This is a preprint of an article that has appeared in The Journal of Computational Finance 17 (1), 43-70, 2013.
http://www.risk.net/journal-of-computational-finance/journal/2292302/latest-issue-volume-17-issue-1-2013

A MONTE CARLO PRICING ALGORITHM FOR AUTOCALLABLES THAT ALLOWS FOR STABLE DIFFERENTIATION

THOMAS ALM[†], BASTIAN HARRACH[‡], DAPHNE HARRACH[†] AND MARCO KELLER[†]

Abstract. We consider the pricing of a special kind of options, the so-called autocallables, which may terminate prior to maturity due to a barrier condition on one or several underlyings. Standard Monte Carlo (MC) algorithms work well for pricing these options but they do not behave stable with respect to numerical differentiation. Hence, to calculate sensitivities, one would typically resort to regularized differentiation schemes or derive an algorithm for directly calculating the derivative.

In this work we present an alternative solution and show how to adapt a MC algorithm in such a way that its results can be stably differentiated by simple finite differences. Our main tool is the one-step survival idea of Glasserman and Staum which we combine with a technique known as GHK Importance Sampling for treating multiple underlyings.

Besides the stability with respect to differentiation our new algorithm also possesses a significantly reduced variance and does not require evaluations of multivariate cumulative normal distributions.

Key words. Monte Carlo, Autocallables, Express Certificate, sensitivities, stable differentiation, one-step survival, GHK importance sampling, variance reduction

1. Introduction. The application of Monte Carlo (MC) simulation to option pricing started with a paper by Boyle [4] who was the first to use MC to price European call options on dividend paying stocks as an extension to the analytical solution for options on non-dividend paying stocks published in the seminal Black-Scholes paper four years before [2]. Boyle already stressed the point that it is advisable to go beyond the standard MC approach by applying techniques to effectively reduce the variance, reasoning however "that it is not possible to formulate general rules regarding the selection and implementation of the most effective technique“ [4]. A variety of techniques already known from areas other than option pricing have been successfully adopted to improve the MC pricing for various types of uni- and multivariate options. Some new techniques have been developed to price specific types of exotic options, making use of distinct features of their respective payoffs. For an overview about the various approaches we refer to the monograph by Glasserman [9].

Within this paper, we will consider MC pricing schemes for so-called autocallable options. Autocallables have been traded intensely in particular in the market for structured retail derivatives (certificates). As an example, autocallables, which are known under the trade name of *Express Certificates* in the German market for retail certificates made up for more than 30% of this market in 2008, see, e.g., Wegner and Alm [1] for an overview. Most of the certificates of this type issued had one underlying (Single Express Certificates), some had two underlyings (Duo Express Certificates). Autocallables with one, resp., multiple underlyings are also called uni-, resp., multivariate.

The name *autocallable* refers to the fact that, just as with a normal callable bond, the autocallable can be terminated prior to its maturity. The difference, however, is that the autocallable is called automatically as soon as a certain barrier condition is fulfilled on one of some predefined observation dates, whereas a callable bond has to be called by its issuer. More precisely, the idea of this type of financial instrument is as follows (cf. figure 1.1): At specific points in time (the observation dates), it is

[†]BHF-BANK, Frankfurt, Germany (thomas.alm@bhf-bank.com, marco.keller@bhf-bank.com and daphne.von-harrach@bhf-bank.com)

[‡]Birth name: Bastian Gebauer, Department of Mathematics, Technische Universität München, Germany (harrach@ma.tum.de)

This is a preprint of an article that has appeared in The Journal of Computational Finance 17 (1), 43-70, 2013.
http://www.risk.net/journal-of-computational-finance/journal/2292302/latest-issue-volume-17-issue-1-2013

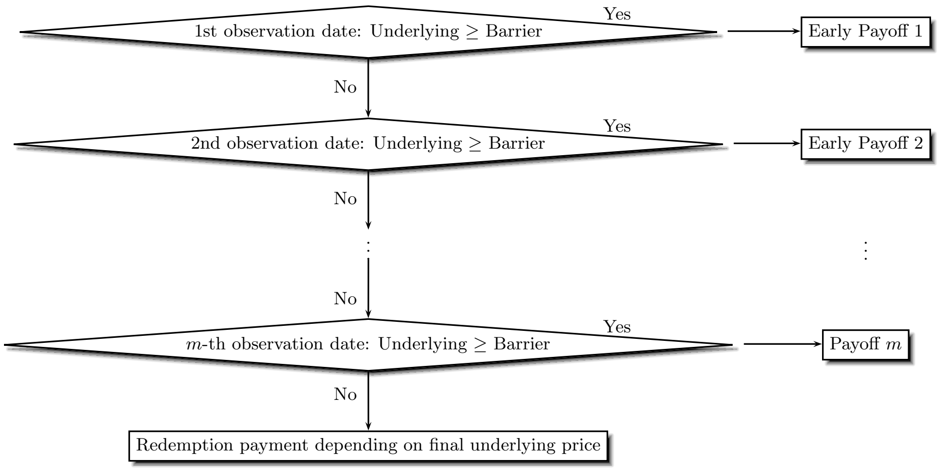

FIG. 1.1. *Payoff profile of an autocallable*

checked whether the underlying (resp., a function of several underlyings) reaches a certain barrier. If this is the case, the buyer of the autocallable gets a pre-defined constant cash-flow (payoff) and the certificate terminates. Otherwise the instrument continues to exist until the next observation date, and so forth. For those cases, where the autocallable survives until maturity, a payoff depending on the underlying(s) is generated at the maturity date. Autocallables can be considered as an exotic type of barrier options, cf., e.g., Zhang [17], the so-called window-barrier knock-out options with rebate, and with the number of windows corresponding to the number of observation dates, and each window having zero length. They are, however, typically treated as an option class of its own (see, e.g., Bouzoubaa and Osseiran [5]).

We consider the pricing of autocallables under the standard assumption that the option price is described by the discounted expected payoff using (correlated) geometric Brownian motion as the stochastic process for the underlyings, see, e.g. Hull [13]. For specific types of autocallables, analytical (closed-form) solutions have been derived in Reder [15]. However, these solutions require the evaluation of multivariate cumulative normal distributions, with the dimension depending on the product of the number of underlyings and the number of observation dates. Higher-variate cumulative normal distributions require numerical methods (see, e.g., Genz and Bretz [7]) and are typically evaluated by Monte Carlo simulations. Since, furthermore, autocallables are an option class which involves a variety of possible payoffs [5], this suggests to use Monte Carlo simulation as a generic way of pricing autocallables. However, it is worth mentioning that closed-form solutions, even if they exist for certain special cases only, may serve as a useful input to improve Monte Carlo simulations, e.g., by using control-variate methods.

In this work, we derive an efficient Monte Carlo algorithm for pricing autocallables, and for calculating their sensitivities. In doing so, our goals are twofold. First, we aim at obtaining reliable prices with as few MC paths as possible. Second, we want to stably determine the option's sensitivities such as Delta (derivative w.r.t. the price of the underlying) and Vega (derivative w.r.t. the option's implied volatility) from the numerically calculated prices.

Note that the second goal is particularly ambitious. Differentiation is generally unstable, as even the smallest numerical errors in the price may have arbitrarily large

This is a preprint of an article that has appeared in The Journal of Computational Finance 17 (1), 43-70, 2013.
http://www.risk.net/journal-of-computational-finance/journal/2292302/latest-issue-volume-17-issue-1-2013

effects on the derivatives. In the mathematical literature this property is known as *ill-posedness*, cf., e.g., Engl, Hanke and Neubauer [6], and, in general, requires sophisticated regularization strategies, cf., e.g., Hanke and Scherzer [12]. Our aim in this work is, however, to design a MC pricing strategy that allows stable differentiation by simple finite differences.

For the univariate express certificate, both our goals can be achieved using the ideas of one-step survival established by Glasserman and Staum [10]. The idea is to restrict the Monte Carlo simulation to the survival zone (i.e., the area below the barrier) and to treat the thus neglected contributions from the barrier analytically. In other words, instead of sampling from the full normal distribution, a truncated normal distribution excluding values above the barrier is used. This approach yields a substantial variance reduction [10]. We will explain and numerically demonstrate that this approach also allows for stable differentiation. The main reason for this behaviour is that the discontinuity generated by any Monte Carlo path crossing the barrier is avoided by construction.

The main new part of this paper is to extend Glasserman and Staum's one-step survival strategy to the multivariate case. The challenge of this extension is how to sample from a truncated multivariate normal distribution without destroying the stability with respect to differentiation. In particular, as we explain in section 2.2, acception-rejection strategies fail in this task. Our tools to solve this challenge stem from a technique known as GHK Importance Sampling Simulator (named after Geweke, Hajivassiliou and Keane, see [8, 11, 14], and the concise description in Shum [16]). To the knowledge of the authors, this is the first time that the ideas of GHK are being adapted to the pricing of options via Monte Carlo simulations.

Ultimately, we derive a Monte Carlo algorithm for multivariate autocallables that allows stable differentiation by simple finite differences. We also numerically demonstrate that combining finite differences with our new algorithm is more efficient than directly calculating the option's sensitivity with a likelihood ratio method. Moreover, our numerical experiments also show that, compared to standard MC algorithms, the new algorithm possesses a substantially reduced variance. Since no evaluations of multivariate cumulative normal distributions are required, this variance reduction is achieved at the cost of only a moderately increased computation time, which indicates that our new algorithm is also an efficient way for pricing autocallables.

The outline of this work is as follows. In section 2 we derive our new Monte Carlo pricing algorithm for the univariate and the multivariate case, and discuss its stability with respect to differentiation. Then, we numerically demonstrate our algorithms' stability and study its variance reduction properties in section 3. Section 4 contains some concluding remarks.

