Monte-Carlo Payoff-Smoothing for Pricing Autocallable Instruments

Frank Koster[*], Achim Rehmet, DekaBank

Abstract

In this paper we develop a Monte-Carlo method to price instruments with discontinuous payoffs and non-smooth trigger functions which allows for a stable computation of Greeks via finite differences. The method extends the idea of smoothing the payoff to the multivariate case. This is accomplished by a coordinate transform and a one-dimensional analytic treatment with respect to the locally most important coordinate and Monte-Carlo sampling with respect to other coordinates. In contrast to other approaches our method does not use importance sampling. This allows to re-use simulated paths to price other instruments or for the computation of finite difference Greeks leading to massive savings in compuational cost. Not using importance sampling leads to a certain bias which is usually very small. We give a numerical analysis of this bias and show that simple local time grid refinement is sufficient to keep the bias always within low limits. Numerical experiments show that our method gives stable finite difference greeks even for situations with payoff discontinuities close to the valuation date.

1 Introduction

Let us briefly describe typical autocallable instruments which make up a good share of the German equity derivatives market. We focus on the multi-asset case here as single asset instruments can be efficiently priced with finite

[*]frank.koster@gmx.de. The opinions expressed in this paper are those of the authors and are not nessecarily the opinions of DekaBank

difference methods. For each period, typically a year, there is one redemption date. On each redemption date one computes a so-called trigger function $U(S_t)$ from the asset spots $S_t^1, .., S_t^d$, for example the worst-of. If $U(S_t)$ was above the coupon level the owner receives a coupon. If $U(S_t)$ is above the redemption level then the product ends and the owner receives the notional. This means there are often two different trigger levels at the redemption date. If the product was not called before or at the last redemption date T the owner receives $U(S_T)$ currency units.

There are more complicated variants such as memory expresses where the owner may not only receive the coupon for the current period but also coupons of previous periods if he or she did not receive these coupons before. Another complication is that the coupon value may be a function of $U(S_t)$ at the redemption date. Last but not least there are instruments with more than one coupon observation date (often daily) for a period and the coupon is only paid if $U(S_t)$ is above the coupon level or within a lower and an upper coupon level at all these dates.

It is well-known that discontinuous payoffs make it hard to compute accurate sensitivities with the Monte-Carlo method by shifting market parameters and calculating the finite difference. There are other methods to compute the sensitivities like likelihood-ratio or path-wise differentiation [5], but these are either not directly applicable to e.g. second order Greeks and/or lack the simplicity and generality of the finite difference method.

Therefore, it is desirable that pricing of the autocallables is so smooth that it allows to compute stable finite difference Greeks. This problem has been addressed by several authors. In Glasserman's classic book on Monte-Carlo methods [5] p. 399 it is shown that smoothing the payoff by analytically computing expected values of the payoff conditional on the state in the time step right before the redemption date effectively replaces the discontinuous payoff by an arbitrarily smooth payoff. Of course the time step size determines how steep the smoothed payoff still is.

A further method is developed in [6] for the single asset case and continuous barriers which exploits that the increments in a time discretization are usually normal or log-normal which allows to compute hit-probabilities analytically and use these to account for the hit-payoffs and simulated paths conditional on not hitting the barrier. A similar work was done in [2] which extends this idea to discontinuous payoffs at discretely monitored barriers of autocallable instruments for the multivariate case and linear or smooth trigger functions U.

The paper [3] extends this work to non-smooth trigger functions like worst-of. Their construction picks up an idea of [4]: analytical tractability with respect to one coordinate (after a clever coordinate transform) is mixed with a Monte-Carlo approach with respect to the other $d-1$ coordinates to get an unbiased scheme. As the authors point out, the coordinate transform is crucial to obtain a smooth dependency of the random numbers on the current state of the path, i.e. to get stable Greeks. However, they gave the transform for $d=2$ only and their approach is rather cumbersome to generalize to higher dimensions as it works with angles.

In this paper we largely follow [3]. One novelty is that we give a very simple construction of the transform for arbitrary dimensions case which uses a Gram-Schmidt orthogonalization and extend it to more general payoffs.

The second novelty of our paper in comparison to [3] and other methods mentioned above is that we do not use importance sampling. For that the simulated paths are not instrument specific. This has a number of advantages:

(i) As the redemption level and the coupon level result in different digital strikes, importance sampling requires a separation of the coupon and redemption payments in several instruments which require different paths. Our method allows to price the separate instruments on the same path or even in one instrument. The speed up is roughly the factor 'number of periods'.

(ii) A simple optimization when computing finite difference Greeks is to compute paths of the shifted constituent only and re-use the paths of the other constituents from the base price call (path recycling). For example, to compute cross gamma for equity instruments one needs to simulate $d + d^2(d+1)$ single constituent paths without path recycling, but only $d + (d+1)$ paths with path recycling. The basket size d is usually between 2 and 10. Path recycling is not possible with importance sampling as the random number stream depends on the state of all constituent paths.

(iii) One can price all autocallable instruments of a portfolio which have the same underlyings with a single set of simulated paths.

A drawback of not using conditional paths is that we introduce a bias. A numerical analysis shows that the bias is relevant only if the next future redemption date is close to the valuation date and the value of the trigger function close to the redemption level. The estimate shows that the bias can be controlled by sufficiently small smoothing time steps. By means of experiments we show that even then payoff-smoothing still allows to compute much more accurate finite difference Greeks than the regular Monte-Carlo scheme.

This paper is organized as follows. In the next section we introduce the problem setup and notation. In the third section we describe the unconditional payoff-smoothing. The generic coordinate transform is described in section four. In section five we analyze the bias. Numerical results are shown in section six.

2 Preliminaries

We denote by $S_t := (S_t^1, ..., S_t^d)^t$ the d-dimensional spot vector at time point t. Here, a^t means the transpose of a vector a. We assume that the basket constituents follow a shifted geometric Brownian motion

$$\begin{aligned} S_t &= \tilde{S}_t + A_t \\ d\tilde{S}_t &= \mu \tilde{S}_t dt + diag(\tilde{S}_t)\sigma L_C dW \ . \end{aligned} \tag{1}$$

where A_t is a d dimensional vector of deterministic offsets, μ, σ are diagonal matrices holding the drift and volatilities and L_C stems from the Cholesky decomposition of the correlation matrix ρ. All these parameters may depend on $\tilde{S}$ or t or additional random factors. We drop these dependencies to keep the notation simple. The vector process (1) covers common dividend models such as Buehler's [1] or pure discrete proportional dividends in which case the offsets A_t are zero.

