Monte-Carlo Simulation with Boundary Conditions

(with Applications to Stress Testing, CEV and Variance-Gamma Simulation)

Christian P. Fries
email@christian-fries.de

Jörg Kienitz
joerg.kienitz@gmx.de

April 27, 2010

Version 1.2

Abstract

In this paper we discuss how to incorporate analytic boundary conditions into a Monte-Carlo simulation framework and discuss their applications. The method introduced can dramatically improve the stability, robustness and accuracy of valuation, calculation of sensitivities and stress testing, i.e., valuation under stressed model parameters.

We propose a Monte-Carlo simulation scheme which features boundary conditions for the underlying value process. The boundary conditions are analogous to the boundary conditions of a PDE. The Monte-Carlo simulation will then be modified to generate paths only within the boundaries and generate the corresponding Monte-Carlo weights. In addition, the valuation algorithm is adjusted to incorporate the analytic boundary conditions, using given or estimated boundary values for the value process.

The complete setup consists of four parts:

- The definition of a boundary and the corresponding in-bound and out-bound regime. This is done for each time step.
- The definition of a Monte-Carlo scheme for which all paths remain within the boundary.
- The definition of a boundary condition which defines the value process in the out-bound region and its valuation conditional to being in the in-bound region at the previous time-step.
- A modified pricing algorithm which allows one to evaluate the product using the Monte-Carlo scheme within the boundary conditions, adding the boundary value process.

We present four different methods for calculating the product specific boundary value process: analytic, super-hedge, sub-hedge and numeric. Among the applications of the method we specifically look at its behavior under stressed data. The method is also useful for schemes with problematic boundary behavior (e.g. Heston or CEV like models).

Contents

1 Introduction

In this paper we extend the idea of Fries and Joshi [6] to a generalized numerical Monte-Carlo method. In short, the idea in this paper is to combine ordinary Monte-Carlo simulation with a feature from PDE schemes: *boundary conditions*. As we will illustrate in the following subsections, boundary conditions are a feature of PDE schemes which result in extra stability of the numerical result, a stability which is often lacking in Monte-Carlo simulation.

As application we will consider two cases:

- The behavior of the (Monte-Carlo) valuation under stressed model parameters.
- The behavior of the (Monte-Carlo) valuation for schemes with biased boundary behavior, like reflecting/absorbing states in Heston or CEV model.

The discussion of these two applications is by no means exhaustive. The method which we will present is general in spirit.

1.1 Problem Description I: Stress Testing

A stress test is an important tool for analyzing a portfolio. A stress test of a portfolio is given by evaluating the portfolio under *stressed* marked data.

Valuation under stressed marked data is demanding in several respects. First, calibration to such market data can become very difficult, if not impossible (e.g., if the stressed marked data constitutes arbitrage violations which cannot be reproduced by an arbitrage free model). If the calibration problem is solved[1], we obtain stressed model parameters from stressed market data. So a second demanding aspect is to ensure that the numerical scheme remains stable (i.e., convergent) under these stressed model parameters.

Very simple examples already illustrate that a valuation using a PDE solver is much more robust with respect to stressed model parameters, compared to a Monte-Carlo valuation. In fact, as we will see next, even for numerical schemes which are simply the exact solution (like the log-Euler scheme of a Black-Scholes model), stressed model parameters can cause Monte-Carlo simulation to fail.

1.1.1 The Paradoical Case of the Monte-Carlo Valuation of a Call under very high Volatility

Consider a simple Black-Scholes model

$$\mathrm{d}S(t) \;=\; rS(t)\mathrm{d}t \;+\; \sigma S(t)\mathrm{d}W^{\mathbb{Q}}(t). \tag{1}$$

with σ becoming very large as part of a stress test.[2] From the solution of the SDE (1),

$$S(t) \;=\; S(0)\,\exp\left(r\,t \;-\; \frac{1}{2}\sigma^2 t \;+\; \sigma W(t)\right),$$

[1] One way to cope with the calibration under stressed market date is simply to relax the calibration accuracy and restrict the stress tests to marked data compatible with the model assumption.

[2] We adapt the notation from [4], e.g., $\mathbb{Q}$ denotes the equivalent martingale measure and $W^{\mathbb{Q}}$ is a Brownian motion under $\mathbb{Q}$.

we see that for large σ the majority of the paths will tend to zero due to the large negative drift. Very few paths will gain very large values. Indeed we have

$$\mathbb{Q}\left(S(T) < \epsilon\right) \ \to \ 1 \qquad \text{for } \sigma \to \infty.$$

For a Monte-Carlo simulation with a finite number of paths n, this implies that only very few paths will come out with $S(T) > K$. The Monte-Carlo error of the value of the payoff of an European call

$$V(T) \ = \ \max(S(T) - K, 0)$$

will become very large. Even worse, the Monte-Carlo valuation will tend to zero if σ tends to infinity.[3]

However, from the Black Scholes formula we have the value of a European call, having payoff $V(T) \ = \ \max(S(T) - K, 0)$, given by

$$V(0) \ = \ S(0)\Phi(d_+(\sigma)) - K\exp(-rT)\Phi(d_+(\sigma)),$$

which tends to $S(0)$ for $\sigma \to \infty$.

These numerical problems can be observed on more complex models as well. For example, in a local volatility model

$$\mathrm{d}S(t) \ = \ r(t)S(t)\mathrm{d}t \ + \ \sigma(t, S(t))\mathrm{d}W(t).$$

A regime where the local volatility is large will effectively push paths away from this regime, and towards boundaries or infinity.

Note that in this example our numerical scheme *is* the exact solution of the SDE, i.e., there is no sampling error due to a time-discretization.

1.1.2 Why a PDE is much more Robust

For the pricing of a European call in a PDE framework it can be observed that the value is much more robust against stresses to the (local) volatility function. The reason is simple: The PDE does not have a problem if the process is pushed to the boundary (i.e., if the density is such that mass is diffused to the boundaries), because it has analytic boundary conditions. If you consider an implicit Euler scheme for the Black-Scholes model, then, in the limit $\sigma \to \infty$ the scheme converges to a simple weighted average of the boundary values. In this case the valuation of a European call with strike K converges to $S(0) - \frac{S(0)}{b}K$ where b is the location of the boundary. For $b >> K$ this is a sufficiently good approximation of the true solution $S(0)$. Moreover, since the boundary values are analytic, there are no additional numerical errors in the limit.

[3] For very large volatilities it becomes more and more likely that no path comes out with $S(T) > K$. Indeed, if our Monte-Carlo simulation is given by the paths $\omega_1, \ldots, \omega_n$, then

$$S(T, \omega_i) \leq S(0) \ \exp\left(r\,T \ - \ \frac{1}{2}\sigma^2 T \ + \ \sigma(\max_{1\leq i\leq n} W(T, \omega_i))\right) \to 0$$

for $\sigma \to \infty$.

1.2 Problem Description II: Models with Biased Boundary Behavior

For some models, the numerical scheme will exhibit biased behavior since the boundaries may be pathological. By pathological we mean that, for example, the process can be absorbed or reflected at 0. To this end a numerical scheme may produce negative values while the model in theory does not. One example is the Euler scheme and the CEV model. Theoretically, the solution of the SDE is guaranteed to be equal to or bigger than 0 but the numerical scheme may produce values smaller than 0. Such problems either make it necessary to consider very small time steps by at the same time increasing the time spent on generating the paths or considering sophisticated and complex sampling techniques, see for instance [13] or, [2] and references therein. The first article discusses the application of an Euler discretization to simulate mean-reverting CEV processes and the problems that arise there, namely the possibility of simulating negative values for the variance. They propose some truncation schemes and show the performance for option pricing for affine stochastic volatility models and the Heston model in particular. For a class of affine stochastic volatility processes a biased free scheme exists. This is derived in the second paper. To this end the characteristic function is derived and a simulation algorithm using this characteristic function is developed and applied. The simulation scheme is known to be slow since special functions are involved.

We show that the Monte-Carlo simulation with boundary conditions can be an alternative in such cases. This is done by excluding the problems arising by using numerical schemes by simply excluding this possibility from the Monte-Carlo simulation and replacing it with an appropriate boundary condition. Then, under the assumption that one is able to efficiently price simple options below, on or above the boundary, for instance calls, puts and digitals, this modified scheme is not only accurate but performs very well and is therefore suitable for applied Monte Carlo simulation in a productive system. We apply our method to the CEV process and the Variance Gamma process but in principle it is possible to extend the range of models beyond this class since efficient pricing formula are often available for financial models. This is due to the fact that models need to be calibrated to liquid simple option prices quoted in the market.

1.3 Solution

We propose a Monte-Carlo simulation scheme which features boundary conditions for the underlying value process (i.e., in the notation above $V(t, S)$). The boundary conditions are analogous to the boundary conditions of a PDE. The Monte-Carlo simulation will then be modified to generate paths only within the boundaries and generate the corresponding Monte-Carlo weights. In addition, the valuation algorithm is adjusted to incorporate the analytic boundary conditions, using given or estimated boundary values for $S \mapsto V(t, S(t))$.

The complete setup consists of four parts:

- The definition of a boundary and the corresponding in-bound regime A and out-bound regime B. This is done for each time step t_i.
- The definition of a Monte-Carlo scheme for which all paths remain within the boundary, i.e., within A.

- The definition of a boundary condition which defines the value process V in the region B and its valuation conditional to being in A at the time-step before.
- A modified pricing algorithm which allows one to evaluate the product using the Monte-Carlo scheme within the boundary conditions, adding the boundary value process.

The Monte-Carlo scheme is a generalization of the conditional analytic Monte-Carlo scheme described in Fries & Joshi [6].

It is possible to define the boundary and the boundary conditions in a path-dependent form. This can be useful when considering the valuation of products with path-dependent features.

The Monte-Carlo scheme itself (which is just one part of the Monte-Carlo simulation with boundary conditions) can be understood as an importance sampling. The calculation of the likelihood ratios is performed at the level of the numerical scheme, which usually allows one to calculate the likelihood ratios for almost any model, even those where analytic solutions for transition densities are not available. The concept here is the same as for the proxy simulation schemes [7, 5].

While an importance sampling (in its original form) simply modifies the location of sampling paths to adapt to product specify features our procedure is different: The sample paths are restricted to a certain domain, usually not adapted to the product. Inside this domain a Monte-Carlo pricing is performed. Outside a (semi-)analytical valuation is performed. The valuation algorithms interweaves the two value processes in each time step to represent the true option value.

The splitting of the domain is (primarily) motivated by model aspects, and not motivated by a product feature.

Within the Monte-Carlo domain we could still apply an additional importance sampling, accounting for product specific features. The Monte-Carlo simulation restricted to the in-bound domain can be combined with a proxy simulation scheme.

1.4 Other Work

To some extend the MCBC can be viewed as a generalization of the conditional analytic pricing discussed in [6], which itself can be viewed as a generalization of the method for valuation of barrier options in Monte-Carlo simulation, e.g. Glasserman and Staum, [9].

For a barrier option the product induces a natural boundary, the barrier of the option. In this setup the boundary condition (to speak in our terms) is much simpler; it is a constant redemption, often simply zero. Another assumption present in this setup is that the product is assumed to terminate when the barrier is hit, i.e. the future value of the product given a barrier hit is zero. For Monte-Carlo valuation of barrier options see, e.g., [11].

In contrast to [9, 6] we introduce a boundary which is completely independent of the product under consideration. We do not make any assumption about the product, except that there is an algorithm for calculating a sufficiently accurate ?boundary value process?, given some boundary conditions. For example, assuming linear boundary

condition we require approximation formulas for the value and slope of the product's continuation value.

We present four different methods for calculating the product specific boundary value process: analytic, super-hedge, sub-hedge and numeric. Among the applications of the method we specifically look at its behavior under stressed data.

1.5 Plan of Paper

We start by defining the general setup of a Monte-Carlo simulation with boundary conditions (MCBC) in Section 2. This section gives an overview of the valuation framework on an abstract level. We show how to adapt the valuation algorithm to the scheme (Section 2.2) and how to define the boundary value process (Section 2.3). In order to define the boundary value process, we chose an extrapolation. While this is a trivial step in a PDE implementation, it is non-trivial in a Monte-Carlo simulation. In Section 2.5 we present three different ways of defining or calculating the value extrapolation in a Monte-Carlo simulation.

In Section 3 we then give the explicit construction of the scheme and the boundary value process by considering Euler schemes of Itô processes. Assuming a functional representation of the boundary, we can adapt the construction of [6] (Section 3.4). For our Euler scheme, the boundary value process can be given by conditional Black-Scholes formulas (Section 3.6), assuming that an estimate or approximation of the conditional expectation of the value process $x \mapsto V(t,x)$ exists, conditional on x being outside the boundary. This completes the discussion of the MCBC for the case of an Euler discretized Itô process.

In Section 4 we generalize the results to CEV and Variance Gamma processes. We conclude the paper with numerical results in 5.