2. Monte Carlo pricing of uni- and multivariate autocallable options.

In this section, we derive our Monte Carlo pricing algorithm for autocallable options. Based on a combination of Glasserman and Staums' one-step survival techniques [10] and the GHK Importance Sampling [8], we will obtain a substantial variance reduction effect as well as the possibility to stably differentiate the numerically calculated prices by simple finite differences.

2.1. The univariate case.

We start with the univariate case where the autocallable option depends on only one underlying. This subsection is more of an introductory nature as the univariate case can be treated with a straight-forward application of the one-step survival techniques in [10] to our specific type of option. We will, however, also use this subsection to elaborate on the estimator's stability

This is a preprint of an article that has appeared in The Journal of Computational Finance 17 (1), 43-70, 2013.
http://www.risk.net/journal-of-computational-finance/journal/2292302/latest-issue-volume-17-issue-1-2013

properties with respect to differentiation which will be a crucial point in the later generalization to the multivariate case.

Let S_t describe the evolution of the underlying spot price and let $(S_1, \ldots, S_m)$ be a vector containing its evaluations at some fixed, chronologically sorted, observation dates $(t_1, \ldots, t_m)$. We will only consider the Black-Scholes model, i.e., S_t is assumed to follow a geometric Brownian motion,

$$\frac{\mathrm{d}S_t}{S_t} = \mu \mathrm{d}t + \sigma \mathrm{d}W_t,$$

where $\mu := r - b$, r denotes the risk-free interest rate, b is the dividend yield, $\sigma > 0$ is the volatility, and W_t is a standard Brownian motion. The extension of the following results to the case of non-constant, but deterministic, parameters is obvious and it also seems possible to extend them to stochastic parameters, cf. our concluding remarks in section 4.

It is well known that this yields (for $j = 0, \ldots, m-1$)

$$S_{j+1} = S_j \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)(t_{j+1} - t_j) + \sigma\sqrt{t_{j+1} - t_j} Z_j\right), \tag{2.1}$$

with independent standard normally distributed Z_j. t_0 and $S_0 = s_0$ are the current time (w.l.o.g. $t_0 < t_1$) and underlying price.

The discounted payoff of a univariate autocallable option is given by (see again figure 1.1)

$$Q(S_1, \ldots, S_m) \tag{2.2}$$
$$= \begin{cases} e^{-r(t_j - t_0)} Q_j & \text{if} \quad S_i/S_{\mathrm{ref}} < B \leq S_j/S_{\mathrm{ref}} \quad \forall i < j, \\ e^{-r(t_m - t_0)} q(S_m/S_{\mathrm{ref}}) & \text{if} \quad S_j/S_{\mathrm{ref}} < B \quad \forall j = 1, \ldots, m, \end{cases}$$

where Q_j denotes the constant payoff that is paid if the performance S_j/S_{ref} of the underlying at time t_j is, for the first time, greater than the barrier value B on an observation date. The performance is measured relative to some reference value S_{ref} which is, e.g., the spot price at the issue date of the autocallable. If the performance S_j/S_{ref} stays below the barrier for all observation dates, then the holder of the option receives a redemption payoff q depending on the final performance S_m/S_{ref} (see subsection 2.3 for the necessary smoothness conditions of q). Note that all of the following holds (with minor modifications) if the barrier level depends on the observation date.

The value PV_{t_0} of such an autocallable option, at the current time t_0, is given by its expected discounted payoff

$$PV_{t_0} = E\left[Q(S_1, \ldots, S_m)\right].$$

The standard Monte Carlo estimator for PV_{t_0} is calculated by sampling a sequence of possible realizations $(s_{1,n}, \ldots, s_{m,n})$, $n = 1, \ldots, N$, of the random variables $(S_1, \ldots, S_m)$ and approximating PV_{t_0} by the average payoff

$$\frac{1}{N}\sum_{n=1}^{N} Q\left(s_{1,n}, \ldots, s_{m,n}\right). \tag{2.3}$$

The sampling process is easily implemented by starting with the current underlying price s_0, and then using (2.1) to subsequently generate s_1 up to s_m. Obviously,

This is a preprint of an article that has appeared in The Journal of Computational Finance 17 (1), 43-70, 2013.
http://www.risk.net/journal-of-computational-finance/journal/2292302/latest-issue-volume-17-issue-1-2013

in doing so, it is not necessary to calculate s_{j+1} if already $s_j/S_{\text{ref}} \geq B$. Thus, calculating each sample tuple $(s_1, \ldots, s_m)$ using (2.1) requires m or less samples z_j from the standard normally distributed variable Z_j. These samples can be generated by setting $z_j = \Phi^{-1}(u_j)$, where u_j is drawn from a uniform distribution over the interval $(0, 1)$, and Φ is the cumulative standard normal distribution.

The one-step survival technique of Glasserman and Staum [10] applied to the univariate autocallable improves the above standard Monte Carlo scheme by sampling only paths which stay below the barrier B for all observation dates t_j, i.e., paths which do not lead to an early payoff. For our case, this can be achieved by changing the last step in the above description to

$$z_j = \Phi^{-1}(p_j u_j) \quad \text{with} \quad p_j := \Phi\left(\frac{\ln(BS_{\text{ref}}/s_j) - (\mu - \frac{\sigma^2}{2})(t_{j+1} - t_j)}{\sigma\sqrt{t_{j+1} - t_j}}\right), \tag{2.4}$$

which corresponds to sampling the argument $p_j u_j$ from a uniform distribution on the restricted interval $(0, p_j)$.

Note that $p_j = P\left(S_{j+1}/S_{\text{ref}} < B \mid S_j = s_j\right)$ also describes the probability that the underlying will stay below the barrier in the next step. As a consequence, the sampling of the Z_j is done from a truncated univariate standard normal distribution for which

$$Z_j < \frac{\ln(BS_{\text{ref}}/s_j) - (\mu - \frac{\sigma^2}{2})(t_{j+1} - t_j)}{\sigma\sqrt{t_{j+1} - t_j}}. \tag{2.5}$$

Of course, this modification will bias the result unless we correct for the missing barrier hits, which should have happened with a probability of $1 - p_j$. To make up for this fact, an accordingly weighted $(j + 1)$-th premature payoff must be added to the total payoff for this sample tuple, and all further payoffs for this sample must be weighted by p_j. Hence, we replace (2.3) by the estimator

$$\widehat{PV_{t_0}} := \frac{1}{N}\sum_{n=1}^{N} \tilde{Q}\left(s_{1,n}, \ldots, s_{m,n}\right), \tag{2.6}$$

which is subsequently denoted as the one-step survival MC estimator. The tuple $(s_{1,n}, \ldots, s_{m,n})$ is sampled according to the one-step survival strategy explained above,

$$\begin{aligned}
&\tilde{Q}(s_1, \ldots, s_m) \\
&:= (1 - p_0)\, e^{-r(t_1 - t_0)} Q_1 + p_0 \Big[(1 - p_1)\, e^{-r(t_2 - t_0)} Q_2 + p_1 \Big[(1 - p_2) e^{-r(t_3 - t_0)} Q_3 \\
&\qquad + \ \ldots \ + p_{m-2} \Big[(1 - p_{m-1})\, e^{-r(t_m - t_0)} Q_m \ + p_{m-1} e^{-r(t_m - t_0)} q(s_m/S_{\text{ref}})\Big] \ \ldots \Big]\Big] \\
&= L_m e^{-r(t_m - t_0)} q(s_m/S_{\text{ref}}) + \sum_{j=0}^{m-1} L_j\, (1 - p_j)\, e^{-r(t_{j+1} - t_0)} Q_{j+1}
\end{aligned} \tag{2.7}$$

and $L_j := \prod_{i=0}^{j-1} p_i$.We summarize a possible implementation in Algorithm 1.

Mathematically, we can interpret the one-step survival strategy as an integral splitting technique. We have that

$$PV_{t_0}(s_0) = e^{-r(t_1 - t_0)} \int_{\mathbb{R}} \varphi(z) PV_{t_1}(s_1(z))\, \mathrm{d}z, \tag{2.8}$$

Algorithm 1 One-step survival MC estimator for a univariate autocallable option.

```
Initialize random seed
for n = 1, ..., N do
  P_n := 0,   L := 1
  for j = 0, ..., m − 1 do
    p := Φ((ln(B S_ref / s_j) − (μ − σ²/2)(t_{j+1} − t_j)) / (σ √(t_{j+1} − t_j)))
    P_n := P_n + (1 − p) L e^{−r(t_{j+1} − t_0)} Q_{j+1}
    L := pL
    Sample u_j ~ U(0,1)
    s_{j+1} := s_j exp((μ − σ²/2)(t_{j+1} − t_j) + σ √(t_{j+1} − t_j) Φ^{−1}(p u_j))
  end for
  P_n := P_n + L e^{−r(t_m − t_0)} q(s_m / S_ref)
end for
return PV̂_{t_0} := (1/N) Σ_{n=1}^{N} P_n
```

where φ is the standard normal distribution,

$$s_1(z) := s_0 \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)(t_1 - t_0) + \sigma\sqrt{t_1 - t_0}\, z\right),$$

$PV_{t_1}(s_1) = Q_1$ for $s_1 \geq B$, and, for $s_1 < B$, $PV_{t_1}(s_1)$ is given by an analog to (2.8). Splitting the integration domain accordingly yields

$$PV_{t_0}(s_0) = (1 - p_0)e^{-r(t_1 - t_0)}Q_1 + e^{-r(t_1 - t_0)} \int_{s_1(z) < B} \varphi(z) PV_{t_1}(s_1(z))\, \mathrm{d}z$$

with $p_0 := \int_{s_1(z)<B} \varphi(z)\, \mathrm{d}z$ as in (2.4). Since φ is no longer a probability density on the restricted integration domain, we normalize the remaining integral and write

$$PV_{t_0}(s_0) = (1 - p_0)e^{-r(t_1 - t_0)}Q_1 + p_0 e^{-r(t_1 - t_0)} \int_{\mathbb{R}} \frac{\varphi(z)}{p_0} \mathbf{1}_{s_1(z)<B} PV_{t_1}(s_1(z))\, \mathrm{d}z$$

with the new properly normalized probability density $\frac{\varphi(z)}{p_0}\mathbf{1}_{s_1(z)<B}$ corresponding to a sampling step with survival condition as explained above. This change to a new probability density can also be interpreted as a special case of importance sampling, with p_0 being the corresponding likelihood ratio (importance sampling weight). By iteratively splitting the integral expression for PV_{t_1}, and the following observation times until maturity, in an analogous manner, and then applying the Monte Carlo technique to the remaining conditional densities, we obtain the estimator (2.6).