The trigger functions used for barrier watching and computation of payoffs considered in this paper are

$$\begin{aligned} U(S_t) &= \sum_i a_i S_t^i \quad \text{(weighted basket)} \\ U(S_t) &= \sum_i a_i sort(S_t^1/b_1, ..., S_t^d/b_d)_i \quad \text{(rainbow basket)} \ . \end{aligned}$$

For example, the common case of 'worst-of' trigger function is a rainbow basket with $a = (1, 0, ..0)^t$ and 'best-of' means $a = (0, ..., 0, 1)^t$. The weights $b_1, ..., b_d$ are typically the spots at a forward start date. In our descriptions we concentrate on the rainbow basket type which is more complicated than the plain weighted basket.

The redemption dates are $T_1, ..., T_m$ and the paths are simulated at time points $0 = t_1, ..., t_n = T_m$. The simulation times include the redemption dates. For each redemption date T_r we define T'_r to be the simulation time point right before T_r and $\tau_r := T_r - T'_r$ is the respective time step size, the so-called *smoothing time*. Payoff-smoothing or importance sampling happens in these time steps only. The value of a path is a function of the trigger function values at the redemption dates

$$F(U(S_{T_1}), .., U(S_{T_m})) \ .$$

3 Payoff-Smoothing

The starting point of our work was [3] where the pricing of multi-asset autocallable instruments using importance sampling was considered. Their basic idea is similar to [6]: simulation and valuation of path go hand in hand. The value is initialized with zero and the cumulative weight with the survival probabilities with 1.0. Simulation of time steps is as usual, except for time steps right before a redemption date. Here, they compute the probability P to hit the redemption barrier conditional on $\mathcal{F}_{T'_r}$ and add $weight \cdot P \cdot$ redemption value to the payoff of the path. Then the weight is updated to $(1 - P) \cdot weight$ and the path is simulated with random numbers drawn conditional on not hitting the barrier. Drawing the n conditional random numbers is accomplished by a clever coordinate transform which allows first to draw $n - 1$ unconditional randoms and then draw a single conditional random number.

Our method mixes the above coordinate transform with the payoff-smoothing technique [5], but drops importance sampling: The paths are simulated independent on the path valuation. The valuation of each path ω steps through the redemption dates in reverse order $T_m, T_{m-1}, .., T_1$. At T'_m the path value V^ω_m is initialized as follows: We assume that the $\tilde{S}$-process is (locally) Gaussian from T'_m to T_m such that

$$\begin{aligned} S_{T_m} &= A_{T_m} + \tilde{S}_{T'_m}(1 + \mu\tau) + \sqrt{\tau_m} diag(\tilde{S}_{T'_m})\sigma L_C W, \text{ where } W \sim N(0, id) \\ &=: E + LW \ . \end{aligned} \tag{2}$$

The time index m is dropped in the following to simplify the notation. Now we compute a rotation matrix R such that $LR \cdot e_n$ is a proxy to the gradient of U. Details are given in the next section. Then we draw

$n-1$ independent $N(0,1)$-Gaussians and put them in a n-dimensional vector $\bar{X} := (X_1, ..., X_{n-1}, 0)^t$. This way

$$u(x) := U(E + LR\bar{X} + LR \cdot e_n x) \tag{3}$$

is a strictly monotone (piecewise) linear function of x. Let $F_m(.)$ be the final payoff function (typically piecewise linear, possibly with jumps). Then we may easily compute the expected value

$$E_{X_n}[F_m(u(X_n))], X_n \sim N(0,1)$$

semi-analytically or exact. Discounting gives the initial path value at T'_m

$$V_m^\omega := DF(T'_m, T_m)E[F_m(u(X_n)) \mid \bar{X}] \; . \tag{4}$$

The actually simulated spot $S^\omega_{T_m}$ is never used by our method as we replace the evaluation of $F_m(S^\omega_{T_m})$ by the computation of the expected value. This is the very idea of payoff-smoothing, see Section 7.2.3 of Glassermann's book and that is why we call our method like that.

Because of our assumption that the process is Gaussian from T'_m to T_m we may introduce a small bias in the above step. However this bias is (much) smaller than the bias of the overall time discretization. For that we may neglect it.

For other redemption dates we proceed as above except that the payoff function F_m is replaced by a piecewise constant or piecewise linear function f_r^ω which depends on the path ω. For example for an instrument with redemption level l_R, coupon level $l_C \leq l_R$, redemption amount N and coupon amount C the function is defined by

$$f_r^\omega(u) := \begin{cases} N & \text{if } l_R \leq u \\ DF(T_r, T'_{r+1})V^\omega_{r+1} + C & \text{if } l_C \leq u < l_R \\ DF(T_r, T'_{r+1})V^\omega_{r+1} & \text{else} \end{cases} ,$$

i.e. piecewise constant. The extension to coupon or redemption amounts which depend on the trigger function value is straight forward, but it is important that for $u < l_C$ we can (and will) only use the constant function value $DF(T_r, T'_{r+1})V^\omega_{r+1}$.

The above scheme with f_r^ω is quite simple, but it also introduces a bias. Let us consider the simple case that no coupon is paid. Then the path value at T'_r is computed as

$$DF(T'_r, T_r)E[f_r^\omega(u(X_n)) \mid \bar{X}] = DF(T'_r, T_r)\left(PN + (1-P)DF(T_r, T'_{r+1})V_{r+1}^\omega\right)$$
$$\text{where } P := P[u(X_n) \geq l_R \mid \bar{X}] = 1 - \Phi(u^{-1}(l_R)) \ . \quad (5)$$

However, to avoid the bias we would need to replace V_{r+1}^ω by

$$E[V_{r+1}^\omega \mid \bar{X} \text{ and } u(X_n) < l_R]$$

in (5). Yet, we just have the continuation value $DF(T_r, T'_{r+1})V_{r+1}^\omega$ for the single realization of path ω at this place. So we can't compute the conditional expectation of the continuation value V_{r+1}^ω. MC schemes where all paths are simulated at once from time step to time step would in principle allow to compute a proxy for this conditional expected value with Longstaff-Schwartz like techniques, but we want to have a more simple and general method.

In Section 5 we analyze the bias and also show ways to reduce the bias and keep them within acceptable bounds.

The algorithm can also be formulated in a forward loop stepping from T_1 to T_m which may be simpler for the implementation. In this case one has to carry the survival probability. The respective algorithm for an autocallable with final payoff function F_m, coupon level l_C and redemption level l_R is given in algorithm 1. The details how to compute the transform matrix R or the piecewise linear function u^{-1} are given in the next section.

