Bumping the Model

Generic robust Monte-Carlo Sensitivities using the Proxy Simulation Scheme Method

Christian P. Fries
email@christian-fries.de

December 29, 2005

Abstract

In this article we give a short overview on sensitivity calculation using Monte-Carlo simulation and an introduction to the proxy simulation scheme method. We shortly discuss the localization technique and the implementation.

Contents

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

1 Introduction

Assuming the existence of a (possibly 'black box') pricing algorithm the simplest way to calculate an approximation of partial derivative with respect to a model parameter (aka the 'sensitivity') is to use reevaluation on finite differences (aka *bumping the model*). Doing so one only needs to apply the pricing for two (or more) different sets of model parameters (e.g. initial values).

This approach is not only popular because of its simplicity. It is also the only way to obtain *generic sensitivities*, i.e. partial derivatives with respect to arbitrary changes in the input data. Such scenarios are common: What a trader sees as being the delta of an option is the price change resulting from a shift of the underlyings spot value followed by a re-calibration to a volatility surface that might have undergone a joint movement. So the delta here is a rather complex change in input data. A total derivative rather than a partial derivative.

1.1 Pricing using Monte Carlo Simulation

In this article we consider a pricing algorithm based on Monte Carlo simulation. To fix notation let

$$\mathrm{d}X \;=\; \mu \mathrm{d}t + \sigma \cdot \mathrm{d}W(t)$$

denote the sde of the model primitives[1]. Let $X^*(T_i)$ denote approximation of $X(T_i)$ generated by some (time-)discretization scheme, e.g. an Euler scheme

$$X^*(T_{i+1}) \;=\; X^*(T_i) + \mu(T_i)\Delta T_i + \sigma(T_i) \cdot \Delta W(T_i)$$

or one of the more advanced schemes[2]. We assume that the risk neutral pricing of a financial product may be expressed as the expectation (with respect to the pricing measure) of a function f of some realizations $Y := (X(T_0), X(T_1), \ldots, X(T_m))$ or is approximated in as such through the realizations of the numerical scheme.

$$\mathrm{E}(f(Y)|\mathcal{F}_{T_0}) \;\approx\; \mathrm{E}(f(Y^*)|\mathcal{F}_{T_0}) \;=\; \mathrm{E}(f((X^*(T_0), X^*(T_1), \ldots, X^*(T_m)))|\mathcal{F}_{T_0}),$$

. Here f denotes the Numéraire relative payoff function. The Monte-Carlo pricing consists of the averaging over (in general) equidistributed sample path ω_i, $i = 1, \ldots, n$

$$\mathrm{E}(f(Y^*)|\mathcal{F}_{T_0}) \;\approx\; \hat{\mathrm{E}}(f(Y^*)|\mathcal{F}_{T_0}) \;:=\; \frac{1}{n}\sum_{i=1}^{n} f(Y^*(\omega_i)).$$

1.2 Sensitivities from Monte Carlo Pricing

The calculation of sensitivities using finite differences on a Monte-Carlo based pricing algorithm is known to exhibit instabilities if the payoff function is not smooth enough, e.g. if the payoff exhibits discontinuities as for a digital option. Assume that θ denote some model parameter[3] or a parametrization of a generic market data movement and let Y_θ denote the model realizations depending on that parameter. Let us further assume that ϕ_{Y_θ} denote the probability density of Y_θ. Then the analytic calculation of the sensitivity is given by

$$\frac{\partial}{\partial\theta}\mathrm{E}(f(Y_\theta)|\mathcal{F}_{T_0}) = \frac{\partial}{\partial\theta}\int_{\mathbb{R}^m} f(y)\phi_{Y_\theta}(y)\mathrm{d}y.$$

[1] As model primitives we view the underlyings like financial products (stocks) or rates (forward rates, swap rates, fx rates).
[2] For alternative schemes see e.g. [4, 9]
[3] So for *delta* θ is an initial value $X(0)$, for *vega* θ denotes a volatility, etc.