2 Monte-Carlo Simulation with Boundary Conditions - General Setup

2.1 Restriction of the Scheme

We consider the Monte-Carlo valuation of a value process $t \mapsto V(t)$. The value process is derived using a Monte-Carlo simulation of model primitives $t \mapsto X(t)$, discretized using a time discretization scheme at $0 = t_0 < t_1 < \ldots$.

Let us first consider the general setup on an abstract level. Assume that the Monte-Carlo simulation is modified to sample only a restricted domain

$$A := \{\omega \mid X(t_i, \omega) \in A_i \, \forall i\},$$

for some given sets A_i. The set A_i can be defined in a path dependent manner, if necessary, as long as it is $\mathcal{F}_{t_{i-1}}$ measurable, i.e., it is allowed that A_i depends on the past up to the previous time-step.

We assume that the Monte-Carlo simulation is restricted to A such that it allows the valuation of all payoffs $V\mathbf{1}_A$. This can be achieved by an importance sampling. Let B_i denote the domain $X(t_i, \Omega) \setminus A_i$.

2.2 Valuation under the Monte-Carlo Simulation with Boundary Conditions

The following section describes how to evaluate financial products with the Monte-Carlo simulation with boundary conditions. To allow for an evaluation of arbitrary products we have to derive a representation of the value process outside the boundary. This can be done from appropriate *boundary conditions*. For the time being we assume that such a representation exists. The discussion of how to obtain the boundary conditions is given in the later sections.

Let $V^i(T_i)$ denote the value of the financial product in time T_i (excluding cash-flows in $T < T_i$). Let $C_i(T_i)$ denote the time T_i value of the cash-flows in $[T_i, T_{i+1})$, i.e.,

$$C_i(T_i) = V^i(T_i) - V^{i+1}(T_i),$$

where[4]

$$V^{i+1}(T_i) = N(T_i) \, \mathrm{E}^{\mathbb{Q}} \left(\frac{V^{i+1}(T_{i+1})}{N(T_{i+1})} \mid \mathcal{F}_{T_i} \right).$$

Furthermore let

$$\begin{aligned} V^{\mathrm{out},i+1}(T_{i+1}) &:= V^{i+1}(T_{i+1}) \mathbf{1}_{B_{i+1}}, \\ V^{\mathrm{in},i+1}(T_{i+1}) &= V^{i+1}(T_{i+1}) \mathbf{1}_{A_{i+1}}, \end{aligned}$$

and thus

$$\begin{aligned} \frac{V^{\mathrm{out},i+1}(T_i)}{N(T_i)} &= \mathrm{E}^{\mathbb{Q}} \left(\frac{V^{\mathrm{out},i+1}(T_{i+1})}{N(T_{i+1})} \mid \mathcal{F}_{T_i} \right), \\ \frac{V^{\mathrm{in},i+1}(T_i)}{N(T_i)} &= \mathrm{E}^{\mathbb{Q}} \left(\frac{V^{i+1}(T_{i+1})}{N(T_{i+1})} \mathbf{1}_{B_{i+1}} \mid \mathcal{F}_{T_i} \right). \end{aligned}$$

[4] Here N denotes the numéraire and $\mathbb{Q}$ the corresponding equivalent martingale measure.

Since $A_{i+1} \cup B_{i+1} = X(T_{i+1}, \Omega)$ we have

$$V^{i+1}(T_i) \;=\; V^{\text{out},i+1}(T_i) + V^{\text{in},i+1}(T_i).$$

$V^{\text{out},i+1}$ is the value of the paths ending in the out-bound domain in time T_{i+1} and $V^{\text{in},i+1}$ is the value of the paths ending in the in-bound domain in time T_{i+1}.

We make the following assumption:

- We have an analytic formula $V^{\text{out},i+1}(T_i)$ (or an approximation thereof), at least when $X(T_i)$ is in the in-bound domain.[5]

Then, we define the modified valuation process recursively (in a backward algorithm) as

$$\tilde{V}^{\text{in},i}(T_i) \;:=\; V^{\text{out},i+1}(T_i) + \tilde{V}^{\text{in},i+1}(T_{i+1}) \frac{N(T_i)}{N(T_{i+1})} \mathbb{Q}(A_{i+1}|\mathcal{F}_{T_i}) + C(T_i)\mathbf{1}_{A_i},$$

given some final value $\tilde{V}^{\text{in},n}(T_n)$.

The important observation here is that $C(T_i)$ has to be evaluated only on A_i, since we define the value of $\tilde{V}^{\text{in},i}(T_i)$ only under the assumption of $X(T_i)$ being in-bound. In other words: Given $V^{\text{out},i+1}(T_i)$ (or an approximation there of), $\tilde{V}^{\text{in},i}(T_i)$ can be constructed from the modified (in-bound) Monte-Carlo simulation.

Note that on $\{X(T_i) \in A_i\}$ we have (by backward induction)

$$\begin{aligned} \mathrm{E}^{\mathbb{Q}}\left(\frac{\tilde{V}^i(T_i)}{N(T_i)} \mid \mathcal{F}_{T_i} \right) \;&=\; \frac{1}{N(T_i)} \left(V^{\text{out},i+1}(T_i) + V^{\text{in},i+1}(T_i) + C(T_i) \right) \\ &=\; \frac{1}{N(T_i)} \left(V^{i+1}(T_i) + C(T_i) \right) \;=\; \frac{V^i(T_i)}{N(T_i)} \end{aligned}$$

and thus

$$V^i(T_i) \;=\; \mathrm{E}^{\mathbb{Q}}\left(\tilde{V}^i(T_i) \mid \mathcal{F}_{T_i} \right) \quad \text{on } \{X(T_i) \in A_i\}.$$

Remark: If the product has an early exercise or any other payout conditional to it's future value, that can be incorporated in $C(T_i)$.

Given that our Monte-Carlo simulation starts within the boundaries, the value of the product at T_0 then is

$$V(T_0) \;=\; \mathrm{E}^{\mathbb{Q}}\left(\tilde{V}^0(T_0) \mid \mathcal{F}_{T_0} \right).$$

2.3 Definition of the Boundary Value Process

The only missing link to complete the valuation is the definition of the boundary value process $V^{\text{out},i+1}(T_i)$. One possible approach to defining it is the following:

- Determine a functional representation of $V^{\text{out},i+1}(T_{i+1})$, e.g.,

$$V^{\text{out},i+1}(T_{i+1}) \;=\; G(t_{i+1}, X(T_{i+1})).$$

[5] We will derive an approximation to $V^{\text{out},i+1}(T_i)$ in Section 2.3.

- Derive valuation formulas for $V^{\text{out},i+1}(T_i)$ using the transition probabilities of the numerical scheme, i.e.,

$$V^{\text{out},i+1}(T_i) \;=\; N(T_i)\mathrm{E}^{\mathbb{Q}}\left(\frac{V^{\text{out},i+1}(T_{i+1})}{N(T_{i+1})} \mid X(T_i)\right),$$

2.4 Example: Linear Boundary Conditions on a Scalar Underlying

In this section we give an example of the case of a linear boundary function in a scalar underlying, i.e., we assume that the value process is a linear function of the underlying in the out-bound region B_i. This is exactly what some PDE solvers do. Assuming that $V^{\text{out},i+1}(T_{i+1})$ is given by a linear function, we can derive formulas for $V^{\text{out},i+1}(T_i)$.

Let us consider that $X(t)$ is a scalar random variable and the boundary in T_{i+1} is given by the scalar values x_l, x_u.

If $X(t_i)$ is a scalar random variable and $V^{\text{out},i+1}(T_{i+1})$ is a linear functional (in $X(T_{i+1})$) for $X(T_{i+1}) < x_l$ and for $X(T_{i+1}) > x_u$, then $V^{\text{out},i+1}(t_i)$ is given by a combination of option (calls/put) and digital option prices. If we consider a log-normal Euler time-step $T_i \to T_{i+1}$, then these option prices are given by standard Black-Scholes formulas.

For

$$\begin{aligned} V(T_{i+1},x) \;&=\; ax+b && \text{for } x < x_l \\ V(T_{i+1},x) \;&=\; cx+d && \text{for } x > x_u \end{aligned}$$

the conditional values of the payoffs are given by option prices. It is

$$\begin{aligned} &\mathrm{E}^{\mathbb{Q}}\left(\frac{V^{\text{out},i+1}(T_{i+1})}{N(T_{i+1})} \;\middle|\; X(T_i)\right) \\ &\quad= (b+ax_l)DP(X(T_i),x_l,T_{i+1}-T_i) - aP(X(T_i),x_l,T_{i+1}-T_i) \\ &\qquad+ (d+cx_u)DC(X(T_i),x_u,T_{i+1}-T_i) + cC(X(T_i),x_l,T_{i+1}-T_i) \end{aligned}$$

where $DP(X,K,T)$ is the value of a digital put, $DC(X,K,T)$ is the value of a digital call, $P(X,K,T)$ is the value of a put, $C(X,K,T)$ is the value of a call, each with spot X, strike K and maturity T.

2.5 Determining the Linear Boundary Value Functional

A PDE implementation usually determines the linear extrapolation of the value function by evaluating the neighboring points. Hence, a PDE uses a linear extrapolation. This is often implicit in the boundary condition of the PDE solver.

For a Monte-Carlo simulation it is not obvious how the linear boundary value functional $x \mapsto V^{\text{out},i+1}(T_{i+1},x)$ can be determined. We will discuss three different variants.

2.5.1 Analytic Linear Extrapolation of the Boundary Value Functional

For products where there is an analytic valuation formula and an analytic formula for its delta we can derive a linear boundary value functional analytically. This is for example

the case for simple European options. Of course, one would not use a Monte-Carlo simulation at all in this case. However, we will consider this case to benchmark our method in Section 5.1.

2.5.2 Boundary Value Functional as Sub- or Super-Hedge

A very simple, but nevertheless useful, definition of the boundary value functional can be given by using a static sub- or super-hedge for the derivative product. Assume that there are functions G^{sup} and/or G^{sub} such that on B_{i+1} we have

$$G^{\text{sub}}(T_{i+1}, X(T_{i+1})) \ \leq \ V^{i+1}(T_{i+1}) \ \leq \ G^{\text{sup}}(T_{i+1}, X(T_{i+1})),$$

where V^{i+1} denotes the true value of the future cashflows of the product under consideration.

A sub-hedge boundary condition is given by using G^{sub} in place of G as an approximation of $V^{\text{out},i+1}(T_{i+1})$. For a sub-hedge boundary condition the Monte-Carlo valuation of $V(t_0)$ will then result in a lower bound of the true option price.

A sub-hedge boundary condition is given by using G^{sup} in place of G as an approximation of $V^{\text{out},i+1}(T_{i+1})$. For a super-hedge boundary condition the Monte-Carlo valuation of $V(t_0)$ will then result in an upper bound of the true option price.

Both methods will only give bounds for the true option prices; however, the deviation of the value depends on the location of the boundary.

2.5.3 Numerical Linear Extrapolation of the Boundary Value Functional

In the general case a linear extrapolation of the boundary value functional can be determined numerically, e.g., using a regression.

Such a numerical calculation of the boundary value process can be designed to be completely product independent. The method then resembles more closely the approach taken by a PDE, where the extrapolation of the value functional is determined numerically.

However, the numerical calculation of the boundary value process using a simple regression is not suitable for our application to stress testing. This is easy to see: The plain regression on the un-modified Monte-Carlo simulation will suffer from the same degeneration as the standard Monte-Carlo valuation: See Figure 4.

3 Monte-Carlo Simulation with Boundary Conditions for Euler-Schemes of Itô Processes

As an example we consider a Monte-Carlo simulation using an Euler scheme for X which is constrained to be below some boundary, where the boundary is given by a scalar function defining some threshold. The setup is hence similar to the the corresponding scheme for a *Conditional Analytic Monte-Carlo Pricing*, as discussed in [6], where such a situation occurred from a product specific barrier feature.

3.1 Underlying Model

We consider a pricing model given by a discretized stochastic process X. For illustrative purposes we consider a model given by an Itô stochastic process [4, 10, 12]

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

where $W = (W_1, \ldots, W_m)$ and W_i are Brownian motions with

$$\mathrm{d}W_i\,\mathrm{d}W_j \;=\; \delta_{i,j}\,\mathrm{d}t,$$

and Σ denotes the square-root of the instantaneous covariance matrix. We focus on this model mainly for illustrative purposes. We allow the possibility that the coefficients are state-dependent.[6]

3.2 Assumption about the Boundary

We assume that the boundary can be written as

$$X(t_j) \text{ is within bounds } \;\Leftrightarrow\; X(t_j) \in A_j \;\Leftrightarrow\; f(t_j, X(t_j), H_j) < 0,$$

where $x \mapsto f(t_j, x, H_j)$ is a linear function of the model's state variables $X(t_j)$ and H_j is some $\mathcal{F}_{t_{j-1}}$-measurable random variable. Note that this would allow for path-dependent boundary conditions (due to the random variables H_j). But for most applications one would stick to $H_j = 0$, i.e., assume that the boundary conditions are a linear function of the model's state variables $X(t_j)$.[7]

3.3 Original Numerical Scheme

Let $\tilde{X}(t_i)$ be an approximation of $X(t_i)$ given by a numerical scheme, e.g., an Euler discretization of our model given by

$$\Delta\tilde{X}(t_i) \;=\; \tilde{\mu}(t_i)\Delta t_i + \Sigma(t_i)\cdot\Gamma(t_i)\cdot\Delta W(t_i), \quad \tilde{X}(0) = X(0).$$

Let $\Delta W_k(t_i)$ be generated by drawings from independent equi-distributed random variables $Z_{i,k}$ using

$$\Delta W_k(t_i) \;=\; \Phi^{-1}(Z_{i,k})\sqrt{\Delta t_i},$$

[6] The LIBOR market model and swap market models are of this type. For the LIBOR market model we have $X_i = \log(L_i)$, where L_i is the forward rate for the period $[T_i, T_{i+1}]$.