To price more complicated structures like a memory express we need to split the option in $m+1$ sub-instruments. Each of them is priced with the above scheme, but the paths do not need to be simulated again and also steps 4 and 5 may be reused. This is *not possible* with importance sampling as the sub-instruments typically have different redemption levels and thus require different conditional paths. This means that our method is faster by factor of about m than importance sampling methods.

4 Coordinate Transforms

In the original paper [3] the coordinate transform aims to move the condition $U(S_{T_r}) < R$ to just one of the n random numbers drawn to compute S_{T_r} from $S_{T'_r}$. For our method the reasoning is the same as we want to evaluate just one-dimensional expected values.

The construction of the transform matrix R given below can be used for the importance sampling method as well.

Algorithm 1 Payoff-Smoothing

simulate path $S_{t_1}, ..., S_{t_n}$
$value = 0$, $w = 1$
for $k = 1, m$ **do**
    compute E, L, R using $\tilde{S}$ and σ at T'_k by (2) and (6).
    draw $\bar{X}$
    if $k = m$ **then**
        $value = value + w \cdot DF(0, T_m) E[F_m(u(X_n)) \mid \bar{X}]$ using the density in e.g. (9)
    else
        compute coupon probability $P_C = 1 - \Phi(u^{-1}(l_C))$ by (7) or (8)
        compute redemption probability $P_R = 1 - \Phi(u^{-1}(l_R))$
        $value = value + w \cdot DF(0, T_k) \cdot (P_R \cdot N + P_C \cdot C)$
        $w = w \cdot (1 - P_R)$
    end if
end for
return $value$

We start with the selection of a vector h which must not be orthogonal to the gradient ∇U. As we will see this ensures that u from (3) is strictly monotone. Some reasonable choices for h are

$h := a$	for weighted baskets
$h := B\mathbf{1}$	for any rainbow basket with $\sum_i a_i \neq 0$
with $B := diag(b_1, ..., b_d)$	and $\mathbf{1} := (1, ..., 1)^t$

Now we augment the vector h by an $n \times (n-1)$ matrix Q such that (Q, h) is non singular. For example, for $h = B\mathbf{1}$ matrix $Q = B(e_1, ..., e_{n-1})$ works. The transform matrix is then

$$R := GramSchmidt(L^{-1}(Q, h)) \ , \tag{6}$$

where the Gram-Schmidt orthogonalization starts from the last column.

The above vectors h make the computation of expected values in (4), or hit probabilities in (5), very simple. For example, consider a rainbow basket with $\sum_i a_i \neq 0$ such as worst-of/best-of. Then $LRe_n = B\mathbf{1}/|L^{-1}h|$ and

$$\begin{aligned} u(x) &= \sum_i a_i sort(B^{-1}(E + LR\bar{X} + B\mathbf{1}x/|L^{-1}h|))_i \\ &= u(0) + (\sum_i a_i)/|L^{-1}h|x, \end{aligned}$$

$$\text{and} \quad u^{-1}(v) = |L^{-1}h| \frac{v - u(0)}{\sum_i a_i} \ , \tag{7}$$

The argument for weighted baskets with $h = a$ also leads to such a simple function $u(x)$.

For 2D worst-of/best-of and the context of a logarithmic spot coordinates another h was chosen in [3] where the rotation matrix R is independent on the current state. This avoids to do the Gram-Schmidt procedure again and again. Using the notation of (2) the equivalent for our context is

$$\begin{aligned} h &= diag(\tilde{S}_{T'_m})\sigma\mathbf{1} \\ \text{and e.g. } Q &= diag(\tilde{S}_{T'_m})\sigma(e_1, .., e_{n-1}) \end{aligned}$$

One easily verifies that R then depends on L_C, only, i.e. is constant for constant correlation matrices. This advantage is bought with a slightly more complicated evaluation of u^{-1}, since $u(x)$ is not a linear affine function anymore as in (7), but it is only piecewise linear in x. For example, for a worst-of

$$u^{-1}(v) = |L^{-1}h| \max_i \frac{vb_i - (E + LR\bar{X})_i}{h_i} \ . \tag{8}$$

For a best-of we need to take the min in the above formula. The computation of hit probabilities is still quite simple, but expected values may be more cumbersome than (7) as the density is a normal distribution with piecewise mean and variance parameters:

$$\begin{aligned} P[u(X_n) < v] &= P[X_n < u^{-1}(v)] = \Phi(u^{-1}(v)) \\ \partial_v \Phi(u^{-1}(v)) &= \frac{|L^{-1}h|}{h_{i_0}\sqrt{2\pi}} e^{-\frac{((vb_{i_0} - (E+LR\bar{X})_{i_0})|L^{-1}h|)^2}{2h_{i_0}^2}} \\ \text{where } i_0(v) &:= argmax_i \frac{vb_i - (E + LR\bar{X})_i}{h_i} \ . \end{aligned} \tag{9}$$

Besides computational complexity we may be interested in transforms with good numerical properties. For example, u and, therefore, the hit probabilities and expected values depend on the additional random number vector $\bar{X}$. It seems to be attractive to chose h such that the variance of these quantities is small. To this end we consider a rainbow basket again for the simple, but very common situation that all basket constituent spots (scaled by B^{-1}) are not within a $O(\sqrt{\tau})$ neighborhood of each other. Then U is linear in a neighborhood of E and the variance of $U(E+LR\bar{X})$ is basically

$$\begin{aligned}
|\nabla ULR(e_1,..,e_{n-1})|^2 &= \nabla ULR(e_1,..,e_{n-1})(e_1,..,e_{n-1})^t R^t L^t \nabla U^t \\
&= \nabla ULR(I-e_n e_n^t) R^t L^t \nabla U^t \\
&= \nabla ULL^t \nabla U^t - \nabla U h h^t \nabla U^t / |L^{-1}h|^2 \ .
\end{aligned}$$

First of all this shows that the variance of the expected values computed in our pricing algorithm does not depend on the particular choice of Q. Moreover, the variance is zero if

$$h = LL^t \nabla U^t \ . \tag{10}$$

The above situation of a good separation of the scaled components of S has a quite high probability. Yet, we need a vector h which is always well-defined and which always depends smoothly on the input data. Even if ∇U does not exist in E.