While the payoff f may be discontinuous the density in general is a smooth function of θ. In that case the expectation $\mathrm{E}(f(Y_\theta)|\mathcal{F}_{T_0})$ (the price) is a smooth function of θ too. The price inherits the smoothness of ϕ_{Y_θ}.

The difficulties aries when we consider the Monte Carlo approximation. It inherits the regularity of the payoff f not that of the density ϕ:

$$\hat{\mathrm{E}}(f(Y_\theta)|\mathcal{F}_{T_0}) \;=\; \frac{1}{n}\sum_{i=1}^{n} f(Y_\theta(\omega_i)).$$

So while $\mathrm{E}(f(Y_\theta)|\mathcal{F}_{T_0})$ may be smooth in θ the Monte Carlo approximation $\hat{\mathrm{E}}(f(Y_\theta)|\mathcal{F}_{T_0})$ may have discontinuities. Is that case a finite difference approximation of the derivative applied to the Monte Carlo pricing will perform poorly.

1.3 Generic Sensitivities

The finite difference approximation calculates the sensitivity by

$$\frac{\partial}{\partial\theta}\mathrm{E}(f(Y_\theta)|\mathcal{F}_{T_0}) \approx \frac{\mathrm{E}(f(Y_{\theta+h})|\mathcal{F}_{T_0}) - \mathrm{E}(f(Y_{\theta-h})|\mathcal{F}_{T_0})}{2h}.$$

This *brute force finite difference* calculation of sensitivities is sometimes referred to as *bumping the model*. Bumping the model has an charming advantage: If you keep your model and your pricing code separated (a design pattern one should always consider) then you may implement a generic code for generating sensitivities by feeding the pricing code with differently bumped models. In other words:

Once the pricing code is written, all sensitivities are available. (1)

It seems as if you get sensitivities almost for free (i.e. without any effort in modeling and implementation) and the only price you pay is a doubling of calculation time compared to pricing. However it is known that applying such a finite difference approximation to a Monte-Carlo implementation will often result in extremely large Monte-Carlo errors. Especially if the payout function of the derivative is discontinuous this Monte-Carlo error even tends to infinity as h tends to zero. And discontinuous payout are present whenever a trigger feature is present.

In general sensitivities in Monte-Carlo is known as a challenge and among the many papers addressing this problem we like to mention the Likelihood Ratio [3] and Malliavin calculus [6, 10].

It appears as if the measures you have to take to improve Monte-Carlo sensitivities will lose the advantage (1) of *bumping the model*. We will present a method (which is also an implementation design pattern) that gives you the possibility to calculate sensitivities through *bumping the model* while providing the accuracy and robustness achieved by the likelihood ratio or Malliavin calculus approach. The method is essentially a likelihood ratio reconsidered on the level of the numerical scheme.

2 Sensitivities in Monte Carlo: Overview

Numerous methods have been proposed to handle sensitivities in Monte Carlo, among them the application of Malliavin Calculus which has drawn more attention recently, [6]. These methods improve the robustness of sensitivities, but require more information. Most notably all these methods loose the potential to calculate generic sensitivities. We start by reviewing the standard (pathwise) finite difference approximation.

2.1 Sensitivities by Finite Differences

The finite difference approximation is given by

$$\begin{aligned}\frac{\partial}{\partial\theta}\mathrm{E}^{\mathbb{Q}}(f(Y_\theta)\mid\mathcal{F}_{T_0}) &\approx \frac{\partial}{\partial\theta}\hat{\mathrm{E}}^{\mathbb{Q}}(f(Y_\theta)\mid\mathcal{F}_{T_0}) \\ &\approx \frac{1}{2h}\big(\hat{\mathrm{E}}^{\mathbb{Q}}(f(Y_{\theta+h})\mid\mathcal{F}_{T_0}) - \hat{\mathrm{E}}^{\mathbb{Q}}(f(Y_{\theta-h})\mid\mathcal{F}_{T_0})\big) \\ &= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2h}\big(f(Y_{\theta+h}(\omega_i) - f(Y_{\theta-h}(\omega_i))\big)\end{aligned}$$