Glasserman and Staum prove in [10] that this estimator is unbiased and demonstrate that it possesses a significantly reduced variance compared to the standard estimator (at the price of only slightly increased computational effort).

Let us now turn to a point that has not been examined in [10]. It is well known that the Black-Scholes model leads to a backward parabolic evolution, which *smoothes out* the discontinuities arising from the autocallable structure. Consequently, the option price PV_{t_0} depends smoothly on the current underlying price s_0.

To investigate the smoothness properties of the estimators in (2.3) and (2.6), let us consider the case of a single sample $N = 1$ (and a fixed random seed). Varying s_0 for the standard MC estimator (2.3) may lead to discontinuities at those points

This is a preprint of an article that has appeared in The Journal of Computational Finance 17 (1), 43-70, 2013.
http://www.risk.net/journal-of-computational-finance/journal/2292302/latest-issue-volume-17-issue-1-2013

s_0, where one of the later values, s_1 to s_m, agrees with the barrier level, since an arbitrarily small change in s_0 may then cause or prevent the premature payoff. The one-step survival estimator (2.6) does not have these discontinuities. All expressions in (2.6) depend smoothly on s_0 and $u_1, \ldots, u_m$. In subsection 2.3 we will show that this smoothness property indeed yields stability with respect to finite differences.

Of course, for a high number of Monte Carlo samples N, also the discontinuities of the standard MC estimator will be smoothed out and it will more and more resemble a smooth function of s_0. However, the process of differentiation amplifies even the smallest discontinuities. In fact, we demonstrate in Section 3 that calculating the derivative with respect to the price of the underlying (the option's Delta) by standard finite differences leads to completely unfeasible results for the standard MC estimator, while giving stable results for the one-step survival estimator.

Let us finish this subsection with some more remarks. First of all, similar arguments as above apply for the other parameters of our autocallable option and consequently to the calculation of other option price sensitivities such as vega (sensitivity w.r.t. the implied volatility). Also, it should be noted that the discontinuity behaviour already occurs in the simple case of determining the sensitivity Delta of a European binary (also called digital) option. Binary options mature with some fixed payoff if the underlying value is above the strike and pay nothing otherwise. Also there, combining finite differences with a standard Monte Carlo pricing routine leads to unfeasibly oscillating results due to the discontinuity of the payoff function. The one-step survival estimator, on the other hand, forces every Monte Carlo path to return zero, and adds the probability weighted payoff, which is already the exact price. Hence, independently of N, it always returns the exact price and thus allows stable differentiation.

Finally, let us comment on alternative ways to obtaining sensitivities of autocallables. First of all, one can differentiate the analytical pricing formulas mentioned in the introduction. This involves computationally expensive evaluations of multivariate cumulative normal distributions, and is also not a very flexible approach, as it is restricted to specific payoffs. On the other hand, it is possible to directly simulate the sensitivities of options also for discontinuous payoffs. The most prominent way to do this is arguably the Likelihood Ratio Method (LRM), cf., e.g., Glasserman [9], which relies on the fact, that a standard Monte Carlo estimator for the price can be turned into an estimator for a sensitivity, by weighting the payoffs of each Monte Carlo path with a special weight. For the sensitivity delta, this weight, $z_1(s_0\sigma\sqrt{t_1 - t_0})^{-1}$, is particularly simple and does not depend on the option's payoff structure at all, which makes the LRM a very flexible and generic method.

However, our numerical results in section 3 show that the combination of finite differences with the one-step survival estimator yields clearly superior results compared to the LRM. Furthermore, from a practical point of view, the finite difference method has the advantage, that sensitivity calculations can be played back to the simulation of the option price. For this reason, it is frequently the method-of-choice in dedicated risk systems used for front- or middle-office activities within financial institutions. Adapting the Monte Carlo pricing routines to allow for stable differentiation eases the integration of new option types in such systems.

2.2. The multivariate case. We will now consider an autocallable option that depends on multiple underlyings. To keep things simple, we restrict ourselves to two underlyings and assume that the premature payoffs depend only on the maximum or the minimum of the underlying prices. Actually, to the knowledge of the authors,

This is a preprint of an article that has appeared in The Journal of Computational Finance 17 (1), 43-70, 2013.
http://www.risk.net/journal-of-computational-finance/journal/2292302/latest-issue-volume-17-issue-1-2013

almost every multivariate autocallable in the German market (the Duo Express Certificates mentioned in the introduction) has its premature payoff depending on the minimum of two underlyings. We nevertheless treat the maximum case first to explain the ideas of our new approach before turning to the more relevant, but also slightly more complicated, minimum case. Let us also stress that the technique herein can also be applied to a higher number of underlyings, more complicated premature payoff conditions, or non-constant parameters in the stochastic model, cf. our concluding remarks in section 4.

Let $(S_t^{(1)}, S_t^{(2)})$ describe the evolution of two underlying spot prices in a Black-Scholes model, i.e., $(S_t^{(1)}, S_t^{(2)})$ solves

$$\frac{\mathrm{d}S_t^{(1)}}{S_t^{(1)}} = \mu_1 \mathrm{d}t + \sigma_1 \mathrm{d}W_t^{(1)} \quad \text{and} \quad \frac{\mathrm{d}S_t^{(2)}}{S_t^{(2)}} = \mu_2 \mathrm{d}t + \sigma_2 \mathrm{d}W_t^{(2)},$$

with $\mu_1, \mu_2 \in \mathbb{R}$, $\sigma_1, \sigma_2 > 0$, and a two-dimensional Brownian motion $(W_t^{(1)}, W_t^{(2)})$ with zero drift and covariance matrix

$$\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}. \tag{2.9}$$

Similar to the univariate case in subsection 2.1, we denote by $(S_1^{(1)}, \ldots, S_m^{(1)})$ and $(S_1^{(2)}, \ldots, S_m^{(2)})$ the vectors containing the underlying prices at some fixed chronologically sorted observation dates $(t_1, \ldots, t_m)$. They satisfy (for $j = 0, \ldots, m-1$)

$$S_{j+1}^{(1)} = S_j^{(1)} \exp\left(\left(\mu_1 - \frac{\sigma_1^2}{2}\right)(t_{j+1} - t_j) + \sigma_1 \sqrt{t_{j+1} - t_j} Z_j^{(1)}\right), \tag{2.10}$$

$$S_{j+1}^{(2)} = S_j^{(2)} \exp\left(\left(\mu_2 - \frac{\sigma_2^2}{2}\right)(t_{j+1} - t_j) + \sigma_2 \sqrt{t_{j+1} - t_j} Z_j^{(2)}\right), \tag{2.11}$$

where $(Z_j^{(1)}, Z_j^{(2)})$ are normally distributed with zero drift, and covariance matrix (2.9). t_0, $S_0^{(1)} = s_0^{(1)}$, and $S_0^{(2)} = s_0^{(2)}$ are the current time (w.l.o.g. $t_0 < t_1$), and underlying prices.

2.2.1. Early payoff depending on the maximum of the underlyings. We first consider a multivariate autocallable option with the following (discounted) payoff

$$Q(S_1^{(1)}, S_1^{(2)}, \ldots) \tag{2.12}$$
$$= \begin{cases} e^{-r(t_j - t_0)} Q_j & \text{if} \quad M_i < B \leq M_j \quad \forall i < j, \\ e^{-r(t_m - t_0)} q(S_m^{(1)}/S_{\mathrm{ref}}^{(1)}, S_m^{(2)}/S_{\mathrm{ref}}^{(2)}) & \text{if} \quad M_j < B \qquad \forall j = 1, \ldots, m, \end{cases}$$

where $M_j := \max\{S_j^{(1)}/S_{\mathrm{ref}}^{(1)}, S_j^{(2)}/S_{\mathrm{ref}}^{(2)}\}$, and Q_j is the constant payoff that is paid if at time t_j the performance of one of the underlyings is for the first time greater than the barrier level B on an observation date. The case where the barrier depends on the minimum of the underlyings poses some subtle additional difficulties and will be treated further below. The redemption payoff q is assumed to be some function of the final underlying prices (again we refer to subsection 2.3 for necessary smoothness conditions on q).

There is an obvious generalization of the Glasserman-Staum survival techniques explained for the one-dimensional case in section 2.1. Starting with the current prices

$(s_0^{(1)}, s_0^{(2)})$ we could subsequently generate samples $(s_j^{(1)}, s_j^{(2)})$ of $(S_j^{(1)}, S_j^{(2)})$ by using (2.10), (2.11), and the additional survival condition that, both, $S_{j+1}^{(1)} < B$ and $S_{j+1}^{(2)} < B$. This boils down to sampling from a *truncated* multivariate normal distribution with

$$Z_j^{(k)} < C_j^{(k)} := \frac{\ln\left(BS_{\text{ref}}^{(k)}/s_j^{(k)}\right) - \left(\mu_k - \frac{\sigma_k^2}{2}\right)(t_{j+1}-t_j)}{\sigma_k\sqrt{(t_{j+1}-t_j)}}, \quad k = 1, 2,$$

and weighting payoffs according to the probability

$$P\left(\max\left\{\frac{S_{j+1}^{(1)}}{S_{\text{ref}}^{(1)}}, \frac{S_{j+1}^{(2)}}{S_{\text{ref}}^{(2)}}\right\} < B \;\middle|\; \left(S_j^{(1)}, S_j^{(2)}\right) = \left(s_j^{(1)}, s_j^{(2)}\right)\right) = \Phi_\rho\left(C_j^{(1)}, C_j^{(2)}\right),$$

where $\Phi_\rho(\cdot,\cdot)$ is the bivariate cumulative normal distribution with correlation ρ.