For worst-of rainbow baskets we propose to replace ∇U by $\nabla \tilde{U}$ where

$$\begin{aligned}
\tilde{U}(S) &:= m(S^1/b_1, m(S^2/b_2, ...m(S^{d-1}/b_{d-1}, S^d/b_d)...)) \\
m(x,y) &:= \frac{1}{2}(x+y-\sqrt{\epsilon^2+(x-y)^2})
\end{aligned}$$

for some suitable smoothing parameter $\epsilon > 0$. Function $m(x,y)$ approximates the minimum of x and y. For best-of rainbows we use $m(x,y) := \frac{1}{2}(x+y+\sqrt{\epsilon^2+(x-y)^2})$. Since all components h_i are > 0, we can again use (8) and (9) to compute hit probabilities and expected values.

We conclude this section with remarks on spreads. For example, the 2D spread 'best - worst' is a rainbow basket with $a = (-1,1)$, i.e $\sum_i a_i = 0$ where or simple choices for h do not function anymore. As an alternative we may chose

$$h = Ba,$$

but then $u(x)$ is not strictly monotone anymore. But, it is strictly monotone for $x < x_0$ and $x > x_0$ where x_0 is the unique solution of $u(x_0) = 0$. This means that the conditional redemption probability P is given by

$$\begin{aligned}P[u(X_n) \geq l_R \mid \bar{X}] &= P[u(X_n) \geq l_R \mid \bar{X} \text{ and } X_n > x_0] + \\ &\quad P[u(X_n) \geq l_R \mid \bar{X} \text{ and } X_n < x_0] \\ &= 1 - \Phi(u^{-1}_{]x_0,\infty[}(l_R)) + 1 - \Phi(u^{-1}_{]-\infty,x_0[}(l_R))\end{aligned}$$

where e.g $u_{]x_0,\infty[}$ is u restricted to the domain $]x_0, \infty[$ where it is a bijection.

While this was still manageable, things get really complicated for the best-worst spread of more than two underlyings. Here we didn't found a simple solution. But, this case is of little practical importance.

5 Analysis of the Bias

There are basically three contributions to the bias in our scheme. The first is the usual discretization error of the path simulation. The second is introduced by the Brownian motion assumption in the smoothing step before T_r, see (2). This is the single time step time discretization error of the (non-log) Euler scheme. These two contributions are well understood, see the standard literature [5], [7], and they are usually quite small for reasonable time grids. The third contribution is intrinsic to step (5) of our scheme where we use the plain continuation value instead of the conditional expectation.

For the analysis we consider the 1D case only. Despite this simplification our analysis captures the essential features also for the multivariate case because the critical part is along the nth coordinate only.

To begin with, recall that $T_r, 1 \leq r \leq m$ are the redemption dates and T'_r are the simulation time points right before T_r. Moreover we need the time distances $\tau_r := T_r - T'_r$. The valuation date is $t = 0$ and T_1 is > 0. For the analysis we need the assumption that the redemption dates are not too close to each other, more precisely

$$T_{r+1} - T_r \gg \tau_r, \tau_{r+1} \ . \tag{11}$$

For simplicity we ignore any dividends and set the drift and interest rates to zero. Volatility σ is constant. In this plain Black-Scholes setting we may easily switch to log spot coordinates which are denoted by s. The log redemption level is l_R for all redemption dates and the rebate is always 1. There are no coupons. The payoff function $F(s_{T_1}, ..., s_{T_m})$ is decomposed recursively

$$
\begin{aligned}
F_m(s_{T_m}) &:= \text{ any function of } s_{T_m} \text{ for the final redemption date} \\
F_r(s_{T_r}, s_{T_{r+1}}, ..., s_{T_m}) &:= \begin{cases} 1 & \text{if } s_{T_r} \geq l_R \\ F_{r+1}(s_{T_{r+1}}, .., s_{T_m}) & \text{else} \end{cases} \quad .
\end{aligned}
$$

This means that $F = F_1$.

The expected values of the decomposed price functions conditional on the spot in T'_r are denoted by

$$V_r(s) := E[F_r(s_{T_r}, ..., s_{T_m}) \mid s_{T'_r} = s] \ .$$

Our method produces biased versions $\tilde{V}_r(s)$ of these expected values and the bias is

$$\epsilon_r(s) := \tilde{V}_r(s) - V_r(s) \ .$$

For the final redemption date we compute $\tilde{V}_m(s)$ as exact expected value in (4). Thus, only the time discretization error matters here. As explained above we may neglect this error and, therefore, have

$$\epsilon_m(s) = 0 \quad \text{ for all } s \ .$$

Now, consider the more interesting case $r < m$. The exact continuation value of the autocallable instrument in T_r is the conditional expectation

$$V_r^c(s) := E[F_{r+1}(s_{T_{r+1}}, ..., s_{T_m}) \mid s_{T_r} = s] \ .$$

The hit probability conditional on T'_r and spot s at this time is

$$P_r(s) := P(s_{T_r} \geq l_R \mid s_{T'_r} = s) \ .$$

The unbiased decomposed prices $V_r(s)$ can be written as

$$V_r(s) = P_r(s) + (1 - P_r(s))E[V_r^c(s_{T_r}) \mid s_{T'_r} = s \text{ and } s_{T_r} < l_R] \ . \tag{12}$$

On the other hand, the respective biased prices $\tilde{V}_r(s)$ are

$$\tilde{V}_r(s) := P_r(s) + (1 - P_r(s))E[\tilde{V}_r^c(s_{T_r}) \mid s_{T'_r} = s] \ , \tag{13}$$

with biased continuation value $\tilde{V}_r^c(s)$.

For the analysis of the bias we have to take into account that i) we ignore the conditioning on $s_{T_r} < l_R$ in (13) and ii) the accumulation of biases from redemption dates $r+1,..,m$ which enter here by the biased continuation values $\tilde{V}_r^c(s)$.