Requirements

- No additional information from the model sde X
- No additional information from the simulation scheme $X^*(T_{i+1})$
- No additional information from the payout f
- No additional information on the nature of θ ($\Rightarrow$ generic sensitivities)

Properties

- Biased derivative for *large* h due to finite difference of order h
- Extremely large variance for discontinuous payouts and *small* h (order h^{-1})

Notes

The most important feature of finite differences is there genericity. Once the the pricing code has been written all kinds of sensitivities may be calculated. For discontinuous payouts finite differences perform poorly. The contribution of a discontinuity to the sensitivity may be calculated analytically. It is the jump size multiplied by the probability density at the discontinuity. Finite differences resolve this contribution only through those sample paths which fall at a neighborhood of the shift size around the discontinuity. Thus, if the shift size is small the discontinuity is resolved by few points, ultimately resulting in a large Monte-Carlo error. If the shift size is large the derivative becomes biased by second order effects (if present).

2.2 Sensitivities by Pathwise Differentiation

The pathwise differentiation method is given by

$$
\begin{aligned}
\frac{\partial}{\partial \theta} \mathrm{E}^{\mathbb{Q}}(f(Y(\theta)) \mid \mathcal{F}_{T_0}) &= \frac{\partial}{\partial \theta} \int_{\Omega} f(Y(\omega,\theta)) \, \mathrm{d}\mathbb{Q}(\omega) = \int_{\Omega} \frac{\partial}{\partial \theta} f(Y(\omega,\theta)) \, \mathrm{d}\mathbb{Q}(\omega) \\
&= \int_{\Omega} f'(Y(\omega,\theta)) \cdot \frac{\partial Y(\omega,\theta)}{\partial \theta} \, \mathrm{d}\mathbb{Q}(\omega) = \mathrm{E}^{\mathbb{Q}}(f'(Y(\theta)) \cdot \frac{\partial Y(\theta)}{\partial \theta} \mid \mathcal{F}_{T_0}) \\
&\approx \hat{\mathrm{E}}^{\mathbb{Q}}(f'(Y(\theta)) \cdot \frac{\partial Y(\theta)}{\partial \theta} \mid \mathcal{F}_{T_0}) = \frac{1}{n} \sum_{i=1}^{n} f'(Y(\omega_i,\theta)) \cdot \frac{\partial Y(\omega_i,\theta)}{\partial \theta}
\end{aligned}
$$

Requirements

- Additional information on the model sde X
- No additional information on the simulation scheme $X(T_{i+1})$
- Additional information on the payout f (derivative of f must be known)
- Additional information on the nature of θ ($\Rightarrow$ no generic sensitivities)

Properties

- Unbiased derivative
- Discontinuous payouts may be handled (interpret f' as distribution)

Notes

The pathwise method requires the knowledge of the payouts derivative and the derivative of the process realizations with respect to the parameter θ. It is thus only applicable for a restricted class of model parameters. In addition it seems as if a discontinuity in the payout may not be handled, but as noted before the impact of a jump may be calculated analytically. See Joshi & Kainth [8] or Rott & Fries [11] for an example on how use pathwise differentiation with discontinuous payouts (there in the context of Default Swaps, CDOs).

Version 0.5 (20051227)
Comments welcome.

2.3 Sensitivities by Likelihood Ratio Weighting

The likelihood ratio weighting [2, 3, 7] is given by

$$\frac{\partial}{\partial\theta}\mathrm{E}^{\mathbb{Q}}(f(Y(\theta)) \mid \mathcal{F}_{T_0}) = \frac{\partial}{\partial\theta}\int_{\Omega} f(Y(\omega,\theta))\,\mathrm{d}\mathbb{Q}(\omega) = \frac{\partial}{\partial\theta}\int_{\mathbb{R}^m} f(y)\cdot\phi_{Y(\theta)}(y)\,\mathrm{d}y$$