[7] Non-linear boundaries can be handled, but we doubt that this is necessary. We will comment on the implementation of non-linear boundaries later.

where Φ^{-1} denotes the inverse of the cumulative standard normal distribution function. The fact that post-discretization the model increments are Gaussian means that the method still applies despite the state-dependence.

Many models in derivative pricing are based on Itô processes and an implementation using an Euler scheme is often sufficient. However, it is straightforward to extend our approach to more general processes and other discretization schemes, as long as there are formulas for the distribution functions of the transition probability of $\Delta X(t_i)$.[8]

3.4 Numerical Scheme adapted to the Boundary

We will define a Monte Carlo valuation scheme that stays within a pre-defined boundary and calculate the corresponding probability measure resulting from this constraint. We will assume that the boundary is linear with respect to the one-step Brownian increment.

Induction Start: Let $X^*(t_0) := X(t_0)$.

Induction Step: Given $X^*(t_i)$ let

$$g(\xi) = f\left(t_{i+1}, X^*(t_i) + \tilde{\mu}(t_i, X^*)\Delta t_i + \xi, H_{i+1}\right).$$

Define

$$v = \nabla g(0) / ||\nabla g(0)|| \tag{2}$$

and let $q \in \mathbb{R}$ be the solution[9] of

$$g(qv) = 0.$$

We will assume now, that g is linear in x, and the discussion of the general case will follow below. If g is linear in x we have

$$g(qv) \;=\; g(0) + \nabla g(0) \cdot qv, \quad \text{i.e., define} \quad q \;:=\; \frac{g(0)}{||\nabla g(0)||}. \tag{3}$$

Then

$$\begin{aligned} X(t_{i+1}) \text{ is within bounds } &\Leftrightarrow\; f(t_{i+1}, X(t_{i+1}), H_{i+1}) < 0 \;\Leftrightarrow\; g(\Sigma\Gamma\Delta W) < 0 \\ &\Leftrightarrow\; \Sigma\Gamma\Delta W < qv \;\Leftrightarrow\; \langle v, \Sigma\Gamma\Delta W\rangle < q \end{aligned}$$

Let

$$\Delta Y := \langle v, \Gamma\Delta W\rangle.$$

We wish to replace the sampling of ΔY with a sampling ΔY^* such that $\Delta Y^* < q$. Clearly, ΔY is a normal distributed random variable with mean 0. Let $\sigma_{\Delta Y}$ denote the standard deviation of ΔY. Then $y = \Phi(\Delta Y/\sigma_{\Delta Y})$ is uniformly distributed. Let $b := \Phi(q)$ and $\Delta Y^* := \Phi^{-1}(by)$. Then we have $by < b$, thus $\Delta Y^* Z < q$. Furthermore,

$$\mathbb{Q}(\Delta Y < K) \;=\; b\mathbb{Q}(\Delta Y^* < K)$$

[8] Generalizations to Levy processes are given in Sections 4.

[9] To be precise, we consider $q \in \mathbb{R}$ conditional $\mathcal{F}_{T_i}$, i.e., q is a $\mathcal{F}_{T_i}$-measurable random variable.

for all $K < q$. i.e., the distribution function of ΔY^* and ΔY differ on $(-\infty, q)$ only by the constant factor b.

In other words: sampling ΔY^* is equivalent to sampling ΔY in the restricted domain $(-\infty, q)$, with a Monte-Carlo weight b. For $\Gamma\Delta W + (\Delta Z - \Delta Y)v$ we have

$$\langle v, \Gamma\Delta W + (\Delta Y^* - \Delta Y)v\rangle \;=\; \Delta Y + \Delta Y^* - \Delta Y \;=\; \Delta Y^* \;\leq\; q$$

In place of $\tilde{X}$ we consider the numerical scheme X^* defined by

$$X^*(t_{i+1}) \;:=\; X^*(t_i) + \tilde{\mu}(t_i)\Delta t_i + \Sigma \cdot (\Delta W + (\Delta Y^* - \Delta Y)v)$$

This scheme has the property that

$$\begin{aligned} f(t_{i+1}, X^*(t_{i+1}), H_{i+1}) \;&=\; f(t_{i+1}, X^*(t_i) + \tilde{\mu}(t_i)\Delta t_i, H_{i+1}) \\ &\quad + \nabla g \cdot \left(\Delta W + (\Phi^{-1}(qU) - \Phi^{-1}(U))v_1\right) \qquad (4) \\ &=\; g(0) + \nabla g \cdot qv \;\leq\; 0 \end{aligned}$$

Thus, for linear boundaries we have that this scheme generates realizations that sample the in-bound region only. For the original increment we had

$$\mathbb{Q}\left(f(t_{i+1}, X(t_{i+1}), H_{i+1}) \;\leq\; 0 \mid X^*(t_i)\right) \;=\; b,$$

for the adapted scheme we have

$$\mathbb{Q}\left(f(t_{i+1}, X^*(t_{i+1}), H_{i+1}) \;\leq 0 \mid X^*(t_i)\right) \;=\; 1,$$

i.e., the Monte-Carlo weight of the corresponding sample path will be multiplied by a factor of b.

Note that this is applied conditionally to t_i in each time step[10].

3.4.1 Inner and Outer Region

Linking to the notation of the modified pricing algorithm described in Section 2.2 we have

$$\begin{aligned} A_i \;&=\; \{f(t_i, X(t_i), H_i) \;\leq 0\} \\ B_i \;&=\; \{f(t_i, X(t_i), H_i) \;> 0\} \\ \mathbb{Q}(A_{i+1}|\mathcal{F}_{T_i}) \;&=\; b \end{aligned}$$

3.5 Some Generalizations

3.5.1 Multiple Boundaries: Upper and Lower Boundaries

We can generalize the above definition to the case where we consider multiple boundaries, e.g., an upper and a lower boundary defining the modification of the scheme in a piece-wise manner. We will work out the case of upper and lower bounds here:

[10] effectively x,y,b are processes

Let $x \mapsto f_u(t_i, x, H_i)$ and $x \mapsto f_l(t_i, x, H_i)$ denote two linear functions. We want to constrain our Monte-Carlo simulation of X to the domain

$$A_i := \{f_u(t_i, X(t_i), H_i) \leq 0\} \cap \{f_l(t_i, X(t_i), H_i) \leq 0\}$$

where we assume that there is some partition of the domain $A_i = A_i^u \cup A_i^l$ such that

$$\begin{aligned} f_u(t_i, x, H_i) &> 0 && \text{for } x \in A_i^l \\ f_l(t_i, x, H_i) &> 0 && \text{for } x \in A_i^u \end{aligned}$$

Then, the Monte-Carlo scheme constrained to the domain A_i is defined as follows

Given $X^*(t_i)$. For paths ω where $X^*(t_i, \omega) + \tilde{\mu}(t_i, X^*(\omega))\Delta t_i + \Gamma \Delta W(t_i, , \omega)\, in A_i^u$ we define $X^*(t_i, \omega)$ using $f = f_u$. For all other paths we define $X^*(t_i, \omega)$ using $f = f_l$.

Obviously we can extend this definition to more complex boundaries, using a partitioning such that on each partition we are in a setup where the boundary is defined by a linear function f as above.

If the boundary is described by a nonlinear function f, a linearization is often feasible.

3.5.2 Other Transition Probabilites

The Monte-Carlo scheme may be generalized to other transition probabilities and other models. The idea is the same as in [16, 7, 5, 6]: the method presented made only a few basic assumptions about the numerical discretization scheme of the model. E.g., the derivation does not assume that $\Delta W(t_i) := \int_{t_i}^{t_{i+1}} \mathrm{d}W(t)$ is a Brownian increment. We just required there to be a sufficiently efficient and accurate algorithm to calculate the conditional cumulative distribution function of $\Delta X(t_i) = X(t_{i+1}) - X(t_i)$ and its inverse. If μ, Σ and Γ are previsible then this is equivalent that an there is an sufficiently efficient and accurate algorithm to calculate the conditional cumulative distribution function of $\Delta W(t_i)$.

We will consider the CEV process in Section 4.1 and the Variance Gamma Process in Section 4.2.

3.6 Definition of the Boundary Value Process

3.6.1 Case of a Scalar Underlying

Let us consider that $X(t)$ is a scalar random variable and the boundary is given by the scalar values $x_l(t)$, $x_u(t)$.

If $X(t_{i+1})$ is a scalar random variable and $V^{\text{out},i+1}(t_{i+1})$ is a linear functional (in $X(t_{i+1})$) for $X(t_{i+1}) < x_l(t_{i+1})$ and for $X(t_i) > x_u(t_{i+1})$, then $V^{\text{out},i+1}(t_i)$ is given by a combination of option (calls/put) and digital option prices as described in Section 2.4.

If we consider a log-normal Euler time-step $t_i \to t_{i+1}$, then these option prices are given by standard Black-Scholes formulas. Let $\sigma(t_i)$ denote the volatility of the Euler-scheme step from $X(t_i)$ to $X(t_{i+1})$. Then, with the notation of Section 2.4, we

have

$$\begin{aligned}
&DP(X(t_i), x_l(t_{i+1}), t_{i+1} - t_i) \\
&\quad = 1 - \Phi(d_-(X(t_i), x_l(t_{i+1}))) \\
&P(X(t_i), x_l(t_{i+1}), t_{i+1} - t_i) \\
&\quad = \exp(x(t_{i+1}))\Phi(d_+(X(t_i), x_l(t_{i+1}))) - \exp(x_l(t_{i+1}))\Phi(d_-(X(t_i), x_l(t_{i+1})))
\end{aligned}$$

and

$$\begin{aligned}
&DC(X(t_i), x_l(t_{i+1}), t_{i+1} - t_i) \\
&\quad = \Phi(d_-(X(t_i), x_u(t_{i+1}))) \\
&C(X(t_i), x_l(t_{i+1}), t_{i+1} - t_i) \\
&\quad = \exp(x_l(t_{i+1}))\Phi(d_+(X(t_i), x_l(t_{i+1}))) - \exp(X(t_i))N(d_-(X(t_i), x_l(t_{i+1})))
\end{aligned}$$

where

$$d_\pm(X, K) \;=\; \frac{X - K \pm \sigma(t_i)^2(t_{i+1} - t_i)}{\sigma(t_i)\sqrt{(t_{i+1} - t_i)}}.$$

Note that we worked in normal coordinates, not log-normal. Hence the $\exp$.

3.6.2 General Case

The general case where X is a vector valued random variable can be reduced to the scalar case when the boundary is described by a (linear) scalar valued function f, as assumed in Section 3.2. In that case we have to replace the above values with the corresponding projections onto the direction v, where v is as in (2).

4 Monte-Carlo Simulation with Boundary Conditions for CEV and Variance Gamma Processes

Currently financial models are extended to cover certain market observations such as the skew or smile phenomenon. To this end other processes than the geometric Brownian motion have been applied. The *constant elasticity of variance* model has been introduced for skew modeling in LIBOR Market models in [1]. For other asset classes Lévy processes have been used. One well known example is the *Variance Gamma* processes introduced in [15].
To this end we apply the Monte Carlo with boundary methodology to these processes.

4.1 The CEV Process

Given $r \geq 0$, $d \geq 0$, we choose $\beta \in [0,1)$ since this is the most relevant for financial applications. We consider the constant elasticity of variance model, CEV, given by the following stochastic differential equation:

$$\begin{aligned} dS(t) &= (r-d)S(t)dt + \sigma S(t)^{\beta} dW(t) \\ S(0) &= S_0 \end{aligned} \tag{5}$$

The CEV model is challenging to simulate because in practice when applying simple schemes it may generate negative values and then crash in the next step when trying to calculate x^{β} with $x < 0$.

