{\phi^*}\right\}$$

is small (i.e. close to 1).

The use of a proxy scheme allows to apply finite differences directly to the Monte-Carlo implementation (*bumping the model*) while retaining the robustness and accuracy a likelihood ratio method. The likelihood ratios are calculated numerically from the transition probability of the numerical scheme (which is in general available in closed form). In this sense the method is model independent.

A List of Symbols

Symbol	Meaning
$\|\| \cdot \|\|_{C^0}$	Sup norm. $\|\|f(x)\|\|_{C^0} = \sup_x \|f(x)\|$
$\|\| \cdot \|\|_{C^1}$	C^1-norm. $\|\|f(x)\|\|_{C^1} = \|\|f(x)\|\|_{C^0} + \|\|\frac{\partial}{\partial x} f(x)\|\|_{C^0}$
$\mathrm{E}^{\mathbb{Q}}$	Expectation operator with respect to the measure $\mathbb{Q}$.
$T_j, i = 0, 1, 2, \ldots$	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.
$t_i, i, 0, 1, 2, \ldots$	Discretisation of simulation time.
K	Logarithm of L.
$K^{\circ}(t_i)$	Realizations of the primary simulation scheme (the proxy scheme). $K^{\circ}(t_i)$ is an approximation to $K(t_i)$.
$K^{*}(t_i)$	Realizations of the secondary simulation scheme (the target scheme). $K^{*}(t_i)$ may be either an improved approximation of $K(t_i)$ or an perturbation of $K^{\circ}(t_i)$.
$\phi^{K^{\circ}}$	Transition probability density of K°.
$\phi^{K^{*}}$	Transition probability density of K^{*}. $\frac{\phi^{K^*}}{\phi^{K^{\circ}}}$ is the Monte-Carlo weight used to correct the scheme K° towards K^{*}.
K^{e}	The Euler scheme.
K^{zd}	The zero drift scheme (Euler scheme with neglected drift).
K^{pc}	The predictor-corrector scheme.
K^{ta}	The trapezoidal average drift scheme.

References

[1] AVELLANEDA, MARCO; GAMBA, ROBERTA: Conquering the Greeks in Monte Carlo: Efficient Calculation of the Market Sensitivities and Hedge-Ratios of Financial Assets by Direct Numerical Simulation. 2002.

[2] BENHAMOU, ERIC: Optimal Malliavin Weighting Function for the Computation of the Greeks. 2001.

[3] BINGHAM, NICHOLAS H.; KIESEL, RÜDIGER: Risk-neutral valuation: pricing and hedging of financial derivatives. (Springer Finance). Springer Verlag, London, 1998. ISBN 1-852-33458-4.

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

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

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

[7] FRIES, CHRISTIAN P.; KAMPEN, JÖRG: Fast converging quadratic expansions of log-densities applied to the LIBOR Market Model. (2005).

[8] FOURNIÉ, ERIC; LASRY JEAN-MICHEL; LEBUCHOUX, JÉRÔME; LIONS, PIERRE-LOUIS; TOUZI, NIZAR: Applications of Malliavin calculus to Monte Carlo methods in finance. Finance Stochastics. 3, 391-412 (1999). Springer- Verlag 1999.

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

[10] GLASSERMAN, PAUL; ZHAO, XIAOLING: Fast greeks in forward LIBOR models. Journal of Computational Finance. 3, 5-39 (1999).

[11] HUNTER, CHRISTOPHER J.; JÄCKEL, PETER; JOSHI, MARK S.: Drift Approximations in a Forward-Rate-Based LIBOR Market Model. *Getting the Drift*. Risk 14, 81-84 (July, 2001).

[12] KAMPEN, JÖRG.: On the fundamental solution of parabolic equations. Preprint, 2005.

[13] 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.

[14] MALLIAVIN, PAUL: Stochastic Analysis (Grundlehren Der Mathematischen Wissenschaften). Springer Verlag, 1997. ISBN 3-540-57024-1 .

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

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

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

[18] VARADHAN, S. R. SRINIVASA: On the behavior of the fundamental solution of the heat equation with variable coefficients. Comm. Pure Appl. Math. 20, 431-455. (1967).