$$= \int_{\mathbb{R}^m} f(y)\cdot\frac{\frac{\partial}{\partial\theta}\phi_{Y(\theta)}(y)}{\phi_{Y(\theta)}(y)}\cdot\phi_{Y(\theta)}(y)\,\mathrm{d}y = \mathrm{E}^{\mathbb{Q}}(f(Y)\cdot w(\theta) \mid \mathcal{F}_{T_0})$$

$$\approx \hat{\mathrm{E}}^{\mathbb{Q}}(f(Y)\cdot w(\theta) \mid \mathcal{F}_{T_0}) = \frac{1}{n}\sum_{i=1}^{n} f(Y(\omega_i))\cdot w(\theta,\omega_i)$$

Requirements

- Additional information on the model sde X ($\rightarrow \phi_{Y(\theta)}$)
- No additional information on the simulation scheme $X(T_{i+1})$
- No additional information on the payout f
- Additional information on the nature of θ ($\Rightarrow$ no generic sensitivities)

Properties

- Unbiased derivative
- Discontinuous payouts may be handled.

Notes

The Likelihood Ratio method requires no additional information on the payout function. However it requires that the density of the model sde's realizations $X(t)$ is known and furthermore that its derivative with respect to the parameter θ is known. This is rarely the case and this is viewed as being the major drawback of the method.

2.4 Sensitivities by Malliavin Weighting

Th Malliavin weighting is similar to the Likelihood Ratio method: the sensitivity is expressed as the expectation of a weighted payout function.

$$\frac{\partial}{\partial\theta}\mathrm{E}^{\mathbb{Q}}(f(Y(\theta)) \mid \mathcal{F}_{T_0}) = \mathrm{E}^{\mathbb{Q}}(f(Y(\theta)) \cdot w(\theta) \mid \mathcal{F}_{T_0})$$

$$\approx \hat{\mathrm{E}}^{\mathbb{Q}}(f(Y(\theta)) \cdot w(\theta) \mid \mathcal{F}_{T_0}) = \frac{1}{n}\sum_{i=1}^{n} f(Y(\theta,\omega_i)) \cdot w(\theta,\omega_i)$$

Requirements

- Additional information on the model sde X ($\rightarrow w$)
- No additional information on the simulation scheme $X(T_{i+1})$
- No additional information on the payout f
- Additional information on the nature of θ ($\Rightarrow$ no generic sensitivities)

Properties

- Unbiased derivative
- Discontinuous payouts may be handled.

Notes

Benhamou [1] showed that the Likelihood Ratio corresponds to the Malliavin weights with minimal variance and may be expressed as a conditional expectation of all corresponding Malliavin weights (we thus view the Likelihood Ratio as an example for the Malliavin weighting method).

However here the weights are derived directly through Malliavin calculus which makes this method more general and applicable even if the density is not known. The derivation of the Malliavin weights requires in depth knowledge of the underlying continuous process X and heavily depends on the nature of θ.

3 Proxy Simulation Scheme

We will propose a design for a Monte Carlo pricing engine that has the remarkable properties that the application of finite differences to the pricing will give us in Likelihood Ratio weighted sensitivities without actually the need to know the density ϕ analytically. Thus it combines the robustness of Likelihood Ratio or Malliavin weighting with the gernericity of finite differences. Consider *two* time discrete schemes for the stochastic process X:

X^*	$T_i \mapsto X^*(T_i)$	$i = 0, 1, 2, \ldots$	time discretization scheme of X $\longrightarrow$ *target scheme*
X°	$T_i \mapsto X^\circ(T_i)$	$i = 0, 1, 2, \ldots$	any other time discrete stochastic process (assumed to be *close* to X^*) $\longrightarrow$ *proxy scheme*

Let $\phi_{Y^\circ}(y)$ denote the $density$ of Y° and $\phi_{Y^*}(y)$ the $density$ of Y^*. We require

$$\forall y : \phi^{Y^\circ}(y) \ = \ 0 \ \Rightarrow \ \phi^{Y^*}(y) \ = \ 0. \tag{2}$$