The transition density for the CEV model is known explicitly: It is a χ^2 distribution. To this end we can compute the probability of ending above the upper or falling below the lower bound. These probabilities determine the weights for applying the importance sampling algorithm.
There are closed form pricing formulas available for calls and puts. We set $\nu = \frac{1}{2(1-\beta)}$. Then, by [14], we have the following formulas for a call and a put. For absorbing boundary conditions we have:

$$\begin{aligned} C &= S(0)\left(1-\chi^2\left(\frac{4\nu^2K^{1/\nu}}{\sigma^2T};2\nu+2;\frac{4\nu^2S(0)^{1/\nu}}{\sigma^2T}\right)\right) \\ &\quad -K\chi^2\left(\frac{4\nu^2S(0)^{1/\nu}}{\sigma^2T};2\nu;\frac{4\nu^2K^{1/\nu}}{\sigma^2T}\right) \end{aligned} \tag{6}$$

$$\begin{aligned} P &= S(0)\chi^2\left(\frac{4\nu^2K^{1/\nu}}{\sigma^2T};2\nu+2;\frac{4\nu^2S(0)^{1/\nu}}{\sigma^2T}\right) \\ &\quad -K\left(1-\chi^2\left(\frac{4\nu^2S(0)^{1/\nu}}{\sigma^2T};2\nu;\frac{4\nu^2K^{1/\nu}}{\sigma^2T}\right)\right) \end{aligned} \tag{7}$$

For non-absorbing boundaries we have:

$$\begin{aligned} C &= S(0)\chi^2\left(\frac{4\nu^2K^{1/\nu}}{\sigma^2T};-2\nu;\frac{4\nu^2S(0)^{1/\nu}}{\sigma^2T}\right) \\ &\quad -K\left(1-\chi^2\left(\frac{4\nu^2S(0)^{1/\nu}}{\sigma^2T};-2\nu+2;\frac{4\nu^2K^{1/\nu}}{\sigma^2T}\right)\right) \end{aligned} \tag{8}$$

$$\begin{aligned} P &= S(0)\left(1-\xi^2\left(\frac{4\nu^2K^{1/\nu}}{\sigma^2T};-2\nu;\frac{4\nu^2S(0)^{1/\nu}}{\sigma^2T}\right)\right) \\ &\quad -K\chi^2\left(\frac{4\nu^2S(0)^{1/\nu}}{\sigma^2T};-2\nu+2;\frac{4\nu^2K^{1/\nu}}{\sigma^2T}\right) \end{aligned} \tag{9}$$

Let us denote $\tau = T-t$, $x = S(t)$, $X = S(T)$, then the transition density is given by:

$$\begin{aligned} p(X,T;x,t) &= \frac{\sqrt{xX^{1-4\beta}}}{(1-\beta)\sigma^2\tau}\exp\left(-\frac{x^{2(1-\beta)}+X^{2(1-\beta)}}{2(1-\beta)^2\sigma^2\tau}\right) \\ &\quad I_\nu\left(\frac{(xX)^{1-\beta}}{(1-\beta)^2\sigma^2\tau}\right) \end{aligned}$$

or in terms of the density p^* given by

$$p^*(x,r,\nu) = \frac{1}{2}\left(\frac{x}{\lambda}\right)^{(r-2)/4}\exp\left(-\frac{x+\lambda}{2}\right)I_{(r-2)/2}(\sqrt{\lambda x})$$

the transition function is:

$$p(X,T;x,t) = \frac{4\nu x X^{1/\nu-2}}{\sigma^2\tau}p^*\left(\frac{4\nu^2X^{1/\nu}}{\sigma^2\tau};2\nu+2;\frac{4\nu^2x^{1/\nu}}{\sigma^2\tau}\right)$$

In the case of absorbing boundary conditions, p is not a probability density since it does not integrate to 1. The probability of the process being absorbed in 0 can be calculated and is

$$\frac{1}{\Gamma(\nu)}\Gamma\left(\nu,\frac{2\nu^2x^{1/\nu}}{\sigma^2\tau}\right)$$

with the non normalized upper incomplete Γ function

$$\Gamma(a,x) = \int_x^\infty t^{a-1}e^{-t}dt$$

In a discretization of the absorbing CEV process this fact makes the model not easy to implement using a simple scheme.
With the above information we can calculate the probability for the process staying above a given barrier level $\epsilon > 0$. It is given by:

$$\chi^2\left(\frac{4\nu^2S(t_i)^{1/\nu}}{\sigma^2(t_{i+1}-t_i)};2\nu;\frac{4\nu^2\epsilon^{1/\nu}}{\sigma^2(t_{i+1}-t_i)}\right)$$

and staying below a given barrier level:

$$\left(1-\chi^2\left(\frac{4\nu^2S(t_i)^{1/\nu}}{\sigma^2(t_{i+1}-t_i)};2\nu;\frac{4\nu^2\epsilon^{1/\nu}}{\sigma^2(t_{i+1}-t_i)}\right)\right)$$

4.2 The Variance Gamma Process

In this subsection we consider the Variance Gamma process. We give two possible methods of applying the methods of this paper to the process. First, we show that using the subordinator approach we can directly apply the methods for the Gaussian setting and using the inverse transform, the transition density and numerical integration we can apply the methods directly to the Variance Gamma process.
We consider the Variance Gamma process X_t because there is a strong ongoing interest in applying this model in finance. The VG process has Lévy density

$$l_{VG}(x) = \frac{\exp\left(\frac{\mu x}{\sigma^2}\right)}{\nu|x|} \exp\left(-\frac{|x|}{\sigma}\sqrt{\frac{2}{\nu}+\frac{\mu^2}{\sigma^2}}\right), \tag{10}$$

and characteristic function

$$\mathbb{E}\left[e^{-iuX_t}\right] = \left(\frac{1}{1-i\mu\nu u-\frac{\sigma^2\nu u^2}{2}}\right)^{t/\nu}. \tag{11}$$

The parameters describe the mean and sign of skewness, μ, the kurtosis ν and σ is a volatility parameter.
The density f_{VG} given the time increment Δ starting at x_t is given by:

$$\begin{aligned} f_{VG}(x,t+\Delta_t;x_t,t) &= \frac{2\exp\left(\frac{\mu z(x)}{\sigma^2}\right)}{\nu^{\Delta/\nu}\sqrt{2\pi}\sigma\Gamma(\Delta_t/\nu)}\left(\frac{z(x)^2}{\frac{2\sigma^2}{\nu}+\mu^2}\right)^{\frac{\Delta}{2\nu}-\frac{1}{4}} \\ &\quad \cdot K_{\Delta/\nu-1/2}\left(\frac{1}{\sigma^2}\sqrt{z(x)^2\left(\frac{2\sigma^2}{\nu}+\mu^2\right)}\right) \end{aligned} \tag{12}$$

where $z(x) = x - x_t - (r+\omega)\Delta_t$ and $K_\nu(z) = \frac{1}{2}\int_0^\infty y^{\nu-1}\exp\left(-\frac{z}{2}(y+y^{-1})\right)dy$ denotes the modified Bessel function of the second kind.
We now describe two methods for generating paths from the Variance Gamma process. The first approach is called the subordinator approach. To this end let $(G_t)_t$ be an increasing stochastic process, the subordinator, which is independent of a given Brownian motion $(W_t)_t$. The process

$$Y(t) := W(G(t)) \tag{13}$$

is called a process obtained by subordinating the Brownian motion $W(t)$. For the Variance Gamma process the process $(G_t)_t$ is a Gamma process.
We state the algorithm to obtain the next step in a sequential Monte Carlo procedure. This algorithm can be found in standard literature like [3, 4, 8].

1. Simulate $Z \sim \mathcal{N}(0,1)$
2. Simulate $G \sim \nu\Gamma(\Delta/\nu, 1)$
3. Set $X(t+\Delta) = X(t) + r\Delta + \mu\nu G + \sigma\sqrt{G}Z$

For risk neutral valuation it is common practice to use an additive adjustment such that the process $S(t)\exp(-rt)$ becomes a martingale. We denote the drift correction at

each time point by $\bar{\omega}(t)$.
If we wish to assure that this asset price process is a martingale we have to add the drift correction $\bar{\omega} = \frac{1}{\nu}\log\left(1 - \mu\nu - \frac{\nu\sigma^2}{2}\right)$. The discretization of the drift corrected process is given by:

$$X(t+\Delta_t) = X(t) + (r+\bar{\omega})\Delta_t + \mu G(t) + \sigma\sqrt{G(t)}Z \tag{14}$$

Another method which has not been used so far is the inversion method since it is computationally expensive. But we expect that using multi-core or GPU applications this method may gain popularity. To numerically invert the cumulative distribution function in case of a Variance Gamma process for a fixed spot s we consider the function arg given by:

$$arg(z) = -s^{-iz}/iz + \varphi(z)$$

Then, for a real number c the cumulative distribution is given by:

$$F(x) = \frac{1}{\pi}\int_{-\infty}^{x} \mathcal{R}\left(arg(z+ic)\right) dx$$

The constant c can be chosen such that the integral can be evaluated conveniently. It corresponds to shifting the integration path into the complex plane. Finally, the inversion can be done by a zero search. F^{-1} is:

$$F^{-1}(y) = \inf\{x : F(x) - y = 0\} \tag{15}$$

The figure 1 illustrates the method of numerical inversion for different values of σ. Now, to apply our method we can proceed as follows:

- Simulate the subordinator $(G_t)_t$. Then, for each path we consider the discrete set $G(t)$, $t \in \mathcal{T} = \{t_0, G(t_1), \ldots, G(t_N)\}$ as a simple grid. Now, consider two consecutive grid points $G(t_j)$ and $G(t_{j+1})$. We denote the step size by Δ_j^M. With respect to (14) the increment $X(G(t_{j+1})) - X(G(t_j))$ is a Gaussian random variable and we can use the inverse method for Gaussian variables and the probabilities for breaching a lower and and upper level can be determined using the normal distribution.

- In this case we use (15) to determine if the generated random variate is feasible, that is, the process stays within the strip given by the lower and the upper bound. The probability weights for the importance sampling can be computed by numerically integrating the density given by (13).

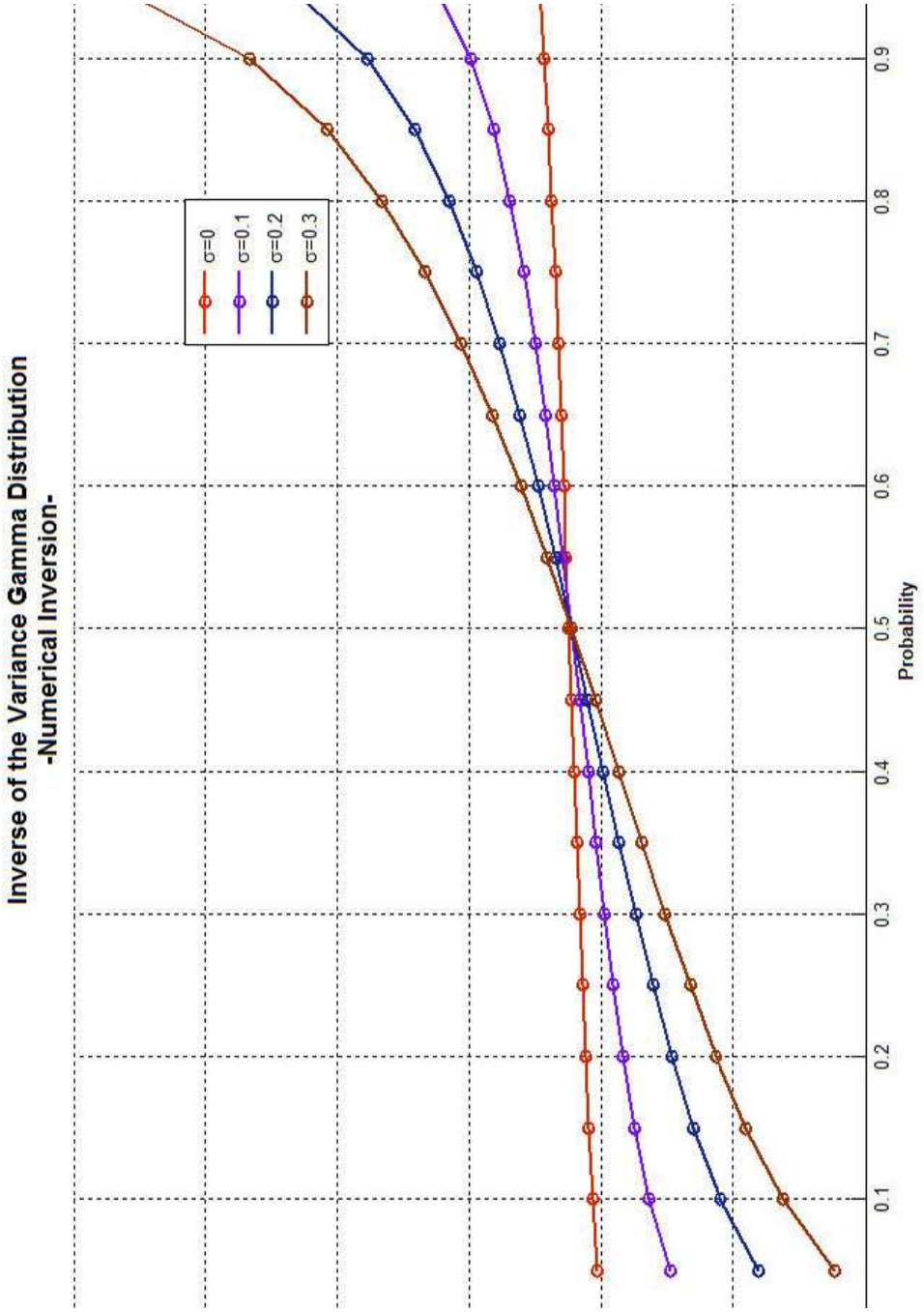

Figure 1: Inverse Cumulative Distribution Function for a Variance Gamma Distribution for different values of σ. $\sigma = 0$ (red), $\sigma = 0.1$ (purple), $\sigma = 0.2$ (blue) and $\sigma = 0.3$ (brown)

5 Numerical Results

5.1 Numerical Results: Robust Stress Testing

In this section we show how to use the MCBC in the context of stress testing. A stressed model is simply a model with large parameters (e.g., large volatilities). In such cases the paths of the Monte-Carlo simulation can exhibit bad sampling properties. The prototypical case is that of a large instantaneous volatility. Here, the drift will pull almost all paths to zero, while a few of will attain very large values.