However, there are two problems with this obvious generalization. The first one is that it is not clear how to sample from a truncated multivariate normal distribution in a way that smoothly depends on all parameters, which is necessary to gain the desired stability with respect to differentiation. In particular, acception-rejection strategies as proposed in [10] do not have this property. The second problem is that this approach requires the evaluation of a bivariate cumulative normal distribution for every observation time and every MC sample, which is computationally expensive. This argument holds even more for the case of autocallables with more than two assets.

Both problems can be solved by applying the ideas of GHK Importance Sampling [8], which enable us to sample one dimension after the other. To explain this approach, we proceed in reverse order of section 2.1. We start with an integral splitting argument, and, afterwards, interpret it as a one-step survival technique. We have that

$$PV_{t_0}(s_0^{(1)}, s_0^{(2)}) = e^{-r(t_1-t_0)} \int_{\mathbb{R}^2} \varphi_\rho(z^{(1)}, z^{(2)}) PV_{t_1}\left(s_1^{(1)}(z^{(1)}), s_1^{(2)}(z^{(2)})\right) \,\mathrm{d}(z^{(1)}, z^{(2)}), \tag{2.13}$$

where

$$s_1^{(k)}(z^{(k)}) = s_0^{(k)} \exp\left(\left(\mu_k - \frac{\sigma_k^2}{2}\right)(t_1 - t_0) + \sigma_k\sqrt{t_1 - t_0}\, z^{(k)}\right), \quad k = 1, 2,$$

φ_ρ is the bivariate normal density with correlation ρ, $PV_{t_1} = Q_1$ for $M(s_1^{(1)}, s_1^{(2)}) \geq B$, and, otherwise, PV_{t_1} is given by an analog to (2.13).

We apply the standard transformation to uncorrelated normal distributions,

$$z^{(1)} = y^{(1)} \quad \text{and} \quad z^{(2)} = \rho y^{(1)} + \sqrt{1-\rho^2}\, y^{(2)}.$$

This yields

$$PV_{t_0}(s_0^{(1)}, s_0^{(2)}) = e^{-r(t_1-t_0)} \int_{\mathbb{R}} \varphi(y^{(1)}) \int_{\mathbb{R}} \varphi(y^{(2)}) PV_{t_1}\left(s_1^{(1)}, s_1^{(2)}\right) \,\mathrm{d}y^{(2)}\,\mathrm{d}y^{(1)}, \tag{2.14}$$

where now (with a slight abuse of notation) $s_1^{(k)} = s_1^{(k)}\left(y^{(1)}, y^{(2)}\right)$, $k = 1, 2$.

The survival condition $M(s_1^{(1)}, s_1^{(2)}) = \max\{s_1^{(1)}/S_{\text{ref}}^{(1)}, s_1^{(2)}/S_{\text{ref}}^{(2)}\} < B$ is equivalent to

$$z^{(1)} < C_1^{(1)} \quad \text{and} \quad z^{(2)} < C_1^{(2)},$$

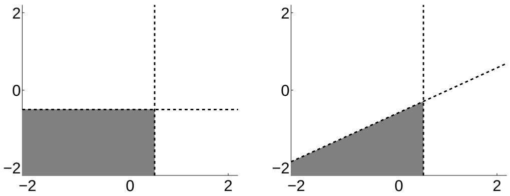

FIG. 2.1. *Survival zone before (left picture) and after (right picture) transformation to uncorrelated variables.*

and thus equivalent to

$$y^{(1)} < C_1^{(1)} \quad \text{and} \quad y^{(2)} < \frac{C_1^{(2)} - \rho y^{(1)}}{\sqrt{1-\rho^2}},$$

see figure 2.1 for a sketch of these survival zones.

We accordingly split the one-dimensional integrals in (2.14) and obtain by an easy calculation

$$PV_{t_0}(s_0^{(1)}, s_0^{(2)}) = e^{-r(t_1-t_0)} \int_{-\infty}^{C_1^{(1)}} \frac{\varphi(y^{(1)})}{p_0^{(1)}} \left\{ \left(1 - p_0^{(1)} p_0^{(2)}\right) Q_1 \right.$$
$$\left. + p_0^{(1)} p_0^{(2)} \int_{-\infty}^{\frac{C_1^{(2)} - \rho y^{(1)}}{\sqrt{1-\rho^2}}} \frac{\varphi(y^{(2)})}{p_0^{(2)}} PV_{t_1}\left(s_1^{(1)}, s_1^{(2)}\right) \, \mathrm{d}y^{(2)} \right\} \mathrm{d}y^{(1)}$$

with

$$p_0^{(1)} := \Phi(C_1^{(1)}), \quad \text{and} \quad p_0^{(2)} := \Phi\left(\frac{C_1^{(2)} - \rho y^{(1)}}{\sqrt{1-\rho^2}}\right).$$

This reformulation now contains two one-dimensional conditional probability densities,

$$\frac{\varphi(y^{(1)})}{p_0^{(1)}} \mathbf{1}_{y^{(1)} < C_1^{(1)}} \quad \text{and} \quad \frac{\varphi(y^{(2)})}{p_0^{(2)}} \mathbf{1}_{y^{(2)} < (C_1^{(2)} - \rho y^{(1)})/\sqrt{1-\rho^2}},$$

each of which is properly normalized, with the particularity that the normalization of the second one depends on the first variable of integration $y^{(1)}$. After treating the integral expression for PV_{t_1} and subsequent observation times in an analogous manner, this reformulation can be implemented as in the univariate case in section 2.1.

This leads to the following multivariate one-step survival strategy: For the $(j+1)$-th observation time, we enforce survival by first sampling the first underlying $s_{j+1}^{(1)}$, such that it is still possible to evade the barrier (by drawing $y^{(1)}$ from an appropriately truncated normal distribution). Then, we sample the second underlying $s_{j+1}^{(2)}$, such that the barrier is indeed evaded (by drawing $y^{(2)}$ from a normal distribution with truncation condition depending on $s_{j+1}^{(1)}$). We account for the missing barrier hit by adding the premature payoff with the weight $1 - p_j^{(1)} p_j^{(2)}$ defined above (which depends on $s_{j+1}^{(1)}$, resp. $y^{(1)}$) and weight all further payoffs with $p_j^{(1)} p_j^{(2)}$.

This is a preprint of an article that has appeared in The Journal of Computational Finance 17 (1), 43-70, 2013.
http://www.risk.net/journal-of-computational-finance/journal/2292302/latest-issue-volume-17-issue-1-2013

We sum up the resulting implementation in algorithm 2. Note that it depends smoothly on all parameters. In section 3, we will investigate an extension of this algorithm numerically and demonstrate that the calculated values can indeed be stably differentiated by simple finite differences.

Algorithm 2 One-step survival GHK-MC estimator for a multivariate autocallable option (barrier on maximum).

Initialize random seed
for $n = 1, \dots, N$ **do**
  $P_n := 0, \quad L := 1$
  for $j = 0, \dots, m-1$ **do**
    $C^{(1)} := \Big(\ln(BS^{(1)}_{\text{ref}}/s^{(1)}_j) - (\mu_1 - \sigma_1^2/2)(t_{j+1} - t_j)\Big) \Big/ \Big(\sigma_1\sqrt{(t_{j+1} - t_j)}\Big)$
    $C^{(2)} := \Big(\ln(BS^{(2)}_{\text{ref}}/s^{(2)}_j) - (\mu_2 - \sigma_2^2/2)(t_{j+1} - t_j)\Big) \Big/ \Big(\sigma_2\sqrt{(t_{j+1} - t_j)}\Big)$
    $p^{(1)} := \Phi(C^{(1)})$
    Sample $u^{(1)} \sim U(0,1)$. $y^{(1)} := \Phi^{-1}(p^{(1)}u^{(1)})$
    $p^{(2)} := \Phi((C^{(2)}_1 - \rho y^{(1)})/\sqrt{1-\rho^2})$
    Sample $u^{(2)} \sim U(0,1)$. $y^{(2)} := \Phi^{-1}(p^{(2)}u^{(2)})$
    $P_n := P_n + (1 - p^{(1)}p^{(2)})Le^{-r(t_{j+1}-t_0)}Q_{j+1}$
    $L := p^{(1)}p^{(2)}L$
    $z^{(1)} := y^{(1)}, \quad z^{(2)} := \rho y^{(1)} + \sqrt{1-\rho^2}y^{(2)}$
    $s^{(1)}_{j+1} := s^{(1)}_j \exp\left(\left(\mu_1 - \frac{\sigma_1^2}{2}\right)(t_{j+1} - t_j) + \sigma_1\sqrt{t_{j+1} - t_j}z^{(1)}\right)$
    $s^{(2)}_{j+1} := s^{(2)}_j \exp\left(\left(\mu_2 - \frac{\sigma_2^2}{2}\right)(t_{j+1} - t_j) + \sigma_2\sqrt{t_{j+1} - t_j}z^{(2)}\right)$
  end for
  $P_n := P_n + Le^{-r(t_m - t_0)}q(s^{(1)}_m/S^{(1)}_{\text{ref}}, s^{(2)}_m/S^{(2)}_{\text{ref}})$
end for
return $\widehat{PV_{t_0}} = \frac{1}{N}\sum_{n=1}^{N} P_n$

Before we turn to the case where the early payoff depends on the minimum of the underlyings, let us stress again that the algorithm developed herein is not the obvious generalization of the survival idea sketched in the beginning of this subsection. Evading the barrier by subsequently sampling first one, and then the other, underlying is not the same as sampling them simultaneously from a truncated bivariate normal distribution. Accordingly, $p^{(1)}_j p^{(2)}_j$ is not the probability of the barrier hit, it is an importance sampling weight depending on the sample of the first underlying. Note that (besides allowing stable differentiation) this approach also has the major advantage that it only requires one-dimensional truncated normal distributions.