Using the additional scheme X° the pricing of a payout function f is now performed in the following way: Let $Y = (X(T_1), \ldots, X(T_m))$, $Y^* = (X^*(T_1), \ldots, X^*(T_m))$, $Y^\circ = (X^\circ(T_1), \ldots, X^\circ(T_m))$. We have $\mathrm{E}^{\mathbb{Q}}(f(Y(\theta)) \mid \mathcal{F}_{T_0}) \approx \mathrm{E}^{\mathbb{Q}}(f(Y^*(\theta)) \mid \mathcal{F}_{T_0})$ and furthermore

$$\begin{aligned}
\mathrm{E}^{\mathbb{Q}}(f(Y^*(\theta)) \mid \mathcal{F}_{T_0}) \ &= \ \int_\Omega f(Y^*(\omega, \theta)) \,\mathrm{d}\mathbb{Q}(\omega) \ = \ \int_{\mathbb{R}^m} f(y) \cdot \phi_{Y^*(\theta)}(y) \,\mathrm{d}y \\
&= \ \int_{\mathbb{R}^m} f(y) \cdot \frac{\phi_{Y^*(\theta)}(y)}{\phi_{Y^\circ}(y)} \cdot \phi_{Y^\circ}(y) \,\mathrm{d}y \ = \ \mathrm{E}^{\mathbb{Q}}(f(Y^\circ) \cdot w(\theta) \mid \mathcal{F}_{T_0}),
\end{aligned}$$

where $w(\theta) = \frac{\phi_{Y^*(\theta)}(y)}{\phi_{Y^\circ}(y)}$.

For the Monte-Carlo approximation this implies that the sample paths are generated from the scheme X° while the probability densities are corrected towards the target scheme X^*.

Notes

- For $X^\circ = X^*$ we have $w(\theta) = 1$ and in this case the proxy simulation scheme corresponds to the ordinary Monte Carlo simulation of X^*.
- The proxy scheme X° and thus its realization vector Y° is seen as being independent of θ. This has important implications on the calculation of sensitivities, see Section 3.2.
- The requirement $\forall y : \phi^{Y^\circ}(y) \ = \ 0 \ \Rightarrow \ \phi^{Y^*}(y) \ = \ 0$ corresponds to the non-degeneracy condition of the diffusion matrix as it appears in the application of the likelihood ratio and Malliavin weights. However, here this requirement is by far less restrictive since we are fee to choose the proxy scheme X°.

3.1 Calculation of Monte-Carlo weights

For the most common numerical scheme the densities ϕ^{Y°, ϕ^{Y^*} and thus the Monte-Carlo weights may be calculated numerically. Consider for example the schemes

target scheme: $X^*(T_{i+1}) \ = \ X^*(T_i) \ + \ \mu^{X^*}(T_i)\Delta T_i \ + \ \Sigma(T_i) \cdot \Gamma(T_i) \cdot \Delta U(T_i)$

proxy scheme: $X^\circ(T_{i+1}) \ = \ X^\circ(T_i) \ + \ \mu^{X^\circ}(T_i)\Delta T_i \ + \ \Sigma^\circ(T_i) \cdot \Gamma^\circ(T_i) \cdot \Delta U(T_i)$

where Σ denotes an invertible volatility matrix and Γ denotes a projection matrix, the factor matrix which defines the correlation structure $R = \Gamma\Gamma^{\mathsf{T}}$.

Assume for simplicity that $\mu^{X^*}(T_i)$ depends on $X^*(T_i)$, $X^*(T_{i+1})$ only (and similar for $\mu^{X^\circ}(T_i)$) (this holds for, e.g. Euler Scheme, Predictor Corrector), then we have for the transition probability densities

$$\phi^{X^*}(T_i, X^*_i; T_{i+1}, X^*_{i+1}) = \frac{1}{(2\Pi\Delta T_i)^{n/2}} \exp\Big(- \frac{1}{2\Delta T_i} \left(\Lambda^{-1/2} F^{\mathrm{T}} \Sigma^{-1} \left(X^*_{i+1} - X^*_i - \mu^{X^*}(T_i)\Delta T_i\right)\right)^2 \Big)$$