If we consider a call option, the only those paths with large values will contribute.

In this setup a Monte-Carlo simulation with an upper and lower boundary condition dramatically improves the stability and convergence of the valuation algorithm. The lower boundary will keep paths away from zero, and hence, for the call option it can be used as an importance sampling. The upper boundary will essentially give the analytic valuation of the call option, completely eliminating the Monte-Carlo error.

5.1.1 Test Case

As a test case we will evaluate a simple product under a simple model: We evaluate a European call option under a Black-Schoes model. Nevertheless we will use a fully general implementation, where the model is simulated using Monte-Carlo simulations with fine time-stepping. Thus, the implementation is also adequate for more complex products (e.g., featuring multiple conditional payoffs) and more complex models (e.g., featuring local or stochastic volatilities).

The restriction to a European call under a Black-Scholes models allows us to benchmark the results against the analytic solution.

The following results show the effect of the location of the boundary and the type of the boundary condition.

5.1.2 Single Time Step

The test case of a simulation consisting of just a single time step is appears to be a fairly trivial one. In this case, the boundary condition is just the integral of a part of the payoff function. Hence, the boundary condition can be evaluated analytically. Since the payoff of our option is linear, the linear boundary condition is just the exact (analytic) solution. In other words: For the single time step we know the boundary estimates analytically and there are no approximation errors resulting from approximating the value function at the boundary.

In this case, the Monte-Carlo simulation with boundary conditions is simply a splitting of the valuation to a regime where Monte-Carlo sampling is used (inbound region) and to a regime where analytic valuation is used (out bound region).

However, this test case illustrates that the defect of the standard Monte-Carlo simulation clearly stems from the paths sampling the out-bound region, see Figure 5. In addition, we see that the simple super hedge boundary condition can give fairly good results, depending on the location of the barrier, see Figure 2 and Figure 3. Using a higher barrier will improve the results for the super-hedge boundary condition, but will also retain the convergence of the method, while the Monte-Carlo error is only

slightly higher. The numerical calculation of the boundary value extrapolation using a simple regression gives a result which is consistent to the limit case of a PDE. We do not converge to the correct value since the linear regression does a global fit to the non-linear payoff; however, the result is still an improvement over the plain Monte-Carlo simulation, see Figure 4. It can be improved further using other methods to numerically calculate the boundary value extrapolation.

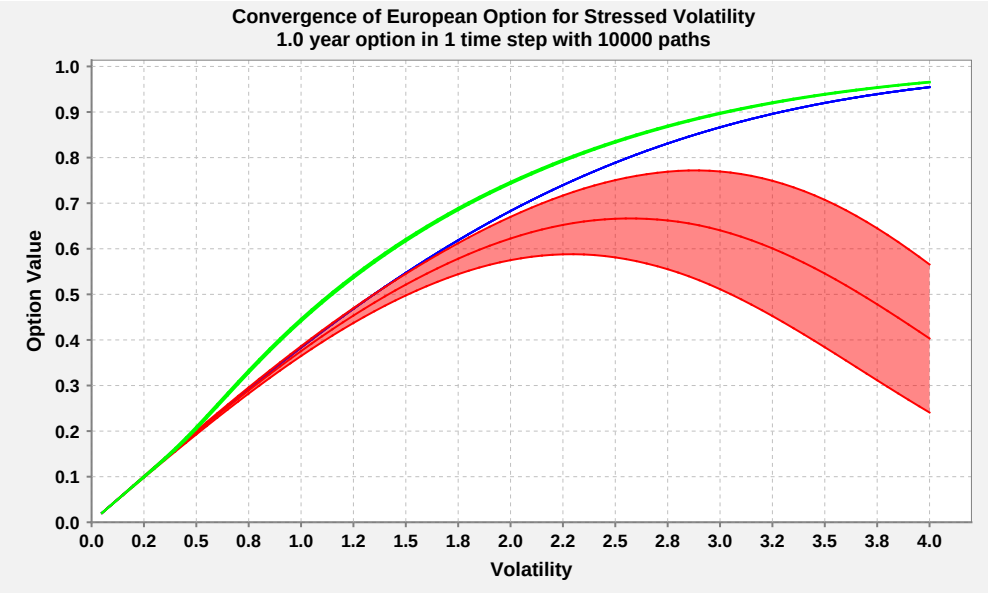

Figure 2: European option under Black-Scholes model, super-hedge boundary condition, 1 year option in 1 time step: Standard Monte-Carlo valuation (red), Monte-Carlo valuation with super-hedge boundary condition at barrier = 3 (green) and analytic benchmark (blue).

5.1.3 Multiple Time Steps with Constant Barrier

A more representative test case is that of a Monte-Carlo simulation with very fine time-stepping. In this case our linear boundary condition will be an approximation, because we are neglecting the convexity of the value function $V(t)$ for $0 \leq t < T$.

As expected, the sub-hedge boundary condition gives a lower bound for the option value. While the lower bound is crude (which should be expected, since the boundary estimate is crude), the results remain stable and convergent, see Figure 6. Increasing distance from the barrier improves the estimate of the lower bound, compare Figure 6 to Figure 7.

Similarly, the super-hedge boundary condition gives an upper bound for the option value, see Figure 8.

The linear boundary condition, where we calculated the linear extrapolation analytically, gives a very good result, see Figures 9, 10.

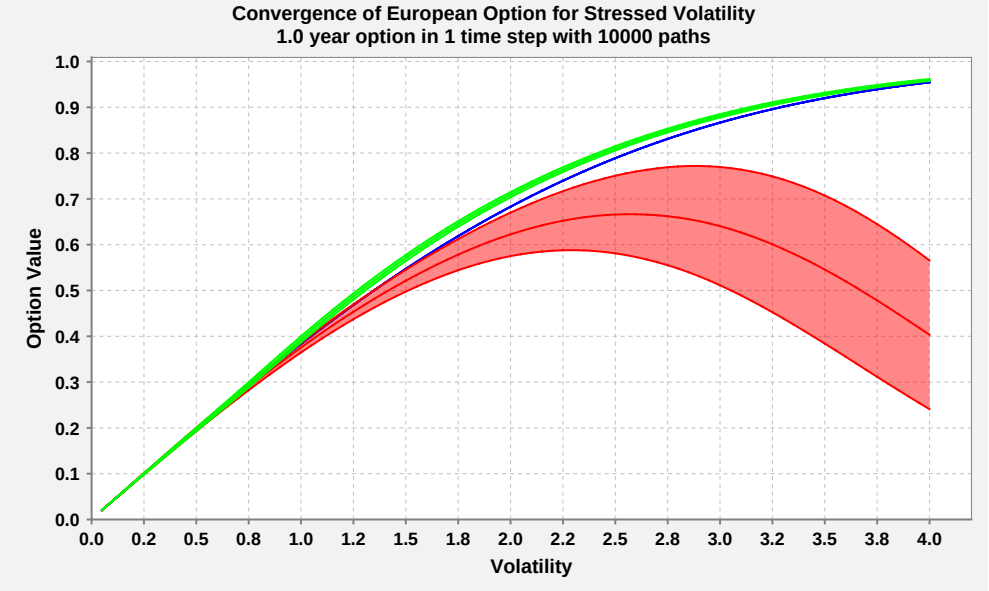

Figure 3: European option under Black-Scholes model, super-hedge boundary condition, 1 year option in 1 time step: Standard Monte-Carlo valuation (red), Monte-Carlo valuation with super-hedge boundary condition at barrier = 6 (green) and analytic benchmark (blue).

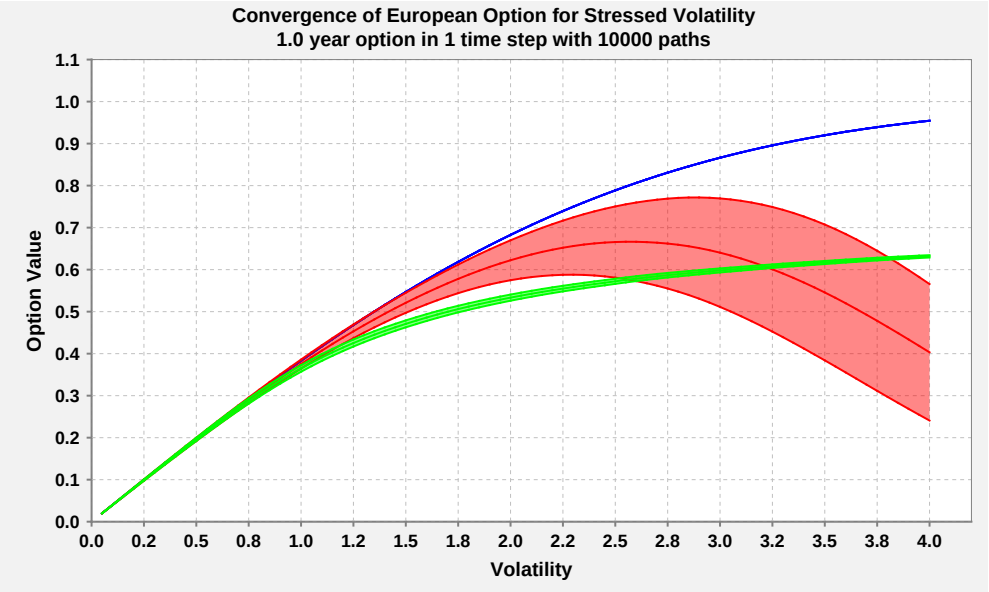

Figure 4: European option under Black-Scholes model, linear (numerically) boundary condition using plain regression, 1 year option in 1 time step: Standard Monte-Carlo valuation (red), Monte-Carlo valuation with super-hedge boundary condition at barrier = 8 (green) and analytic benchmark (blue).

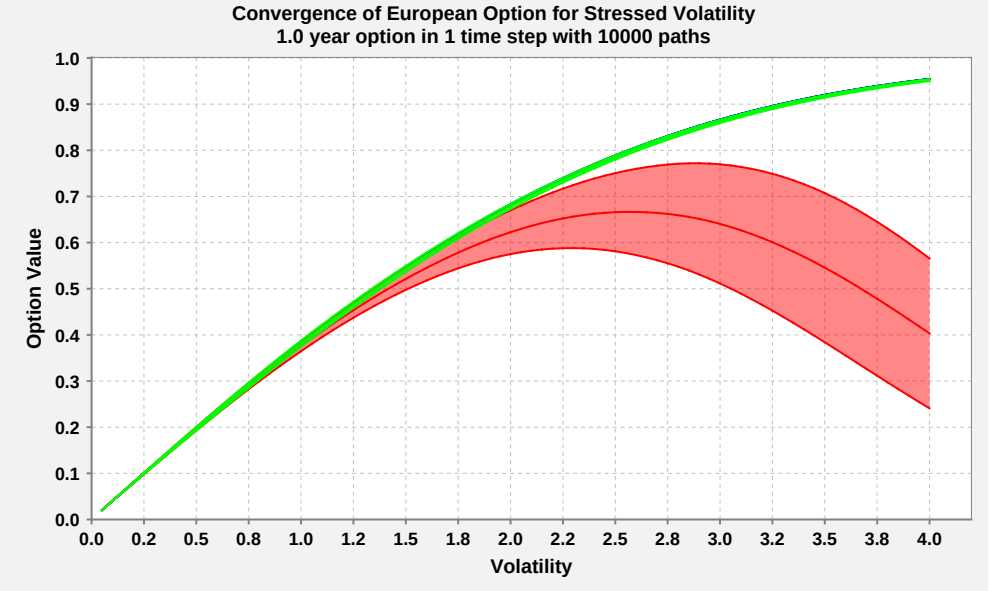

Figure 5: European option under Black-Scholes model, linear (analytic) boundary condition, 1 year option in 1 time step: Standard Monte-Carlo valuation (red), Monte-Carlo valuation with linear (analytic) boundary condition at barrier = 6 (green) and analytic benchmark (blue).