2.2.2. Early payoff depending on the minimum of the underlyings. Now we consider the practically more relevant case of a multivariate autocallable option, where the premature payoff is paid when both of the underlyings are larger than the barrier level, i.e., an option with the (discounted) payoff profile (2.12) and $M_j := \min(S^{(1)}_j/S^{(1)}_{\text{ref}}, S^{(2)}_j/S^{(2)}_{\text{ref}})$. Now the survival zone, i.e., the region in which no premature payoff happens, is no longer rectangular but L-shaped, see the left side of figure 2.2.

Proceeding along the lines of subsection 2.2.1 leads to the sampling of the first underlying with no additional condition (as it is always still possible to evade the barrier). Depending on this sample, the second underlying is then sampled either

This is a preprint of an article that has appeared in The Journal of Computational Finance 17 (1), 43-70, 2013.
http://www.risk.net/journal-of-computational-finance/journal/2292302/latest-issue-volume-17-issue-1-2013

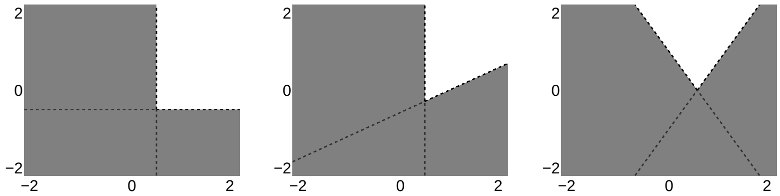

Fig. 2.2. *Survival zone before (left picture) and after (middle picture) transformation to uncorrelated variables, and after an additional rotation step (right picture).*

with no additional condition (if the first underlying is under the barrier), or with the additional condition that it has to stay below the barrier. More precisely, with the notation from subsection 2.2.1, we have to draw $y^{(1)}$ from a standard normal distribution and then draw $y^{(2)}$ from either an non-truncated standard normal distribution (if $y^{(1)} < C^{(1)}$), or (otherwise) under the additional condition $y^{(2)} < \frac{C_1^{(2)} - \rho y^{(1)}}{\sqrt{1-\rho^2}}$, see the middle picture in figure 2.2 for a sketch of this domain.

This means that, depending on $y^{(1)}$, the truncation condition for $y^{(2)}$ abruptly becomes active. The samples do no longer depend smoothly on the parameters that control the domain bounds, and so the resulting algorithm will no longer have the desired stability with respect to differentiation.

To alleviate this problem and obtain a (Lipschitz) continuous parameterization of the bounds for the second sample with respect to the first one, we introduce an additional rotation step and set

$$\begin{pmatrix} y^{(1)} \\ y^{(2)} \end{pmatrix} := \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x^{(1)} \\ x^{(2)} \end{pmatrix}.$$

The survival condition $M(s_1^{(1)}, s_1^{(2)}) = \min(s_1^{(1)}, s_1^{(2)}) < B$ is then equivalent to

$$x^{(2)} < \max\left\{ \frac{C_1^{(1)} - x^{(1)}\cos\alpha}{\sin\alpha}, \frac{C_1^{(2)} - \rho x^{(1)}\cos\alpha + \sqrt{1-\rho^2}x^{(1)}\sin\alpha}{\rho\sin\alpha + \sqrt{1-\rho^2}\cos\alpha} \right\} =: C_1 \tag{2.15}$$

see the right side of figure 2.2 for a sketch of this domain. A simple choice for α that makes the right hand side of (2.15) a (Lipschitz) continuous function for all parameters is to take half the angle between the two bounding lines,

$$\alpha := \frac{1}{2}\left(\frac{\pi}{2} - \arctan\left(-\frac{\rho}{\sqrt{1-\rho^2}}\right)\right).$$

With this additional rotation step we proceed as in subsection 2.2.1, and obtain

$$\begin{aligned} PV_{t_0}(s_0^{(1)}, s_0^{(2)}) &= e^{-r(t_1-t_0)} \int_{\mathbb{R}^2} \varphi_\rho(z^{(1)}, z^{(2)}) PV_{t_1}\left(s_1^{(1)}(z^{(1)}), s_1^{(2)}(z^{(2)})\right) \, \mathrm{d}(z^{(1)}, z^{(2)}) \\ &= e^{-r(t_1-t_0)} \int_{\mathbb{R}} \varphi(x^{(1)}) \int_{\mathbb{R}} \varphi(x^{(2)}) PV_{t_1}\left(s_1^{(1)}, s_1^{(2)}\right) \, \mathrm{d}x^{(2)} \, \mathrm{d}x^{(1)} \\ &= e^{-r(t_1-t_0)} \int_{\mathbb{R}} \varphi(x^{(1)}) \Bigg\{ (1-p_0)\, Q_1 \\ &\qquad + p_0 \int_{-\infty}^{C_1} \frac{\varphi(x^{(2)})}{\Phi(C_1)} PV_{t_1}\left(s_1^{(1)}, s_1^{(2)}\right) \, \mathrm{d}x^{(2)} \Bigg\} \, \mathrm{d}x^{(1)}, \end{aligned} \tag{2.16}$$

with $p_0 := \Phi(C_1)$.

Again, this reformulation now contains only two one-dimensional conditional probability densities and can be implemented as before. We sum up the resulting implementation in algorithm 3. Note that, due to the additional rotation, the resulting estimator is now a (Lipschitz) continuous, but not necessarily differentiable, function of the parameters. In the next subsection we will show that this is sufficient to allow stable first order differentiation by finite differences. In order to gain stability also for higher order derivatives one may consider replacing the non-differentiable function $\max(\cdot,\cdot)$ in the definition of C by a smoothed approximate version, or splitting also the outer integral in (2.16).

Algorithm 3 One-step survival GHK-MC estimator for a multivariate autocallable option (barrier on minimum).

Initialize random seed
for $n = 1, \dots, N$ **do**
  $P_n := 0, \quad L := 1$
  for $j = 0, \dots, m-1$ **do**
    $C^{(1)} := \left(\ln(BS_{\text{ref}}^{(1)}/s_j^{(1)}) - (\mu_1 - \sigma_1^2/2)(t_{j+1}-t_j)\right) \Big/ \left(\sigma_1\sqrt{(t_{j+1}-t_j)}\right)$
    $C^{(2)} := \left(\ln(BS_{\text{ref}}^{(2)}/s_j^{(2)}) - (\mu_2 - \sigma_2^2/2)(t_{j+1}-t_j)\right) \Big/ \left(\sigma_2\sqrt{(t_{j+1}-t_j)}\right)$
    Sample $u^{(1)} \sim U(0,1)$. $x^{(1)} := \Phi^{-1}(u^{(1)})$
    $C := \max\left\{ \frac{C^{(1)} - x^{(1)}\cos\alpha}{\sin\alpha}, \frac{C^{(2)} - \rho x^{(1)}\cos\alpha + \sqrt{1-\rho^2}x^{(1)}\sin\alpha}{\rho\sin\alpha + \sqrt{1-\rho^2}\cos\alpha} \right\}$
    $p := \Phi(C)$
    $P_n := P_n + (1-p)Le^{-r(t_{j+1}-t_0)}Q_{j+1}$
    $L := pL$
    Sample $u^{(2)} \sim U(0,1)$. $x^{(2)} := \Phi^{-1}(pu^{(2)})$
    $y^{(1)} := x^{(1)}\cos\alpha + x^{(2)}\sin\alpha, \quad y^{(2)} := -x^{(1)}\sin\alpha + x^{(2)}\cos\alpha$
    $z^{(1)} := y^{(1)}$, $z^{(2)} := \rho y^{(1)} + \sqrt{1-\rho^2}y^{(2)}$
    $s_{j+1}^{(1)} := s_j^{(1)} \exp\left(\left(\mu_1 - \sigma_1^2/2\right)(t_{j+1}-t_j) + \sigma_1\sqrt{t_{j+1}-t_j}\,z^{(1)}\right)$
    $s_{j+1}^{(2)} := s_j^{(2)} \exp\left(\left(\mu_2 - \sigma_2^2/2\right)(t_{j+1}-t_j) + \sigma_2\sqrt{t_{j+1}-t_j}\,z^{(2)}\right)$
  end for
  $P_n := P_n + Le^{-r(t_m-t_0)}q(s_m^{(1)}/S_{\text{ref}}^{(1)}, s_m^{(2)}/S_{\text{ref}}^{(1)})$
end for
return $\widehat{PV_{t_0}} = \frac{1}{N}\sum_{n=1}^{N} P_n$

This is a preprint of an article that has appeared in The Journal of Computational Finance 17 (1), 43-70, 2013.
http://www.risk.net/journal-of-computational-finance/journal/2292302/latest-issue-volume-17-issue-1-2013

2.3. Smoothness and stability: the mathematical background. Intuitively, it seems clear that the Monte Carlo estimators should smoothly depend on the parameters in order to allow stable differentiation (cf. the end of subsection 2.1). We now briefly describe the mathematical background of this connection between *smoothness* and *stability*, and summarize the stability properties of our algorithms. A more in-depth discussion on the stability of calculating sensitivities can be found, e.g., in the monograph of Glasserman [9].