$$\phi^{X^\circ}(T_i, X^\circ_i; T_{i+1}, X^\circ_{i+1}) = \frac{1}{(2\Pi\Delta T_i)^{n/2}} \exp\Big(- \frac{1}{2\Delta T_i} \left(\Lambda^{\circ-1/2} F^{\circ\mathrm{T}} \Sigma^{\circ-1} \left(X^\circ_{i+1} - X^\circ_i - \mu^{X^\circ}(T_i)\Delta T_i\right)\right)^2 \Big).$$

And the proxy scheme weights are given by

$$w(T_{i+1})\,|_{\mathcal{F}_{T_k}} = \prod_{j=k}^{i} \frac{\phi^{X^*}(T_j, X^\circ_j; T_{j+1}, X^\circ_{j+1})}{\phi^{X^\circ}(T_j, X^\circ_j; T_{j+1}, X^\circ_{j+1})},$$

see [4] for details.

Notes

- We used the factor decomposition (PCA) $\Gamma = F \cdot \sqrt{\Lambda}$ where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_m)$ are the non-zero Eigenvalues of $\Gamma \cdot \Gamma^{\mathsf{T}}$.
- A change of market data / calibration enters into transition probabilities only.

3.2 Sensitivities by Finite Differences on a Proxy Simulation Scheme

$$\begin{aligned}\frac{\partial}{\partial\theta}\mathrm{E}^{\mathbb{Q}}(f(Y^{*}(\theta))\,|\,\mathcal{F}_{T_0}) &\approx \frac{1}{2h}\left(\mathrm{E}^{\mathbb{Q}}(f(Y^{*}(\theta+h))\,|\,\mathcal{F}_{T_0}) - \mathrm{E}^{\mathbb{Q}}(f(Y^{*}(\theta-h))\,|\,\mathcal{F}_{T_0})\right)\\ &= \frac{\partial}{\partial\theta}\int_{\mathbb{R}^m} f(y)\cdot\frac{1}{2h}(\phi_{Y^{*}(\theta+h)}(y)-\phi_{Y^{*}(\theta-h)}(y))\,\mathrm{d}y\\ &= \int_{\mathbb{R}^m} f(y)\cdot\frac{\frac{1}{2h}(\phi_{Y^{*}(\theta+h)}(y)-\phi_{Y^{*}(\theta-h)}(y))}{\phi_{Y^{\circ}}(y)}\cdot\phi_{Y^{\circ}}(y)\,\mathrm{d}y\\ &\approx \frac{1}{n}\sum_{i=1}^{n} f(Y^{\circ}(\omega_i))\cdot\frac{1}{2h}(w(\theta+h,\omega_i)-w(\theta-h,\omega_i))\end{aligned}$$

Requirements

- No additional information on the model sde X
- Additional information on the simulation scheme $X^{*}(T_{i+1})$, $X^{\circ}(T_{i+1})$
- No additional information on the payout f
- No additional information on the nature of θ ($\Rightarrow$ generic sensitivities)

Properties

- Biased derivative (but small shift h possible!)
- Discontinuous payouts may be handled.

Notes

We noted above that additional information on the simulation scheme is required, which are the densities of the two schemes. Note however that we require these densities to setup the pricing algorithm. For the sensitivity calculation no additional information is needed. Note also that the required densities are densities of numerical schemes which may usually calculated numerical from known transition probability densities (see Section 3.1).

3.3 Localization

If the payout function f is smooth then ordinary finite differences perform better than the weighting techniques. The latter show an increase in Monte-Carlo variance of the sensitivity. This effect is not only visible for smooth payouts f, but also for large finite difference shifts, see Figure 2.

A solution that has been proposed in [6] is localization. Here the weighting is applied only to a region where the payoff is discontinuous.