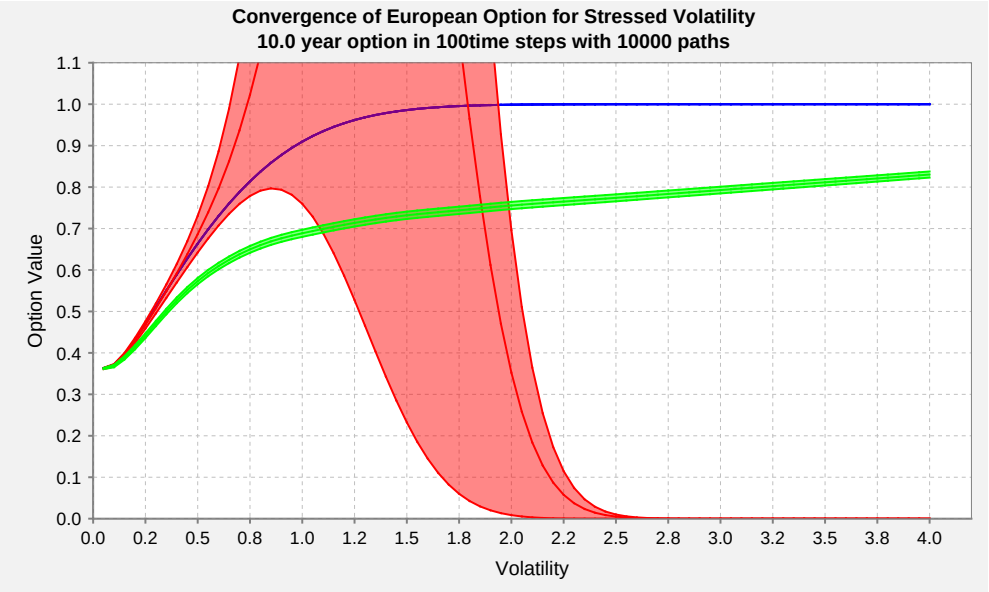

Figure 6: European option under Black-Scholes model, sub-hedge boundary condition, 10 year option in 100 time steps: Standard Monte-Carlo valuation (red), Monte-Carlo valuation with sub-hedge boundary condition at barrier = 3 (green) and analytic benchmark (blue).

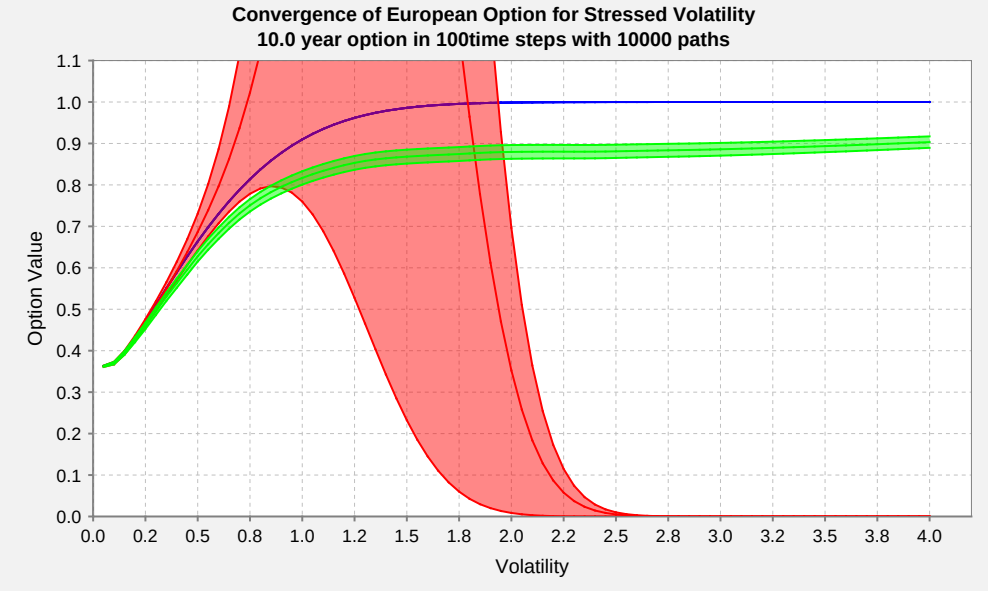

Figure 7: European option under Black-Scholes model, sub-hedge boundary condition, 10 year option in 100 time steps: Standard Monte-Carlo valuation (red), Monte-Carlo valuation with sub-hedge boundary condition at barrier = 6 (green) and analytic benchmark (blue).

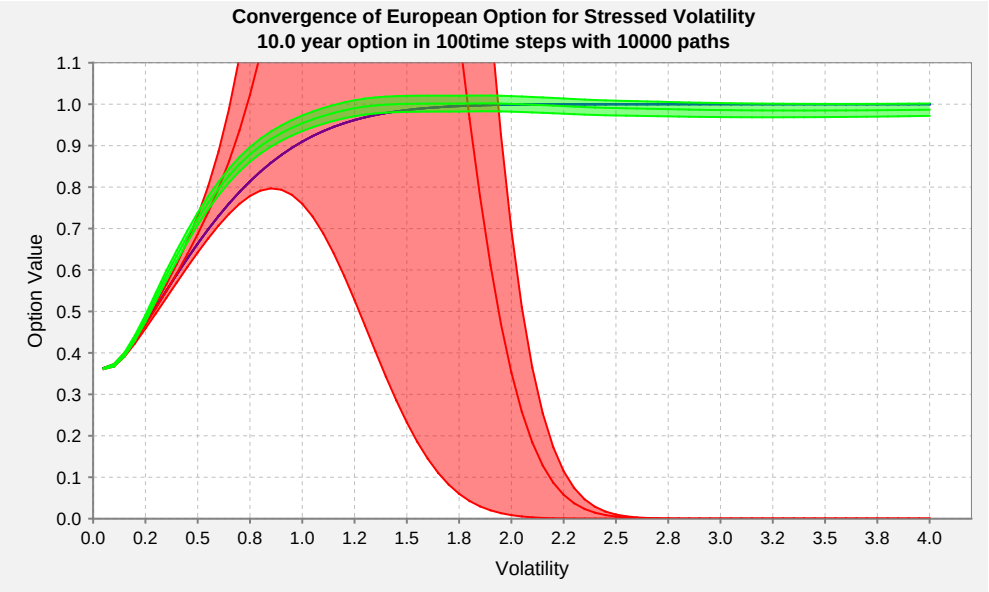

Figure 8: European option under Black-Scholes model, super-hedge boundary condition, 10 year option in 100 time steps: Standard Monte-Carlo valuation (red), Monte-Carlo valuation with super-hedge boundary condition at barrier = 6 (green) and analytic benchmark (blue).

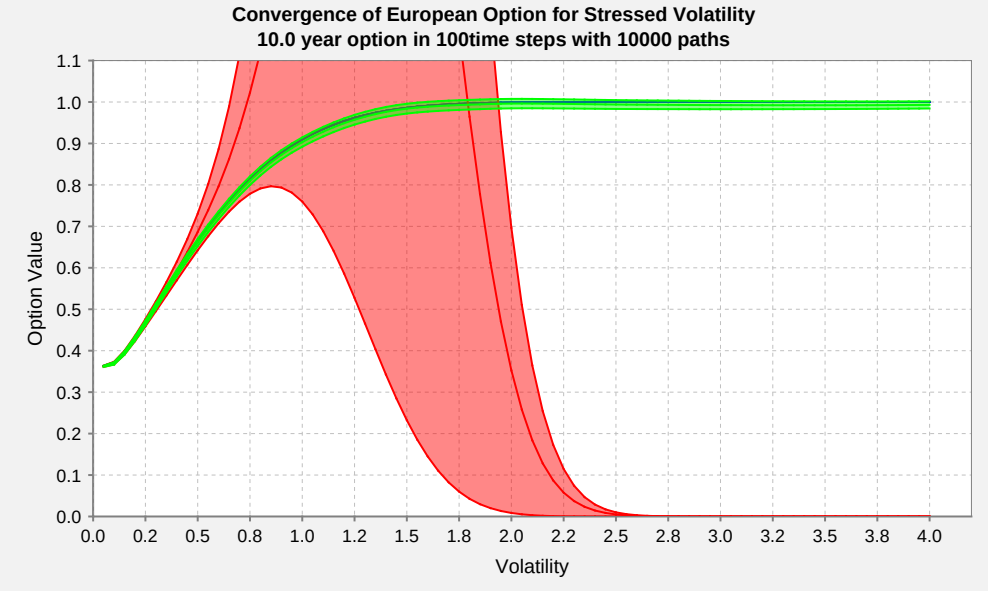

Figure 9: European option under Black-Scholes model, linear (analytic) boundary condition, 10 year option in 100 time steps: Standard Monte-Carlo valuation (red), Monte-Carlo valuation with linear (analytic) boundary condition at barrier = 6 (green) and analytic benchmark (blue).

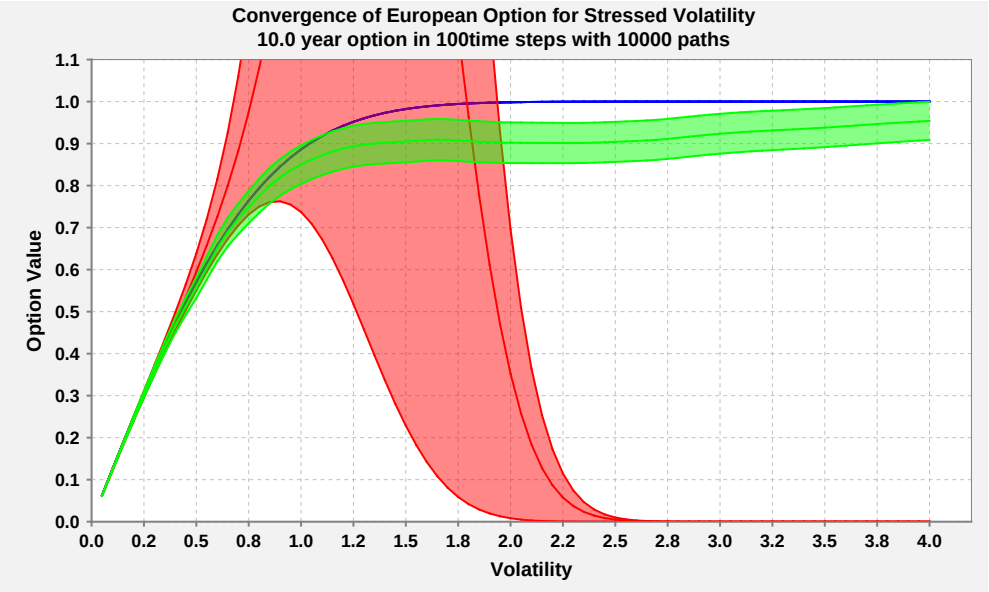

Figure 10: European option under Black-Scholes model, linear (analytic) boundary condition, 10 year option in 100 time steps: Standard Monte-Carlo valuation (red), Monte-Carlo valuation with linear (analytic) boundary condition at barrier = 40 (green) and analytic benchmark (blue).

5.1.4 Numerical Calculation of the Boundary Value Extrapolation using Regression

The boundary value extrapolation can be determined fully numerically without any assumption about the product under consideration. This is maybe the closest analog to the case of a PDE. However, you cannot trust this method blindly. For example, Figure 11 shows that the regression simply gives a wrong estimate as the Monte-Carlo simulation degenerates. So the Monte-Carlo simulation with boundary condition does not show a huge Monte-Carlo error, but the valuation is wrong due to the false boundary value extrapolation.