Because of i) and ii) we have for the biased continuation value

$$\tilde{V}_r^c(u) = \int_{s_{T'_{r+1}}=-\infty}^{\infty} (V_{r+1} + \epsilon_{r+1})(s_{T'_{r+1}}) p(s_{T'_{r+1}} \mid s_{T_r} = u) ds_{T'_{r+1}}$$

where $p(. \mid .)$ is the transition density between T_r and T'_{r+1}

$$p(s_{T'_{r+1}} \mid s_{T_r} = u) = \frac{1}{\sqrt{2\pi(T'_{r+1} - T_r)}\sigma} exp\left(-\frac{(s_{T'_{r+1}} - u + \frac{1}{2}\sigma^2(T'_{r+1} - T_r))^2}{2\sigma^2(T'_{r+1} - T_r)}\right).$$

We use $p(. \mid .)$ also for the transition between T'_r and T_r which is defined similar to above. The bias ϵ_r is

$$\begin{aligned}
\epsilon_r(s) = (1 - P_r(s)) &\int_{s_{T_r}=-\infty}^{\infty} \int_{s_{T'_{r+1}}=-\infty}^{\infty} (V_{r+1} + \epsilon_{r+1})(s_{T'_{r+1}}) p(s_{T'_{r+1}} \mid s_{T_r}) ds_{T'_{r+1}} p(s_{T_r} \mid s) ds_{T_r} \\
&- \int_{s_{T_r}=-\infty}^{l_R} \int_{s_{T'_{r+1}}=-\infty}^{\infty} V_{r+1}(s_{T'_{r+1}}) p(s_{T'_{r+1}} \mid s_{T_r}) ds_{T'_{r+1}} p(s_{T_r} \mid s) ds_{T_r} \\
= (1 - P_r(s)) &E[\epsilon_{r+1}(s_{T'_{r+1}}) \mid s_{T'_r} = s] \\
+ (1 - P_r(s)) &\int_{s_{T_r}=-\infty}^{\infty} V_r^c(s_{T_r}) p(s_{T_r} \mid s) ds_{T_r} - \int_{s_{T_r}=-\infty}^{l_R} V_r^c(s_{T_r}) p(s_{T_r} \mid s) ds_{T_r}
\end{aligned}$$

The second row simplifies to

$$\int_{s_{T_r}=l_R}^{\infty} V_r^c(s_{T_r}) p(s_{T_r} \mid s) ds_{T_r} - P_r(s) \int_{s_{T_r}=-\infty}^{\infty} V_r^c(s_{T_r}) p(s_{T_r} \mid s) ds_{T_r} =: Q$$

Under quite weak assumptions on F_m, and if the redemption dates are not too close to each other, the continuation value V_r^c is a smooth function. We employ Taylor expansion around $s^* := s - \frac{1}{2}\sigma^2\tau_r$ and neglect second and higher order terms. With $\Delta_r = dV_r^c/ds_{T_r}$ this leads to

$$\begin{aligned}
Q \approx & \int_{s_{T_r}=l_R}^{\infty} (V_r^c(s^*) + \Delta_r(s_{T_r} - s^*))p(s_{T_r} \mid s)ds_{T_r} \\
& - P_r(s) \int_{s_{T_r}=-\infty}^{\infty} (V_r^c(s^*) + \Delta_r(s_{T_r} - s^*))p(s_{T_r} \mid s)ds_{T_r} \\
= & \Delta_r \left(\int_{s_{T_r}=l_R}^{\infty} (s_{T_r} - s^*)p(s_{T_r} \mid s)ds_{T_r} - P_r(s) \int_{s_{T_r}=-\infty}^{\infty} (s_{T_r} - s^*)p(s_{T_r} \mid s)ds_{T_r} \right) \\
= & \Delta_r \Big(\int_{s_{T_r}=l_R}^{\infty} (s_{T_r} - s^*)p(s_{T_r} \mid s)ds_{T_r} - P_r(s) \cdot 0 \Big) \\
= & \Delta_r \sigma \sqrt{\frac{\tau_r}{2\pi}} \int_{s_{T_r}=l_R}^{\infty} \frac{s_{T_r} - s^*}{\sigma^2 \tau_r} exp\left(-\frac{(s_{T_r} - s^*)^2}{2\sigma^2\tau_r} \right) ds_{T_r} \\
= & \Delta_r \sigma \sqrt{\frac{\tau_r}{2\pi}} exp\left(-\frac{(l_R - s + \frac{1}{2}\sigma^2\tau_r)^2}{2\sigma^2\tau_r} \right) \ , \qquad (14)
\end{aligned}$$

This is the important contribution to the bias.

We assume that $|\Delta_r|$ is globally bounded by Δ then for $r < m$

$$\begin{aligned}
|\epsilon_r(s)| \quad \leq \quad & (1 - P_r(s))E[|\epsilon_{r+1}(S_{T'_{r+1}})| \mid S_{T'_r} = s] \\
& + \ c\Delta\sigma\sqrt{\tau_r} exp\left(-\frac{(l_R - s + \frac{1}{2}\sigma^2\tau_r)^2}{2\sigma^2\tau_r} \right) \ , \qquad (15)
\end{aligned}$$

where $c > 1/\sqrt{2\pi}$ is a suitable constant to compensate for the neglected high order terms in (14). Now we show by induction that the total bias is bounded by

$$|\epsilon_r(s)| \leq c\Delta\sigma\sqrt{\tau_r} exp\left(-\frac{(l_R - s + \frac{1}{2}\sigma^2\tau_r)^2}{2\sigma^2\tau_r} \right) + O(\sum_{i=r+1}^{m-1} \tau_i) \ . \qquad (16)$$

For $r = m$ we have $|\epsilon_m(s)| = 0$ as explained in the beginning of this section. For $r = m - 1$ there is no accumulated error and therefore,

$$|\epsilon_{m-1}(s)| \ \leq \ c\Delta\sigma\sqrt{\tau_{m-1}} exp\left(-\frac{(l_R - s + \frac{1}{2}\sigma^2\tau_{m-1})^2}{2\sigma^2\tau_{m-1}} \right) \ .$$

For $r < m - 1$ we use the recursion (15) and

$$E[|\epsilon_{r+1}(s_{T'_{r+1}})| \mid s_{T'_r} = s]$$

$$\begin{aligned}
&\leq \; E\Big[c\Delta\sigma\sqrt{\tau_{r+1}}exp\left(-\frac{(l_R - s_{T'_{r+1}} + \frac{1}{2}\sigma^2\tau_{r+1})^2}{2\sigma^2\tau_{r+1}}\right) + O(\sum_{i=r+2}^{m-1}\tau_i) \mid s_{T'_r} = s\Big] \\
&\leq \; c\Delta\sigma\sqrt{\tau_{r+1}}E\Big[exp\left(-\frac{(l_R - s_{T'_{r+1}} + \frac{1}{2}\sigma^2\tau_{r+1})^2}{2\sigma^2\tau_{r+1}}\right) \mid s_{T'_r} = s\Big] + O(\sum_{i=r+2}^{m-1}\tau_i) \\
&= c\Delta\sigma\frac{\tau_{r+1}}{\sqrt{T_{r+1} - T'_r}}exp\left(-\frac{(l_R - s + \frac{1}{2}\sigma^2(T_{r+1} - T'_r))^2}{2\sigma^2(T_{r+1} - T'_r)}\right) + O(\sum_{i=r+2}^{m-1}\tau_i) \\
&= O(\sum_{i=r+1}^{m-1}\tau_i) \;.
\end{aligned}$$