Let g denote the localization function, i.e. a smooth function $0 \leq g \leq 1$ such that $g = 1$ at discontinuities of f. Consider the decomposition

$$f = (1-g)\cdot f \;+\; g\cdot f.$$

We define the pricing of the payout f as

$$\mathrm{E}(f(Y^{*})|\mathcal{F}_{T_0}) \;=\; \mathrm{E}((1-g(Y^{*}))\cdot f(Y^{*})|\mathcal{F}_{T_0}) \;+\; \mathrm{E}(g(Y^{\circ})\cdot f(Y^{\circ})\cdot\frac{\phi_{Y^{*}}}{\phi_{Y^{\circ}}}|\mathcal{F}_{T_0}).$$

Version 0.5 (20051227)
Comments welcome.

In other words: We use a pricing based on a proxy simulation scheme for $g \cdot f$ and a pricing based on direct simulation for $(1 - g) \cdot f$.

It should be noted that localization is carried out by a redefinition of the payout. The product is split in two parts, where one is priced by a direct simulation scheme and the other is prices by a proxy simulation scheme method. This allows to implement localization on the product level, completely independent of the actual simulation properties. In addition, localization does not reduce the ability to calculate generic sensitivities.

3.4 Object Oriented Design

The proxy scheme simulation method may in part also be viewed as an implementation design. In Figure 1(a) we depict the object oriented design of a standard Monte-Carlo simulation where a change in market data results in a change of simulation path. In Figure 1(b) we contrast the proxy scheme simulation method where a change in market data results in a change of Monte-Carlo weights. In practice we propose that the model driving the proxy schemes paths generation is calibrated to market data used for pricing while a market data scenario used for sensitivity calculation, i.e. bumping the model, impacts the Monte-Carlo weights only. A method should be offered to reset the proxy simulation's market data to the target simulation's market data.

4 Numerical Example

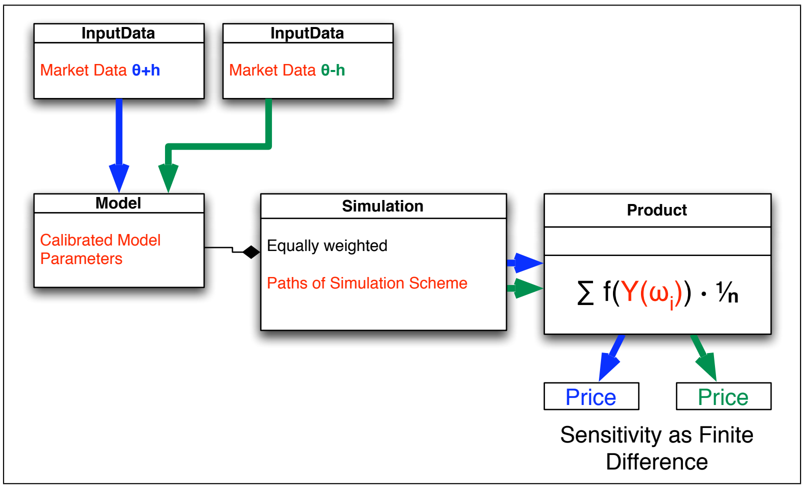
We consider a standard LIBOR Market Model and compare the sensitivity calculation by applying finite differences to a direct simulation of paths generated by an Euler scheme and by applying finite differences to a proxy scheme simulation using two Euler schemes.

We conduct several independent Monte-Carlo calculations for different sizes of the finite difference shift. Figure 2 shows the result for the gamma (second derivative with respect to initial data) of a digital caplet.