To keep the mathematical formalism to a minimum, let us first observe that, in all our cases, the stochastic process of the underlyings(s) can be described by some parameters, and a number of independently uniformly distributed random variables on the unit interval. The payoff function then depends on the prices of the underlying(s) at the observation times and some more parameters. Now assume that we wish to differentiate with respect to one parameter and keep all others fixed. To ease presentation, we concentrate on the current underlying price s_0; all other parameters can be treated completely analogously.

Then, the true value PV_{t_0}, and its Monte Carlo estimator $\widehat{PV}_{t_0}$, can be written as a parameter-dependent integral (resp., sum)

$$PV_{t_0}(s_0) := \int_{(0,1)^M} \mathcal{Q}(s_0, u)\ \mathrm{d}u \approx \frac{1}{N} \sum_{n=1}^{N} \mathcal{Q}(s_0, u_{\cdot,n}) =: \widehat{PV}_{t_0}(s_0), \tag{2.17}$$

where $\mathcal{Q}(s_0, u)$ denotes the (discounted) payoff according to the parameter value s_0 and random variable values $u \in (0,1)^M$, N is the number of samples, $u_{\cdot,n} = (u_{1,n}, \ldots, u_{M,n})$ are random draws from $U(0,1)^M$, and $\mathcal{Q}(s_0, u_{\cdot,n})$ is the (discounted) Monte Carlo payoff according to this draw.

To be more precise, for the univariate case, the standard Monte Carlo estimator can be written in the above form by setting

$$\mathcal{Q}^{\mathrm{uni}}_{\mathrm{standard}}(s_0, u) := Q(s_1(s_0, u), \ldots, s_m(s_0, u)), \qquad u = (u_1, \ldots, u_m), \tag{2.18}$$

with Q given by (2.2), $M = m$, and the $s_j(s_0, u)$ are recursively defined by using (2.1) with $z_j = \Phi^{-1}(u_j)$.

The one-step survival estimator from section 2.1 can be written in the same form by setting

$$\mathcal{Q}^{\mathrm{uni}}_{\mathrm{oss}}(s_0, u) := \tilde{Q}(\tilde{s}_1(s_0, u), \ldots, \tilde{s}_m(s_0, u)), \qquad u = (u_1, \ldots, u_m), \tag{2.19}$$

where now $\tilde{Q}$ from (2.7) is used, $M = m$, and the $\tilde{s}_j(s_0, u)$ are recursively defined by using (2.1) with $z_j = \Phi^{-1}(p_j u_j)$ and p_j defined by (2.4). By the integral splitting argument in section 2.1, it immediately follows that

$$PV^{\mathrm{uni}}_{t_0}(s_0) = \int_{(0,1)^M} \mathcal{Q}^{\mathrm{uni}}_{\mathrm{standard}}(s_0, u)\ \mathrm{d}u = \int_{(0,1)^M} \mathcal{Q}^{\mathrm{uni}}_{\mathrm{oss}}(s_0, u)\ \mathrm{d}u,$$

so that the one-step survival estimator is unbiased by construction, cf. also Glasserman and Staum [10].

Analogously, the standard and the one-step survival estimators for the bivariate case with payoff depending on the minimum, resp., maximum from section 2.2 can be written in the above form, yielding

$$PV^{\max}_{t_0}(s_0) = \int_{(0,1)^M} \mathcal{Q}^{\max}_{\mathrm{standard}}(s_0, u)\ \mathrm{d}u = \int_{(0,1)^M} \mathcal{Q}^{\max}_{\mathrm{oss}}(s_0, u)\ \mathrm{d}u,$$

This is a preprint of an article that has appeared in The Journal of Computational Finance 17 (1), 43-70, 2013.
http://www.risk.net/journal-of-computational-finance/journal/2292302/latest-issue-volume-17-issue-1-2013

resp.,

$$PV_{t_0}^{\min}(s_0) = \int_{(0,1)^M} \mathcal{Q}_{\text{standard}}^{\min}(s_0, u)\ \mathrm{d}u = \int_{(0,1)^M} \mathcal{Q}_{\text{oss}}^{\min}(s_0, u)\ \mathrm{d}u.$$

Here, $M = 2m$, the definitions of $\mathcal{Q}_{\text{standard}}^{\max}(s_0, u)$ and $\mathcal{Q}_{\text{standard}}^{\min}(s_0, u)$ are obvious, and the form of $\mathcal{Q}_{\text{oss}}^{\max}(s_0, u)$ and $\mathcal{Q}_{\text{oss}}^{\min}(s_0, u)$ is most easily derived from algorithm 2 and algorithm 3.

To study the stability of Monte Carlo estimators in the general form (2.17), let us introduce the forward finite difference operator D_h, that operates with respect to the parameter s_0,

$$D_h PV_{t_0}(s_0) = \frac{1}{h}\left(PV_{t_0}(s_0 + h) - PV_{t_0}(s_0)\right).$$

All of the following is easily extended to the case of backward or central finite differences. Now we can give a precise definition of stability.

Definition 2.1. *We say that a Monte Carlo estimator allows stable differentiation by finite differences if there exists $C(h) > 0$ such that*

$$\mathrm{Var}(D_h \widehat{PV}_{t_0}(s_0)) \le \frac{1}{N} C(h),$$

and $C(h)$ stays bounded for $h \to 0$.

Since $D_h PV_{t_0}(s_0) \to \frac{\partial}{\partial s_0} PV_{t_0}(s_0)$ for $h \to 0$, and since, by linearity, $D_h \widehat{PV}_{t_0}(s_0)$ is an unbiased estimator for $D_h PV_{t_0}(s_0)$, stability means that applying finite differences to the estimator will yield a good approximation for the true sensitivity

$$D_h \widehat{PV}_{t_0}(s_0) \approx \frac{\partial}{\partial s_0} PV_{t_0}(s_0),$$

whenever h is small and N is large. If the estimator does not allow stable differentiation, i.e., $C(h) \to \infty$, then, generally, for each fixed N, the error of the differentiated Monte Carlo estimator will blow up for $h \to 0$. In this case, one can still obtain convergence by adequately balancing N and h, but more accurate finite differences will then require increasingly infeasibly high numbers of Monte Carlo samples.

Theorem 2.2. *If both, $PV_{t_0}(s_0)$ and the Monte Carlo payoff $\mathcal{Q}(s_0, u)$, depend Lipschitz continuously on s_0, resp., (s_0, u), then the estimator allows stable differentiation.*

Proof. For an estimator in the form (2.17) and $U \sim U(0,1)^M$, we have that

$$\begin{aligned} \mathrm{Var}\left(D_h \widehat{PV}_{t_0}(s_0)\right) &= \frac{1}{N}\mathrm{Var}\left(D_h \mathcal{Q}(s_0, U)\right) \\ &= \frac{1}{N}\int_{(0,1)^M} \left(D_h \mathcal{Q}(s_0, u) - D_h PV_{t_0}(s_0)\right)^2\ \mathrm{d}u, \end{aligned} \tag{2.20}$$

When both, $PV_{t_0}(s_0)$ and $\mathcal{Q}(s_0, u)$, are Lipschitz continuous, then the right hand side of (2.20) is bounded by an h-independent constant which completes the proof. □

In the preceeding subsections, we demonstrated how to switch from a discontinuous Monte Carlo payoff to one that is (at least) Lipschitz continuous. For the univariate autocallable, the standard Monte Carlo payoff $\mathcal{Q}_{\text{standard}}^{\text{uni}}(s_0, u)$ defined in (2.18) possesses the discontinuities of the autocallable's original payoff function Q given in (2.2). On the other hand, for the one-step survival estimator, it is easily checked by analyzing algorithm 1, that the dependance of the Monte Carlo payoff $\mathcal{Q}_{\text{oss}}^{\text{uni}}(s_0, u)$ from (s_0, u) is a composition of infinitely often differentiable functions and the redemption payoff function q.

The same holds for the bivariate autocallable in the case where the payoff depends on the maximum of the underlyings, treated in subsection 2.2.1. We implement the survival idea by subsequently sampling first one and then the other underlying. In doing so, the bound on the second underlying depends smoothly on the sample of the first one, cf. the survival zone in the right graph of figure 2.1. This yields that the samples depend smoothly on all parameters, which is a general feature of the underlying GHK Importance Sampling technique for convex polyhedrons, cf. [3]. Hence, we obtain by analyzing algorithm 2 that, again, the dependance of the Monte Carlo payoff from (s_0, u) is a composition of infinitely often differentiable functions and the redemption payoff function q.

In the last case, where the payoff depends on the minimum of the underlyings (cf. subsection 2.2.2), the survival zone is no longer convex but L-shaped, cf. 2.2, which destroys the smoothness properties of the GHK samples. Nevertheless, with our additional rotation step, we are able to ensure that the bound on the sampled second underlying does depend Lipschitz continuously on the sample of the first one, cf. figure 2.2. As a consequence, our estimator in algorithm 3 consists of a composition of infinitely differentiable functions, the non-differentiable but Lipschitz continuous maximum function, and the redemption payoff. Hence, in all three settings, we obtain from theorem 2.2:

COROLLARY 2.3. *If the redemption payoff q is Lipschitz continuous, then the estimators in algorithm 1–3 allow stable differentiation.*

Note that, by the above arguments, a higher regularity of the redemption payoff will lead to a higher regularity for the estimators in algorithm 1 and 2, thus allowing also higher order stable differentiation. For the last algorithm 3 this is not the case due to the presence of the non-differentiable maximum function. Furthermore note that also a piecewise smooth redemption payoff function can be handled in a smooth way by applying the very same integral splitting, resp., importance sampling methods that we used for the barriers.