This proves estimate (16). By a similar computation as above for $E[exp(..) \mid S_{T'_r} = s\Big]$ we can compute the bias in the final price at $t = 0$ and initial log spot s_0

$$E[\epsilon_1(s_{T'_1}) \mid s_0] = c\Delta\sigma\frac{\tau_1}{\sqrt{T_1}}exp\left(-\frac{(l_R - s_0 + \frac{1}{2}\sigma^2 T_1)^2}{2\sigma^2 T_1}\right) + O(\sum_{r=2}^{m-1}\tau_i). \quad (17)$$

This estimate shows that our method systematically overprices standard products and that the bias is usually $O(\sum_r \tau_r)$. But, if $T_1 \to \tau_1$ and if the spot s_0 is close to the redemption level the bias is of order $\sqrt{\tau_1}$ only. For that we should take a rather small time step τ_1 for the first redemption date T_1. This should be of order $O((average\ time\ step)^2)$. Then the total bias is not much larger than that of the Euler-scheme itself. The contribution of all other redemption dates to the bias is still first order $O(\tau_r)$ which means that these time steps do not need to be so small. In our code we employed a quite simple heuristics to set the smoothing time steps τ_r depending on T_r and parameters τ_{near} and τ_{far} which control the smoothing time at the close and far end, respectively.

$$\tau_r := \tau_{near} q + \tau_{far}(1 - q) \;, \quad q := \frac{2}{\pi} atan(\sqrt{T_r}\frac{\pi}{2}) \;. \quad (18)$$

In Section 6 we show that even for quite small smoothing time steps the convergence of Greeks is pretty good and that the benefit over conventionial

Monte-Carlo is larger for difficult Greeks like gamma than for more simple Greeks like delta.

Besides a careful choice of the smoothing time steps we may use the above estimates to reduce the bias. One idea is to pick a suitable (may be even coarse) proxy $\tilde{\Delta}_r$ to dV_r^c/dS_{T_r} and approximate the trigger process $U(S_t)$ by a Brownian motion with (local) volatility $\tilde{\sigma}$. Then we approximate in step (5) the unbiased conditional expectation by

$$
\begin{aligned}
(1-P)E[V_r^c(S_{T_r}) \mid S_{T_r'} \text{ and } U(S_{T_r}) < l_R]& \\
\approx (1-P_r)E[V_r^c(S_{T_r}) \mid S_{T_r'}]& \\
- \tilde{\Delta}_r \tilde{\sigma} \sqrt{\frac{\tau_r}{2\pi}} exp \left(- \frac{(l_R - U(S_{T_r'}) + \frac{1}{2}\tilde{\sigma}^2\tau_r)^2}{2\tilde{\sigma}^2\tau_r} \right)& \qquad (19)
\end{aligned}
$$

instead of just $(1-P_r)E[V_r^c(S_{T_r}) \mid S_{T_r'}]$. This is done in each smoothing time step for each path to be able to use a locally good proxy for the volatility.

In our context of pricing an autocallable option the proxy delta $\tilde{\Delta}$ may be the Black-Scholes delta of a single underlying autocallable instrument with similar payoff as the original instrument.

If the trigger function is a simple weighted basket then the proxy variance is simply

$$\tilde{\sigma}^2 := \sum_{i,j} a_i S^i \sigma_i \rho_{ij} S^j a_j \sigma_j \ .$$

If the trigger function is a worst-of or best-of there are the following extreme cases

- The weighted constituents are quite separated and i_0 is the index of the worst-/best-of constituent. Then the volatility of the trigger process should be $\sigma_{i_0} S_{i_0}/b_{i_0}$. This is the most common situation as long as the correlations are not very close to 1.
- The weighted constituents are all nearly at the same level. Then the volatility of the trigger process should be lower than the individual local volatilities.

The following blending formula accounts for these considerations

$$
\begin{aligned}
v_i &:= 2(\sigma_i S^i_{T'_r}/b_i)^2 \tau_r \\
w_i &:= \frac{v_i}{v_i + (S^i_{T'_r}/b_i - U(S_{T'_r}))^2} \\
d^+ &:= \sum_i w_i \quad \text{(effective dimension as } d^+ \in [1, d[\ ) \\
\eta^+ &:= \sqrt{\sum_i w_i v_i / (d^+ \tau_r)} \quad \text{(average total volatility)} \\
\tilde{\sigma} &:= \eta^+ (1/d^+ + \frac{3}{4}(1 - 1/d^+)) \quad \text{(reduce volatility if } d^+ > 1) \qquad (20)
\end{aligned}
$$

In Section 6 we show that the modifications from (19) can significantly reduce the bias.

6 Numerical Experiments

In this section we report numerical results for synthetic autocallable instruments. These instruments have three yearly periods. The redemption value is 1 and there are no coupons and no final payoff. Therefore, the delta of continuation value Δ is larger than for common autocallable products where the final payoff is typically a zero strike call below the final redemption level. In our tests the redemption levels are all the same and we do vary this level.The first redemption date T_1 is one day and10 days ahead.

The trigger function is a worst-of three constituents. The volatilities are flat 25%, no interest, no dividends and correlations are = 0.7. The initial spots are all 100.0 as this is the worst case from a numerical perspective. If all constituents start at the same level then the worst-of process behaves locally quite different from a plain geometrical or Brownian motion.

All instruments were priced using a regular Monte-Carlo and our payoff-smoothing method with 12500, 25000,..., 3.2mio paths. For the most expensive simulation the standard error was always < 0.0003 (=3 Bps). For both methods, the number of time steps was roughly 60 time steps per year with local refinement at the front end to improve convergence for simulations with real market data. We used the variance minimizing h of (10) for payoff-smoothing.

(Bias) Our first focus is on the bias introduced by our scheme. For that we compare prices obtained with 3.2mio paths and redemption levels ranging from 80 to 120 in Figure 1. Price differences are scaled by 10000. Payoff-smoothing was done in four variants: two different smoothing times, with and without debiasing (19).