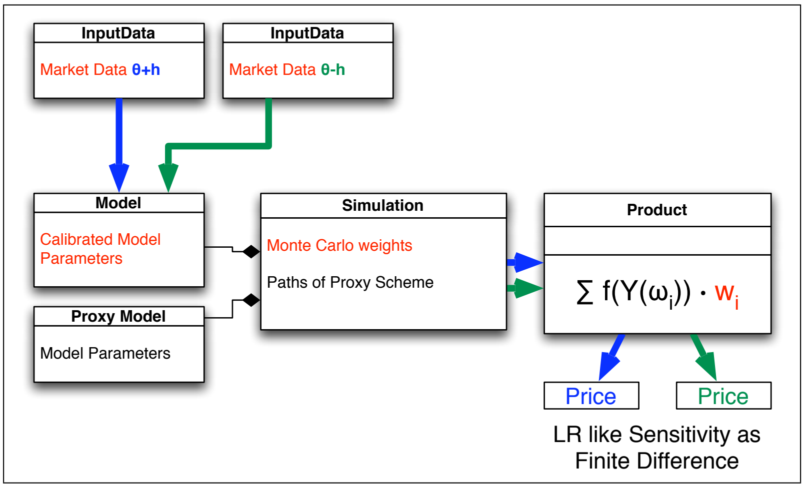
5 Conclusion

On first glance the method proposed looks like a numerical calculation of a Malliavin weight (or Likelihood Ratio). Let us shortly note the key differences of the proxy scheme simulation method to established methods for sensitivity calculations like Likelihood Ratio or Malliavin weighting.

- We define a weighting for the calculation of prices not for the calculation of sensitivities. The weighting is totally independent from sensitivity calculation.
- The weights are defined through probability densities of numerical schemes, not through probability densities of a model sde. This allows to calculate the weights numerically.
- Within some constraints the choice of the proxy scheme is free.
- From the target scheme only the probability density is required. This allows to implement weak schemes that achieve higher accuracy, see [4, 5].
- Once a pricing framework has been implemented, sensitivity calculation similar to the Likelihood Ratio or Malliavin weighting is obtain by applying finite difference to the pricing.



(a) Standard Monte Carlo Simulation



(b) Proxy Scheme Monte Carlo Simulation

Figure 1: *Object Oriented Design of the Monte Carlo Pricing Engine: We depict the impact of a change of different market data scenarios* $\theta + h$ *and* $\theta - h$ *on the pricing code of a standard Monte-Carlo simulation and a proxy scheme simulation.*

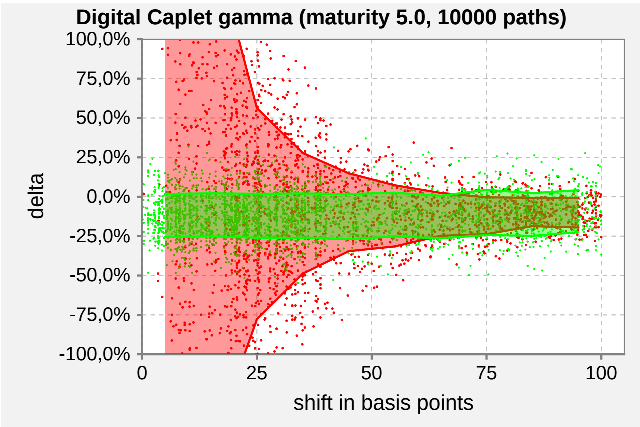


Figure 2: *Dependence of the Digital Caplet gamma on the shift size of the finite difference approximation. Finite difference is applied to a direct simulation (red) and to an proxy scheme simulation (green). Each dot corresponds to one Monte-Carlo simulation with the stated number of paths. The red 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.

References

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

[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.; KAMPEN, JÖRG: Proxy Simulation Schemes for generic robust Monte-Carlo sensitivities and high accuracy drift approximation (with applications to the LIBOR Market Model). (2005). http://www.christian-fries.de/finmath/proxyscheme

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

[6] 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.

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

[8] JOSHI, MARK S.; KAINTH, DHERMINDER: Rapid computation of prices and deltas of n^{th} to default swaps in the Li Model. *Quantitative Finance*, volume 4, issue 3, (June 04), p. 266- 275.
http://www.quarchome.org/.

[9] 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.

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

[11] ROTT, MARIUS G.; FRIES, CHRISTIAN P.: Fast and Robust Monte Carlo CDO Sensitivities and their Efficient Object Oriented Implementation. 2005.
http://www.christian-fries.de/finmath/cdogreeks