3. Numerical Results. We now provide some numerical examples for the algorithms developed in section 2. Let us first consider a simple univariate autocallable with only one observation date before maturity. We used the parameters in table 3.1 as default values and, in each of the following plots, varied one of these parameters. The example is fictitious but typical for univariate autocallables traded in the German market.

The first column in figure 3.1 shows the estimated value of the autocallable as a function of the underlying price. The results of the standard Monte Carlo estimator and the one-step survival Monte Carlo estimator described in algorithm 1 are plotted in a black dashed line, and in a gray solid line, respectively. The rows use $N = 10^2$, $N = 10^3$, $N = 10^4$, and $N = 10^5$ Monte Carlo samples from top to bottom for our new algorithm (see below for the standard algorithm), and, for each of these and the following calculations, the same Monte Carlo random seed was used.

The second column shows the first derivative of the autocallable value with respect to the underlying price (the Delta) calculated by applying forward finite differences

$$\frac{\widehat{PV}_{t_0}(s_0 + \delta s) - \widehat{PV}_{t_0}(s_0)}{\delta s} \quad \text{with } \delta s = 10.$$

TABLE 3.1
Parameters of the univariate and the multivariate example.

	Univariate	Multivariate		
Current time:	$t_0 = 0$	$t_0 = 0$		
Current underlying price(s):	$s_0 = 3500$	$s_0^{(1)} = 3500$	$s_0^{(2)} = 7000$	
Reference value(s):	$S_{\text{ref}} = 4000$	$S_{\text{ref}}^{(1)} = 4000$	$S_{\text{ref}}^{(2)} = 8000$	
Barrier:	$B = 1$	$B = 1$		
No. of observation dates:	$m = 2$	$m = 5$		
Observation dates:	$t_1 = 1$	$t_1 = 1$	$t_3 = 3$	$t_5 = 5$
	$t_2 = 2$	$t_2 = 2$	$t_4 = 4$	
Premature payoffs:	$Q_1 = 110$	$Q_1 = 110$	$Q_3 = 130$	$Q_5 = 150$
	$Q_2 = 120$	$Q_2 = 120$	$Q_4 = 140$	
Redemption payoff:	$q(s) = 100s$	$q(s^{(1)}, s^{(2)}) = 100 \min\{s^{(1)}, s^{(2)}\}$		
Risk-free interest rate:	$r = 4\%$	$r = 4\%$		
Dividend yield(s):	$b = 0$	$b_1 = 0$	$b_2 = 0$	
Volatility(ies) and correlation	$\sigma = 30\%$	$\sigma_1 = 30\%$	$\sigma_2 = 40\%$	$\rho = 0.5$

The third and fourth columns show analog plots for the autocallable's value as a function of the volatility, and for its derivative (the Vega) calculated with forward finite differences using $\delta\sigma = 5 \cdot 10^{-4}$.

The gray solid line in figure 3.2 shows again the Delta calculated by applying forward differences to our one-step survival algorithm and compares it to the results of the Likelihood Ratio Method (black dashed line), cf. the end of subsection 2.1.

Each Monte Carlo sample in algorithm 1 requires slightly more time (less than 10% in the average for this numerical experiment) compared to the standard Monte Carlo algorithm, and evaluating the finite differences requires estimating two prices. Hence, figures 3.1 and 3.2 are *runtime-adjusted*: we used $1.10N$ samples for the standard Monte Carlo algorithm and $2.20N$ samples for the Likelihood Ratio Method, so that our new algorithm did not have a longer runtime than the others. Of course, these factors introduce some ambiguity. They depend on our specific implementations (which are, however, quite similar for all the considered algorithms). Moreover, in the standard Monte Carlo algorithm, it is not necessary to continue paths after a barrier hit, so that the runtime is greatly affected by the number of paths with premature payoff. Nevertheless, we believe that these average factors allow a somewhat fair comparison.

The plots clearly demonstrate the instability of the standard Monte Carlo estimator with respect to numerical differentiation and the stability of the one-step survival MC estimator as explained in subsection 2.1. They also clearly show that combining finite differences with one-step survival Monte Carlo estimator yields superior results when compared to the Likelihood Ratio Method.

Our second example is a multivariate autocallable with barrier depending on the minimum of two underlyings, cf. subsection 2.2.2. The parameters are summarized in table 3.1. Again, we used a fictitious but typical example. Note that the redemption payoff is Lipschitz continuous, but not differentiable.

The first, resp., second column in Figure 3.3 shows the value of the autocallable as a function of the two underlying prices calculated with a standard Monte Carlo procedure, resp., with our new one-step survival GHK-MC estimator, cf. algorithm 3. The third and the fourth row show the derivative with respect to the first underlying's price calculated by finite differences with $\delta s^{(1)} = 10$. Again, the plots are runtime-

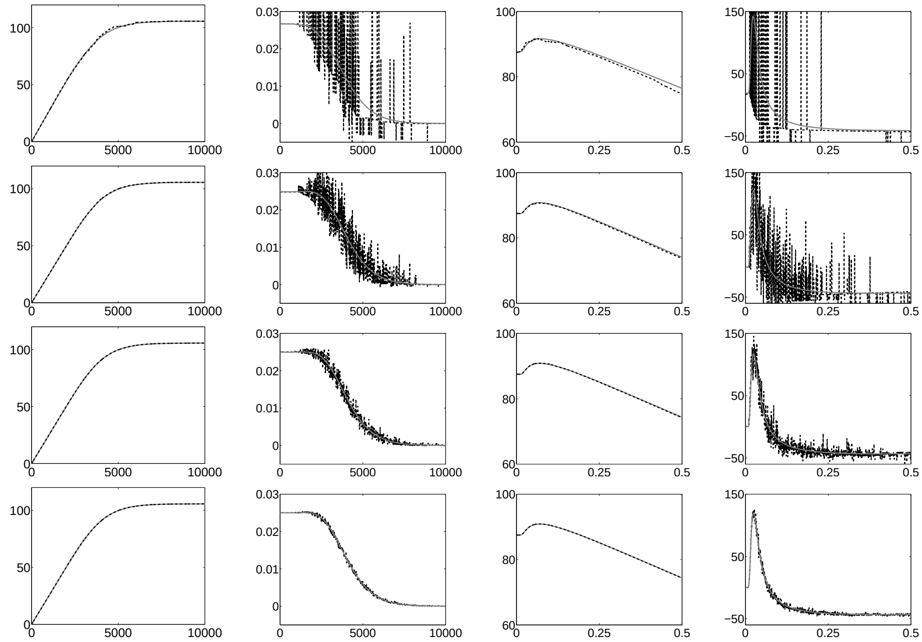

Fig. 3.1. *Value of the autocallable calculated with standard Monte Carlo (black dashed line) and with the one-step survival estimator (gray solid line) as a function of the underlying's price (first column) and volatility (third column) using $N = 10^2$ to $N = 10^5$ Monte Carlo samples (from top to bottom) for the one-step survival estimator. Second and fourth columns show the respective first derivatives calculated by finite differences.*

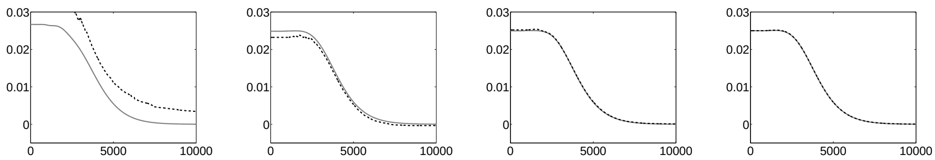

Fig. 3.2. *Delta of the autocallable calculated with the Likelihood Ratio Method (black dashed line) and by applying finite differences on the one-step survival estimator (gray solid line) using $N = 10^2$ to $N = 10^5$ Monte Carlo samples for the one-step survival estimator.*

adjusted. We observed that the Monte Carlos samples in algorithm 3 required (in the average) less than 50% more time than in the standard Monte Carlo algorithm, and accordingly used $N = 10^2$ and $N = 10^4$ samples for algorithm 3, and $1.5N$ samples for the standard method. The instability of the standard Monte Carlo estimator with respect to numerical differentiation and the stability of our new estimator is clearly visible.

The stability of our new algorithm with respect to other derivatives is demonstrated in Figure 3.4. From left to right it shows (as a function of the underlying prices): the derivative of the multivariate autocallable value with respect to the volatility of the first underlying, their correlation, the current time, and the risk-free interest rate. The derivatives were calculated by applying algorithm 3 with $N = 500$ Monte Carlo samples, and using forward finite differences with $\delta\sigma_1 = 0.03\%$, $\delta\rho = 5 \cdot 10^{-4}$, $\delta t_0 = 10^{-3}$, and $\delta r = 4 \cdot 10^{-3}\%$. All plots show smooth functions with no signs of oscillations that would indicate instability.