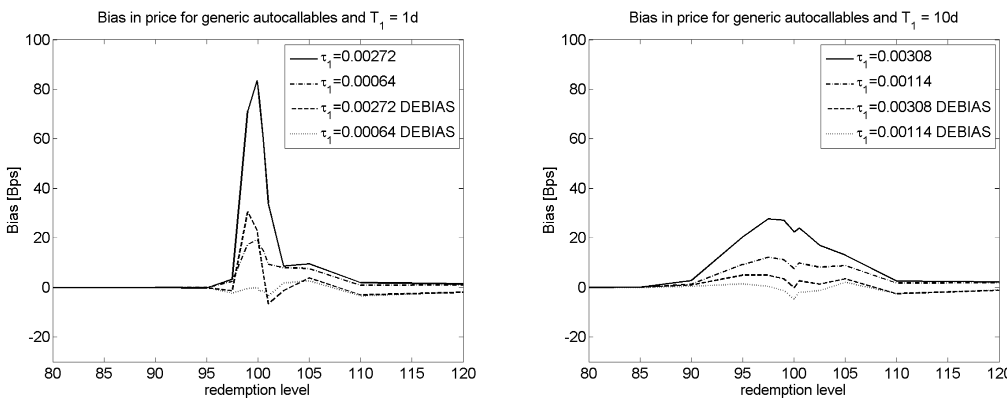

Figure 1: Price differences to regular MC at 3.2mio simulations scaled by 10000. Left $T_1 = 1d$. Right $T_1 = 10d$.

The results show that with $\tau_1 = 0.0006$ the bias is below 20 Bps even for the very hard situation that the trigger function value is very close to the redemption level one day before a redemption date. This is just acceptable. The results with the small smoothing time and debiasing are very good, while the results with the larger smoothing time $\tau_1 = 0.00272$ without debiasing are not good enough. If the redemption date is further in the future, or if the redemption level is not too close to ATM, the bias is much lower which is in agreement with (17). This estimate also predicts that the payoff-smoothing consistently overestimates the true price which is also seen in the test results. Of course, we did more tests with T_1 further in the future. The results were better than for 10days ahead, in particular for the large τ_1 and no debiasing.

(Greeks) The next experiments compare the accuracy of the finite-difference gamma which is usually quite noisy for the Monte-Carlo method. We used a spot shift of 1%. In Figure 2 the convergence for gamma with respect to the first constituent is shown for the two autocallable instruments with redemption level = 100 (ATM) and 101. The results for the other redemption levels are of similar or better quality. The plotted numbers are the scaled differences

$$(gamma_{sims}^{method} - gamma_{3.2mio\ sims}^{MC}) \cdot 10000 \cdot (0.03 \cdot spot_1)^2 \frac{1}{2} \qquad (21)$$

to see the impact of gamma errors on a P&L explain under a hypothetical shift of the market spot by 3%. Ideally this should be of a similar order as the error in price, i.e. less than 20Bps. The respective scaling for delta is $(delta_{sims}^{method} - delta_{3.2mio\ sims}^{MC}) \cdot 10000 \cdot (0.03 \cdot spot_1)$. For $T_1 = 1d$ the best

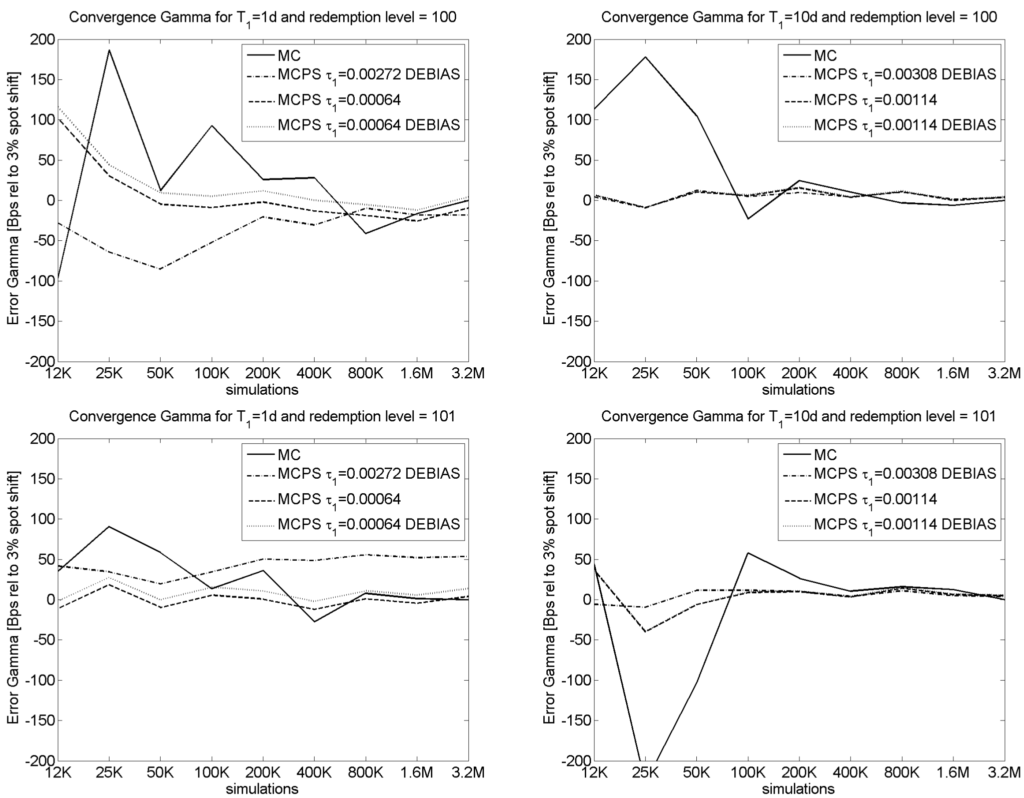

Figure 2: Gamma differences to regular MC vs # simulations. Left $T_1 = 1d$ Right $T_1 = 10d$.

result is obtained with the small $\tau_1 = 0.00064$ and the difference between debiased and not debiased is rather small. The results for $\tau_1 = 0.00272$ are too biased, even with the debiasing technique. For $T_1 = 10d$ (and larger) the picture changes a little bit. In particular for the redemption level of 101 (and others) the larger smoothing time stabilizes the computation of gamma without the penalty of a significant bias. This means that we probably should fine tune the parameters in (18) which is not the scope of this report.

It is well-known that the shift-size, denoted by δ in the following, is crucial for the computation of finite difference Greeks as it determines the variance which controls the standard error [5] Section 7. In his book, Glasserman explains that for second order Greeks the variance is basically $O(\delta^{-3})$ for discontinuous payoffs, but $O(\delta^{-1})$ for continuous payoffs. In terms of the Monte-Carlo standard error (root mean square error) this means that this error should scale like $O(\delta^{-3/2})$, or $O(\delta^{-1/2})$ respectively for fixed number of simulations. As he also explains in Section 7.2.3. smoothing discontinuous payoffs may shift the behavior from $O(\delta^{-3/2})$ to $O(\delta^{-1/2})$. This is what we see with our method.