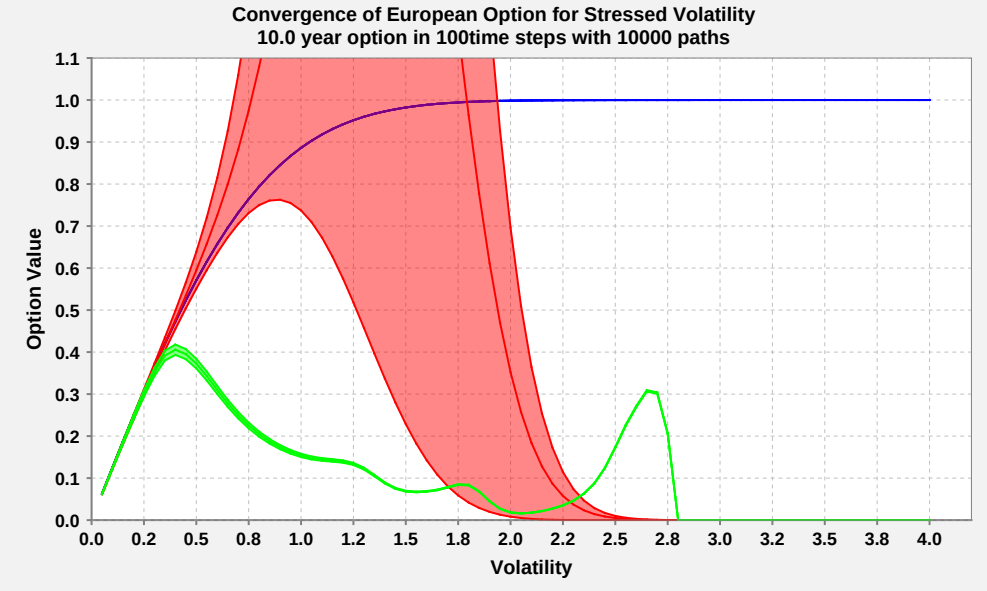

Figure 11: European option under Black-Scholes model, linear (numerical) boundary condition using regression, 10 year option in 100 time steps: Standard Monte-Carlo valuation (red), Monte-Carlo valuation with super-hedge boundary condition with time dependent barrier (green) and analytic benchmark (blue).

5.1.5 Multiple Time Steps with Time Dependent Barrier

Since the location of the barrier is important, the obvious generalization is a time-dependent barrier. For local or stochastic volatility models a stochastic barrier can be considered.

An example of a time dependent barrier is given by

$$b(t) \ := \ K(t) \exp(b_0 \sigma_0 \sqrt{(t)}),$$

where K is, in our case, the forward $K(t) = S(0) \exp(rt)$ and σ_0 is a fixed reference volatility and b_0 is a constant factor.

In Figures 12 and 13 we show the result for $b_0 = 4$ and $\sigma_0 = 0.25$. In other words, considering the base scenario of 25% volatility the boundary is located four standard

deviations away from the forward, in log-coordinates. Of course, this choice is to some extent arbitrary and the desired choice depends on the application.

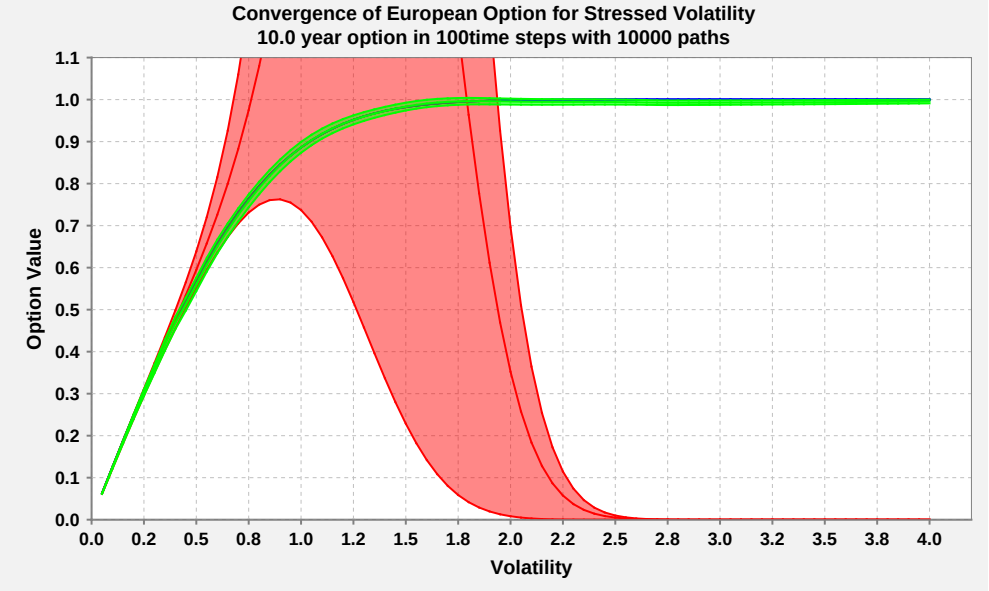

Figure 12: European option under Black-Scholes model, linear (analytic) boundary condition, 10 year option in 100 time steps: Standard Monte-Carlo valuation (red), Monte-Carlo valuation with linear (analytic) boundary condition with time dependent barrier (green) and analytic benchmark (blue).

5.1.6 Conclusion

Figure 13 shows that the super hedge boundary condition is a very promising choice when considering stress tests. It gives a stable upper bound for the true value having low Monte-Carlo error and the upper bound is as sharp as the original Monte-Carlo simulation when the model in in its non-stressed region.

It appears tempting to have the barrier depend on the model volatility σ instead of using a fixed reference volatility σ_0. For the application of stress tests this choice is counterproductive. Note that for a stress test we would like to have our model use the robust Monte-Carlo simulation if the model parameter is around some reference level σ_0 and we like to switch to analytic approximation if we are far beyond this reference level. Hence, it is desirable that the boundary should not be adapted to all model parameters.

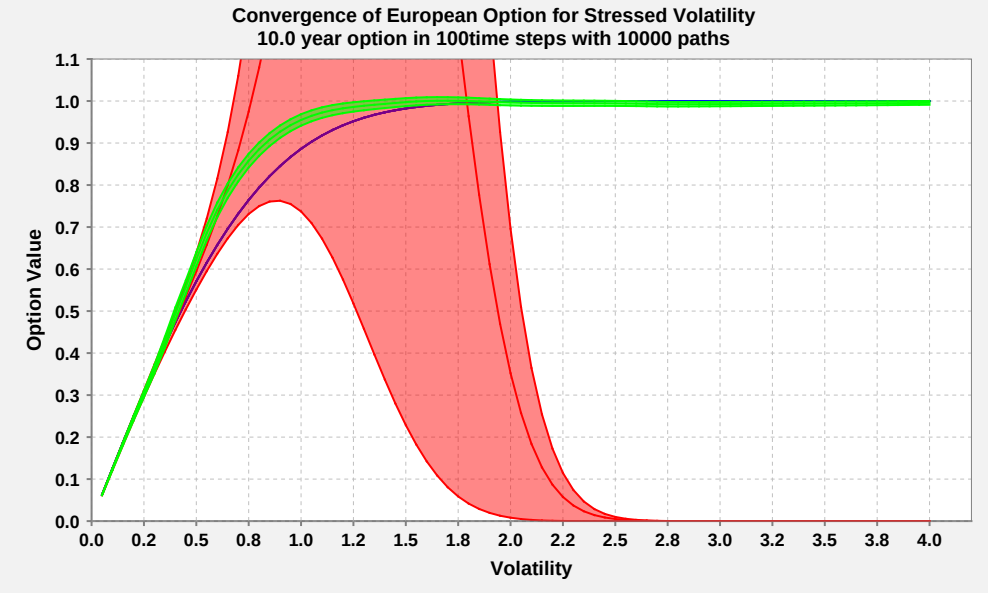

Figure 13: European option under Black-Scholes model, super-hedge boundary condition, 10 year option in 100 time steps: Standard Monte-Carlo valuation (red), Monte-Carlo valuation with super-hedge boundary condition with time dependent barrier (green) and analytic benchmark (blue).

5.2 Numerical Results for the CEV model

5.2.1 Test Case

We will consider the valuation of a put option using a CEV model with different model parameters. Depending on model parameters and model boundary conditions the behavior of a numerical scheme at the boundary $\{S = 0\}$ is critical for the valuation. We illustrate that we can stick to a simple Euler scheme for the CEV model using a Monte-Carlo simulation with boundary condition with a boundary at $\{S = \epsilon\}$, namely

$$S(t + dt) \;=\; S(t) + \sigma S(t)^{\beta}\sqrt{dt}Z \qquad \text{restricted to } S \geq \epsilon > 0. \tag{16}$$

As we have remarked earlier, there are different formulas for pricing call and put options depending on the coefficient β. For $\beta \geq 0.5$ the model has a unique solution. For $0 < \beta < 0.5$ the solution is not unique and its behavior in $S = 0$ needs to be specified in terms of a boundary condition. Two such condition are "absorbing" boundary and "reflecting" boundary and therefore there are two different valuation formulas.

In financial modeling it is common to use absorbing boundary conditions. We consider the CEV model with absorbing boundary conditions and three different scenarios. Common to all scenarios are spot price $S(0) = 0.03$, strike $K = 0.03$, and maturity $T = 1$. Then, we consider:

Scenario 1	Scenario 2	Scenario 3
$\beta = 0.1$	$\beta = 0.3$	$\beta = 0.5$

Thus, we consider values for β where boundary conditions have to be specified ($\beta = 0.1$,$\beta = 0.3$) and the case where the process solution is unique, but the process may reach 0 ($\beta = 0.5$). For each scenario we consider different values for the implied volatility, namely $\sigma = 0.1$, $\sigma = 0.3$ and $\sigma = 1.0$. The location of the boundary for the Monte-Carlo scheme with boundary condition was taken to be $\epsilon = 0.0001$.

We consider the valuation using several schemes suggested for discretizing the CEV model, namely

$$
\begin{aligned}
S(t+dt) &= S(t) + \sigma |S(t)|^{\beta} \sqrt{dt} Z && (17)\\
S(t+dt) &= |S(t)| + \sigma |S(t)|^{\beta} \sqrt{dt} Z && (18)\\
S(t+dt) &= S(t) + \sigma \max(S(t),0)^{\beta} \sqrt{dt} Z && (19)\\
S(t+dt) &= \max(S(t),0) + \sigma \max(S(t),0)^{\beta} \sqrt{dt} Z && (20)\\
S(t+dt) &= S(t) + \sigma S(t) \min(\epsilon^{(\beta-1)}, S(t)^{(\beta-1)}) \sqrt{dt} Z, \quad \epsilon > 0 && (21)
\end{aligned}
$$

The schemes are called reflect1 (17), reflect2 (18), absorb1 (19), absorb2 (20), and lcev (21). Note that these scheme differ only in their treatment of the boundary and that they are equivalent to (16) on $S(t) \geq \epsilon > 0$.

In Table 1 we list the calculated prices for the schemes absorb1, absorb2 and for the Monte-Carlo scheme with boundary condition.

MC absorb1(StdErr)	MC absorb2(StdErr)	MC boundary	Analytic
0.0242(0.0002)	0.0207(0.0001)	0.0207(0.0000)	0.0207
0.0535(0.0004)	0.0239(0.0001)	0.0270(0.0000)	0.0271
0.1507(0.0013)	0.0266(0.0002)	0.0290(0.0000)	0.0292
0.0138(0.0001)	0.0126(0.0001)	0.0133(0.0001)	0.0134
0.0324(0.0002)	0.0223(0.0001)	0.0247(0.0000)	0.0248
0.0911(0.0004)	0.0268(0.0002)	0.0289(0.0000)	0.0290
0.0069(0.0000)	0.0069(0.0001)	0.0069(0.0000)	0.0068
0.0194(0.0001)	0.0175(0.0001)	0.0185(0.0000)	0.0184
0.0540(0.0003)	0.0266(0.0001)	0.0282(0.0000)	0.0283

Table 1: Monte-Carlo valuation of put option using different schemes: absorb1, absorb2 and Monte-Carlo simulation with boundary condition. We considered 100000 paths with 12 time steps each. Parameters are $\beta = 0.1$, 0.3 and 0.5 and $\sigma = 0.1$,0.3 and 1.0 in the CEV model.

The numerical results in Table 1 suggest that the Monte Carlo scheme with boundary conditions performs well in all parameter configuration. The chosen "standard Monte-Carlo scheme" absorb1 fails in some parameter configurations, because of its wrong representation of models boundary behavior. Our the scheme is comparable (yet a bit better) than the scheme absorb2.

The values for the other schemes are depicted in Figures 14 to 17. The schemes reflect1, absorb1 and lcev have a positive probability that S falls below 0. This fact leads to overpricing the put option. The schemes reflect2 and absorb2 and the

Monte-Carlo simulation with boundary condition do not show this behavior. The scheme reflect2 creates a different boundary behavior (namely reflecting). The scheme absorb2 creates the absorbing boundary behavior, but shows For theses schemes the correct boundary behavior is taken into account and the schemes produce more reliable results.