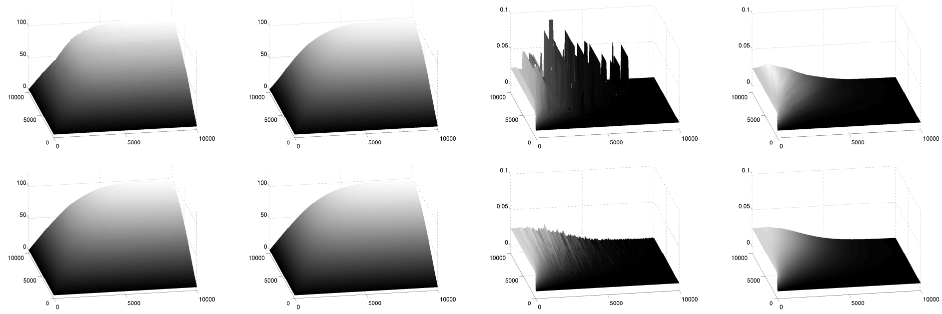

FIG. 3.3. *Value of a multivariate autocallable calculated with standard Monte Carlo (first column) and with the new one-step survival GHK-MC estimator in algorithm 3 (second column) as a function of the two underlying prices for* $N = 10^2$ *(upper row) and* $N = 10^4$ *Monte Carlo samples. Third and fourth column show the respective derivatives with respect to the first underlying price calculated by finite differences.*

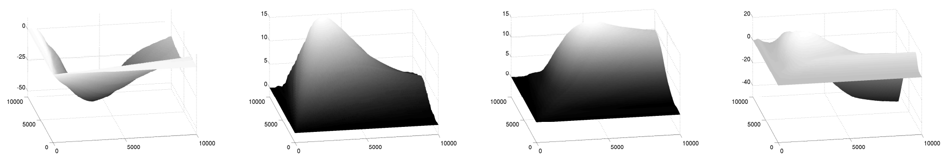

FIG. 3.4. *From left to right: First derivative with respect to the volatility of the first underlying, their correlation, the current time and the risk-free interest rate, calculated with the new one-step survival GHK-MC estimator in algorithm 3 with* $N = 500$ *MC samples and finite differences.*

We end this section with a quantitative study on the variance reduction effect of our new algorithm. Table 3.2 shows the mean value, and the variance of the Monte Carlo samples for the value of the multivariate autocallable, and for its derivative with respect to the first underlying price (calculated by finite differences as above). We used the parameters in table 3.1 and compared the standard Monte Carlo algorithm with our new one-step survival GHK-MC estimator in algorithm 3. The table shows that our new algorithm reduces the variance of the Monte Carlo samples by a factor of three for the autocallable's value and by a factor of more than 500 for the derivative. Consequently, the accuracy of our new algorithm with N samples is comparable to that of the standard algorithm with $3N$, resp., $500N$, samples.

Again, this clearly demonstrates the new algorithm's stability with respect to numerical differentiation. As mentioned above, our new algorithm required, for this multivariate example, on average less than 50% more time than the standard Monte Carlo algorithm. Hence, it requires only half the time to obtain the autocallable's value with the same precision.

At this point, let us stress again, that heuristic averaged runtime factors have to be treated with caution, since the runtime of standard Monte Carlo algorithms for autocallables is greatly affected by prematurely ending paths. Obviously, by taking a large number of barriers and choosing the current underlying price so high, that almost all paths hit the first barrier, one can construct examples where the standard Monte Carlo algorithm is the most efficient one. However, for practically relevant applications, our numerical examples indicate that the one-step survival GHK-MC estimator does not only allow stable differentiation but it is also an efficient way of pricing the autocallable.

TABLE 3.2
Quantitative study of the variance reduction effect.

		$\widehat{PV}_{t_0}$		$\frac{\widehat{PV}_{t_0}(s_0^{(1)}+\delta s^{(1)})-\widehat{PV}_{t_0}(s_0^{(1)})}{\delta s^{(1)}}$	
		MC	Algo. 3	MC	Algo. 3
$N = 10^1$	mean	73.10	66.87	$1.130 \cdot 10^{-3}$	$9.735 \cdot 10^{-3}$
	variance	2240	629.4	$1.277 \cdot 10^{-5}$	$4.641 \cdot 10^{-5}$
$N = 10^2$	mean	74.96	70.53	$2.303 \cdot 10^{-3}$	$9.664 \cdot 10^{-3}$
	variance	1567	499.3	$2.558 \cdot 10^{-5}$	$9.163 \cdot 10^{-5}$
$N = 10^3$	mean	68.89	66.34	$-1.259 \cdot 10^{-3}$	$9.823 \cdot 10^{-3}$
	variance	1584	557.3	$640.2 \cdot 10^{-5}$	$8.658 \cdot 10^{-5}$
$N = 10^4$	mean	67.72	67.74	$11.39 \cdot 10^{-3}$	$9.683 \cdot 10^{-3}$
	variance	1555	538.8	$5035 \cdot 10^{-5}$	$8.472 \cdot 10^{-5}$
$N = 10^5$	mean	67.37	67.60	$9.167 \cdot 10^{-3}$	$9.712 \cdot 10^{-3}$
	variance	1576	542.7	$4534 \cdot 10^{-5}$	$8.573 \cdot 10^{-5}$
$N = 10^6$	mean	67.54	67.57	$9.874 \cdot 10^{-3}$	$9.719 \cdot 10^{-3}$
	variance	1581	538.8	$4801 \cdot 10^{-5}$	$8.559 \cdot 10^{-5}$

4. Conclusions. We adapted the idea of one-step survival Monte Carlo simulations suggested by Glasserman and Staum [10] to the pricing of uni- and multivariate autocallable options. For the univariate case, this application is straightforward. It leads to the variance reduction already discussed in [10], and, in addition, it has the advantage that the calculated prices can be stably differentiated by simple finite differences. From a practical point of view, this feature is particularly important for calculating the risk of such options having only their prices at ones disposal.

The main aim of this paper was to demonstrate how to generalize the idea of one-step survival to the multivariate case. Straightforward implementations using acception-rejection strategies destroy the stability property. Using a new combination of GHK importance sampling with one-step survival, we were able to overcome this problem, and derived a multivariate pricing algorithm that allows for stable differentiation. Our new algorithm does not require evaluating any multivariate normal distributions, but merely uses repeated draws from, properly truncated, univariate standard normal distributions. Numerically, we observed a significant variance reduction at the price of only a moderately increased computation time, which makes the algorithm also an efficient way for pricing autocallables.

To simplify the presentation and concentrate on the main ideas, we have described our new algorithm only for uni- and bivariate autocallables within the classical Black-Scholes setting with constant parameters. The extension to the case where the underlyings follow an Itô-process with non-constant (but deterministic) parameters is obvious: By using smaller time-steps for the Monte Carlo simulation, we can assume that the parameters are constant in each step, and then apply our combination of GHK sampling and one-step survival for those time steps, in which a barrier hit is possible.

Stochastic parameters will lead to more complicated survival zones (where now also the stochastic parameters play a role), but the ideas presented herein should be applicable as well. The same is true for higher numbers of underlyings and more complicated barrier conditions. In principle, this can be done along the lines outlined in this work. However, care has to be taken to parameterize the survival zones in such

This is a preprint of an article that has appeared in The Journal of Computational Finance 17 (1), 43-70, 2013.
http://www.risk.net/journal-of-computational-finance/journal/2292302/latest-issue-volume-17-issue-1-2013

a way that the samples depend smoothly on the parameters. We also believe that the integral-splitting, resp., survival sampling, techniques presented in this work can be adapted to other kinds of options with path-dependent, discontinuous payoffs.

Another possible extension of the method is to go beyond multivariate normal increments, and use, e.g., multivariate student-t-increments. A corresponding generalization of the GHK-algorithm to t-distributions is given in [8].

Finally, it should be mentioned that our approach can be combined with other variance reduction methods, such as control variates or antithetic variables [10].

REFERENCES

[1] T. Alm and A. Wegner: Strukturierte Produkte im Vermögensmanagement, in *Vertrauenskrise Wealth Manager, was nun?*, Gabler 2008.

[2] F. Black and M. Scholes: The pricing of options and corporate liabilities, *Journal of Political Economy* **81**(3), 637–654, 1973.

[3] A. Börsch-Supan and V. Hajivassiliou: Smooth Unbiased Multivariate Probability Simulators for Maximum Likelihood Estimation of Limited Dependent Variable Models, *Journal of Econometrics* **58(3)**, 347-368, 1993.

[4] P. P. Boyle: Options: A Monte Carlo Approach, *Journal of Financial Economics* **4**(3), 323–338, 1977.

[5] M. Bouzoubaa and A. Osseiran: *Exotic Options and Hybrids*, Wiley Finance, John Wiley & Sons, 2010.

[6] H. W. Engl, M. Hanke and A. Neubauer: *Regularization of inverse problems.* Mathematics and its Applications, 375. Kluwer Academic Publishers Group, Dordrecht, 1996.

[7] A. Genz and F. Bretz: *Computation of Multivariate Normal and t Probabilities*, Springer, Berlin Heidelberg, 2010.

[8] J. Geweke: Efficient Simulation from the Multivariate Normal and Student t-Distributions Subject to Linear Constraints, *Computing Science and Statistics: Proceedings of the 23rd symposium on the interface*, 571–578, 1991.

[9] P. Glasserman: *Monte Carlo Methods in financial Engineering*, vol. 53 of Stochastic Modelling and Applied Probability, Springer 2003.

[10] P. Glasserman and J. Staum: Conditioning on one-step survival for barrier option simulations, *Operations Research* **49**(6), 923–937, 2001.

[11] V. Hajivassiliou and D. McFadden: The method of Simulated Scores with applications to Models of External Debt Crises. *Cowles Foundation Discussion Paper No. 967*, 1990.

[12] M. Hanke and O. Scherzer: Inverse problems light: numerical differentiation. *Amer. Math. Monthly* **108**(6), 512–521, 2001.

[13] J.C. Hull: *Options, Futures and other Derivatives*, 7th edition, Prentice Hall, 2008.

[14] M. Keane: A Computational Efficient Practical Simulation Estimator for Panel Data, with Applications to Estimating Temporal Dependence in Employment and Wages, *University of Minnesota*, mimeo, 1990.

[15] R. A. Reder: Auto Trigger Securities: Closed-form Solutions and Applications, *Diplomarbeit*, Technische Universität München, 2005.

[16] M. Shum: Lecture Notes: Simulation. `http://www.hss.caltech.edu/~mshum/gradio/simulation.pdf`

[17] P.G. Zhang: Exotic Options. A Guide to Second Generation Options, World Scientific, 1998.