As an example, we computed gamma (and delta) w.r.t. the first constituent with shift-sizes $1\%, 0.1\%$ and 0.01% of the spot for the generic autocallable with redemption level 100 and $T_1 = 1d$, i.e a very hard situation from the numerical point of view. The smoothing time was $\tau_1 = 0.00064$ as this was the parameter which performed best in our previous experiments.

The results are shown in Table 1. The numbers in brackets are the standard errors scaled by 10000 for the price and like (21) for gamma.

For payoff-smoothing the standard error for the 200K simulations with is 0.00016 with spot-shift 1% and 0.0004 with spot-shift 0.0037%. The ratio $0.0037/0.0004 = 9$ while the variance estimates predicts a ratio of $\sqrt{1/0.01} = 10$. For the regular Monte-Carlo scheme, the respective standard errors are 0.0008 and 0.7847. The ratio $0.7847/0.0008 = 980$ is close to the predicted ratio of $\sqrt{(1/0.01)^3} = 1000$.

This means that the regular Monte-Carlo scheme may not be used together with such small shift-sizes while the method presented here is quite robust with respect to very small shift-sizes even for very hard situations.

For first order Greeks like delta the arguments of [5] predict a variance $O(\delta^{-1})$ for the regular Monte-Carlo scheme and $O(1)$ for payoff-smoothing. Again this is the case in Table 1.

Similar results are obtained by other authors, see [2], [3], [8], [9] to give an incomplete list.

value	sims	MC	stdErr(Bps)	PS	stdErr(Bps)
price	12K	0.5435	0.0045(45)	0.5490	0.0038(38)
price	50K	0.5489	0.0022(22)	0.5480	0.0019(19)
price	200K	0.5487	0.0011(11)	0.5496	0.0010(10)
delta_1 @ 1%	12K	0.0572	0.0014(43)	0.0585	0.0010(29)
delta_1 @ 1%	50K	0.0616	0.0007(22)	0.0628	0.0005(15)
delta_1 @ 1%	200K	0.0621	0.0004(11)	0.0619	0.0002(7)
delta_1 @ 0.1%	12K	0.0536	0.0046(138)	0.0604	0.0014(42)
delta_1 @ 0.1%	50K	0.0667	0.0026(77)	0.0651	0.0007(22)
delta_1 @ 0.1%	200K	0.0643	0.0013(38)	0.0645	0.0004(11)
delta_1 @ 0.01%	12K	0.0280	0.0106(317)	0.0610	0.0014(43)
delta_1 @ 0.01%	50K	0.0610	0.0078(234)	0.0653	0.0007(22)
delta_1 @ 0.01%	200K	0.0620	0.0039(118)	0.0646	0.0004(11)
gamma_1 @ 1%	12K	-0.0506	0.0030(135)	-0.0459	0.0014(61)
gamma_1 @ 1%	50K	-0.0482	0.0016(70)	-0.0483	0.0007(32)
gamma_1 @ 1%	200K	-0.0479	0.0008(35)	-0.0482	0.0004(16)
gamma_1 @ 0.1%	12K	0.0960	0.0926(4167)	-0.0548	0.0050(225)
gamma_1 @ 0.1%	50K	-0.0580	0.0517(2324)	-0.0542	0.0025(111)
gamma_1 @ 0.1%	200K	-0.0390	0.0254(1141)	-0.0588	0.0013(57)
gamma_1 @ 0.01%	12K	0.8000	2.1166(95247)	-0.0764	0.0170(763)
gamma_1 @ 0.01%	50K	1.0000	1.5620(70292)	-0.0524	0.0074(331)
gamma_1 @ 0.01%	200K	-1.9000	0.7874(35433)	-0.0556	0.0037(165)

Table 1: Price delta and gamma with MC and payoff-smoothing for different numbers of simulations and spot-shift sizes. The instrument has redemption level 100 and $T_1 = 1d$.

7 Conclusions

The method presented in this paper extends the idea of payoff-smoothing as in [5] to the multivariate case by means of a coordinate transform and local one-dimensional analytic treatment as in [3]. In contrast to other approaches our method does not use importance sampling which has significant algorithmic advantages for the pricing of more complex autocallable instruments, or the use of path recycling for the computation of Greeks. For example, for equity Greeks one can achieve speeds ups of order $O(d)$ for single constituent Greeks like delta, gamma, vega and $O(d^2)$ for cross gamma, vega. We have introduced a generic method to compute the coordinate transform that extends the scheme in [3] and can be used also in the context of this importance sampling method.

Unlike importance sampling based methods, our method introduces a bias for which we derived a suitable estimate. This estimate shows that the bias is relevant only if the next redemption date is close to the valuation date and if the trigger function value is close to the next redemption level. The bias can be easily kept within acceptable limits by a careful choice of the smoothing time step size or, with some more implementation effort, by a debiasing scheme based on the estimate.

Numerical experiments have shown that the methods allows to compute stable finite-difference Greeks and that the standard error for the Greeks grows quite slowly if the shift size goes to zero.

References

[1] *Volatility and dividends - volatility modelling with cash dividends and simple credit risk* H. Buehler, Available at SSRN: http://ssrn.com/abstract=1141877

[2] *Conditional analytic Monte-Carlo pricing scheme of autocallable products* C.P. Fries and M.S. Joshi, Available at SSRN: http://ssrn.com/abstract=1125725

[3] *A Monte-Carlo pricing algorithm for autocallables that allows for stable differentiation* T. Alm, B. Harrach, D. Harrach and M. Keller, J. Comput. Finance, Vol. 17.2013, 1, p. 43-70

[4] J. Geweke *Efficient simulation from the multivariate mormal and student t-distributions subject to linear constraints*; Proceedings of the 23rd symposium on the interface, 571578, 1991.

[5] P. Glasserman *Monte-Carlo methods in financial engineering*; vol. 53 of Stochastic Modelling and Applied Probability, Springer 2003.

[6] P. Glasserman and J. Staum *Conditioning on one-step survival for barrier option simulations*; Operations Research 49(6), 923937, 2001.

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

[8] G. Liu and L.J. Hong *Kernel estimation of the Greeks for options with discontinuous payoffs*; Operations Research, 59, pp. 96-108.

[9] G. Liu *Importance sampling for option Greeks with discontinuous payoffs*; Operations Research 49(6), under review by Informs Journal on Computing.