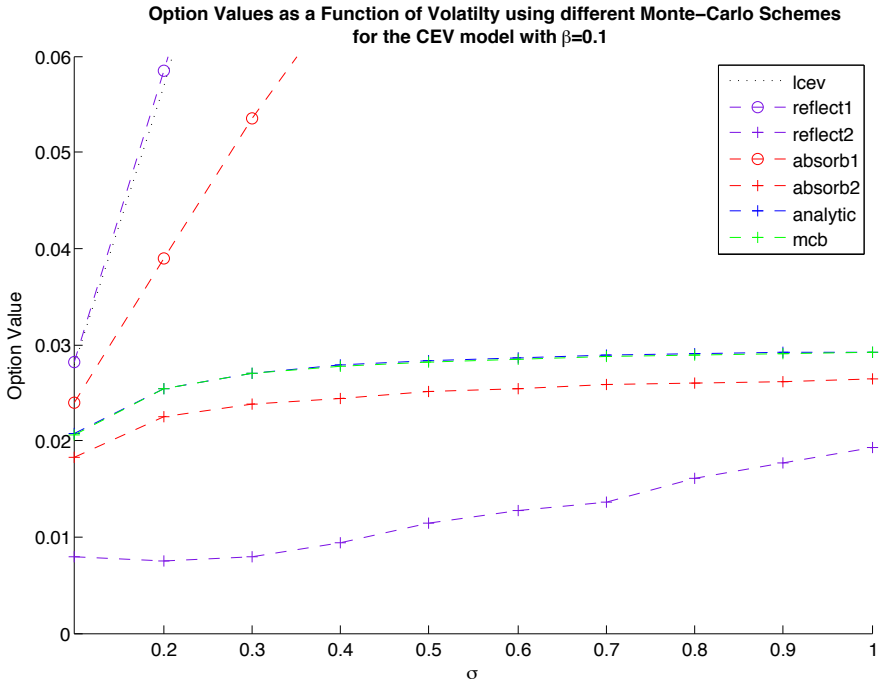

Figure 14: Results for $\beta = 0.1$ and different values of σ

For the range $0.5 < \beta \leq 1$ we observe that all the schemes tend to the analytic solution with the observation that for larger values of β the difference to the analytic solution gets smaller. Figure 17 illustrates this observation. Note however that, choosing values between 0.5 and 1.0 for β means that computing prices of digitals for applying the Monte Carlo with boundaries scheme involves calculating a non-central χ^2 distribution which is numerically very challenging. Therefore, certain parameter settings may cause some numerical bias in the Monte-Carlo results.

5.2.2 Conclusion

For our test of the CEV model we considered different schemes, namely (17), (18), (19), (20), (21) and compared them to the Monte-Carlo simulation with boundary condition. While all schemes agree on the inbound region $\{S > \epsilon\}$ their option pricing differs significantly. The schemes (17), (18), (19), (20), (21) inherently have a boundary assumption as part of the scheme and it is not immediately clear how well the models boundary behavior is captures by the numerical scheme. However, the corresponding Monte-Carlo simulation with boundary condition uses always the same (simple) Monte-Carlo scheme in the inbound region and the boundary behavior is provided separately

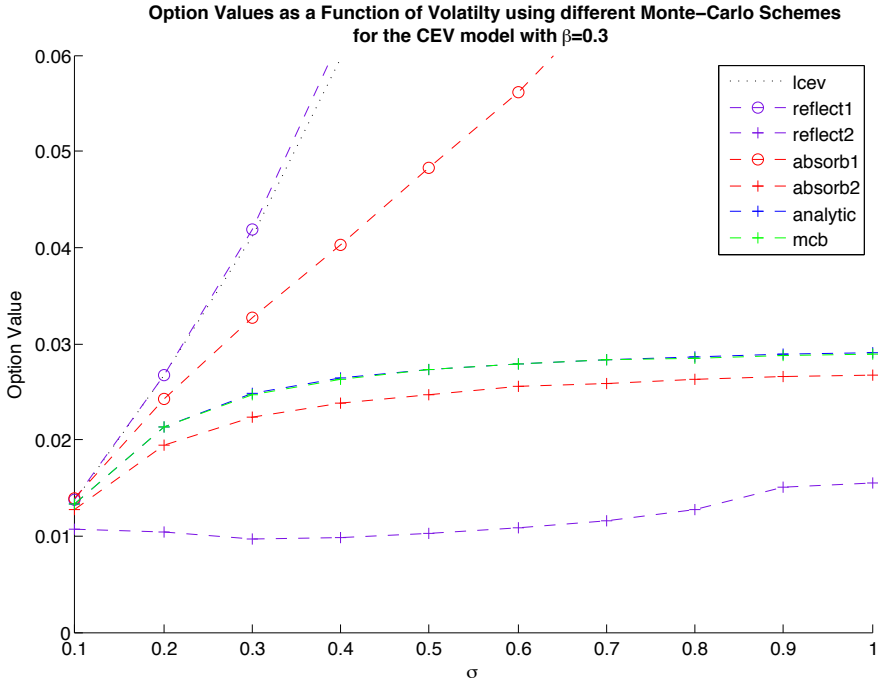

Figure 15: Results for $\beta = 0.3$ and different values of σ

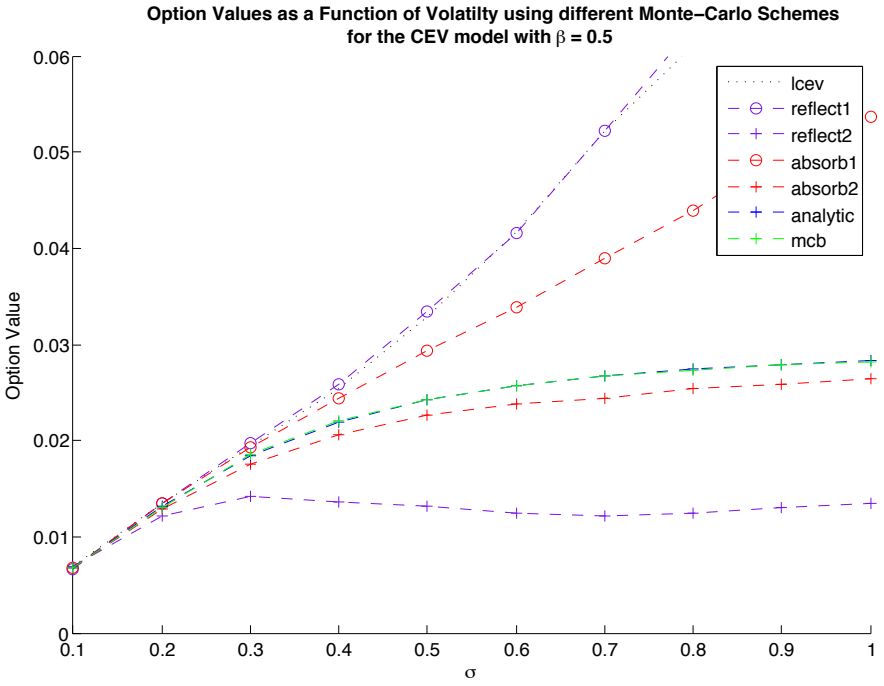

Figure 16: Results for $\beta = 0.5$ and different values of σ

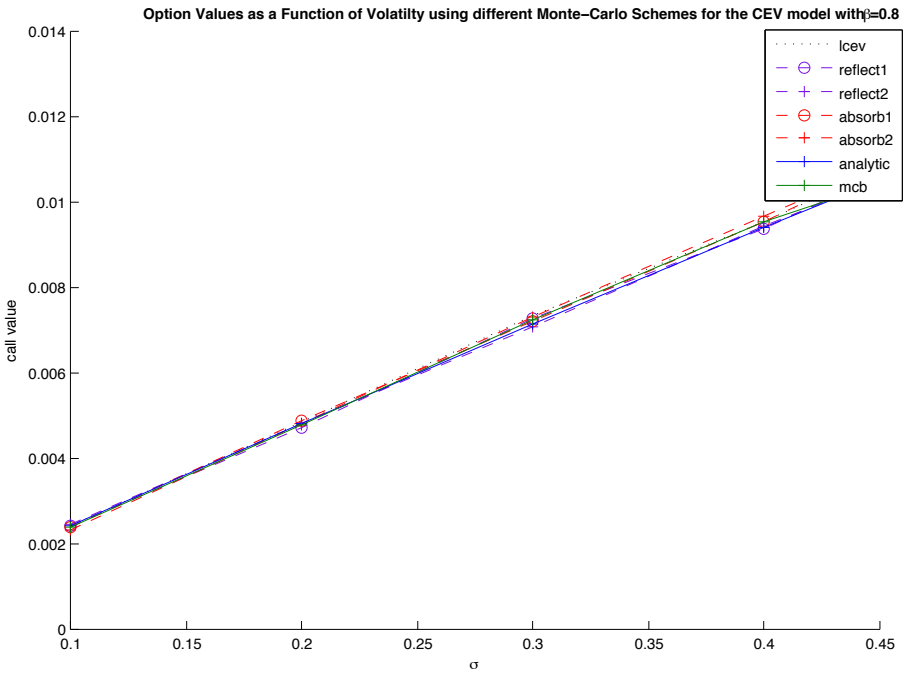

Figure 17: Results for $\beta = 0.8$ and different values of σ

by the analytic formula for the boundary condition.

The Monte-Carlo simulation with boundary gives the best result or is close to the scheme giving the best result.

The advantage of the Monte-Carlo simulation with boundary thus is that we can stick to a simple scheme and specify the correct boundary condition independently from the chosen discretization scheme. This is especially advantageous if a discretization scheme compatible with the boundary conditions is not known.[11]

5.3 Numerical Results for the VG model

5.3.1 Test Case

We consider the Variance Gamma model and three different scenarios, namely $S(0) = 0.03$, $K = 0.01$, $T = 1$, $\theta = -0.2094$, $\nu = 0.2570$ and

Scenario 1	Scenario 2	Scenario 3
$\sigma = 0.1325$	$\beta = 1.1325$	$\beta = 2.1325$

For each scenario we consider different values for implied volatility, namely $\sigma = 0.1$, $\sigma = 0.75$ and $\sigma = 1.15$.

[11] In our example, the best standard scheme for reflecting boundary conditions is (18) and the best standard scheme for absorbing boundary conditions is (20).

5.3.2 Result

The result is summarized in the following Table 5.3.2.

MC1(StdErr)	MC2(StdErr)	Analytic
Call		
5.04878(0.01683)	5.05930 (0.01552)	5.06930
20.59659(0.15981)	20.52839 (0.09612)	20.51896
31.38604(0.37417)	55.28593 (0.14913)	54.64931
Put		
2.89363(0.01683)	2.98120 (0.01552)	2.99120
18.11482(0.15981)	18.45029 (0.09612)	18.44086
28.97055(0.37417)	53.20783 (0.14913)	52.57120

Table 2: Results for the simulation of the VG process. MC1: Monte Carlo simulation with 100000 path and 12 time steps (Euler Method), MC2: Monte Carlo simulation with Boundary conditions with 100000 paths and 12 time steps, Call and Put give the analytic prices for the corresponding option.

5.3.3 Conclusion

The numerical results suggest that the proposed Monte Carlo scheme is much more efficient and applicable to a wide range of parameters. Even for stressed parameters in the Variance Gamma setting we obtain robust results. But we have to remark that this is applicable only in the case where we are able to compute stable analytic prices for call, puts and digitals. To this end we remark that the method is not applicable if we further stress the parameter σ. For $\sigma >$ we no longer obtain stable analytic solutions. Therefore, the method does not apply for these parameters.

6 Conclusions

We presented a general framework for creating Monte-Carlo simulations with boundary conditions. For the definition of the boundary value process we considered a linear boundary value functional. For the calculation of the boundary value functional we proposed

- Analytic calculation, which can be performed only for a very small class of products.
- A super- and sub-hedge, which are often easy to obtain and can server to guarantee that the valuation gives upper or lower bounds.
- A regression to estimate the boundary functional as it is done for Bermudan option pricing in Monte-Carlo.

Among the applications of the Monte-Carlo simulation with boundary conditions we presented two:

- valuing derivatives under stressed model parameters (stress testing).
- modes with boundary behavior resulting in biased discretization schemes.

The method presented is generic and (to some extent) product independent. It can be set up for a large class of models. We have given two examples, namely the CEV and the Variance Gamma model. We may further extend the method to models which can be represented as subordinated Brownian motion such as the NIG or even stochastic volatility Lévy models such as the Variance Gamma model with Gamma-Ornstein-Uhlenbeck or CIR clock.

With respect to the application to stress tests: We have illustrated how a Monte-Carlo simulation with boundary conditions can be employed to allow robust stress tests on a portfolio of complex products within a possibly complex Monte-Carlo model.

With respect to CEV and Variance Gamma models, we have shown how the boundary condition is derived from analytic formulas for call and digital options, given the model under consideration, to define a boundary condition which excludes the biased boundary behavior of the numerical scheme.

A drawback of the method is, of course, that the calculation of the boundary value process results in additional computational costs. In the cases where only rough approximations are sufficient (e.g., a very rough sub- oder super-hedge boundary condition) this effect can be reduced and the computational costs of a Monte-Carlo simulation with boundary conditional are comparable to a Monte-Carlo simulation with importance sampling. However, even the numerical calculation of boundary conditions using numerical integration or FFT could still be considered as computational power becomes cheaper.

References

[1] ANDERSEN, L; ANDREASEN, J.: Volatility skews and extensions of the Libor market model. Applied Mathematical Finance, 7(1), 247-270 (2000)

[2] BROADIE, MARK; KAYA, OZGUER: Exact Simulation of Stochastic Volatility and other Affine Jump Diffusion Processes. Operations Research, 2006, Vol. 54, No. 2, 217-231.
http://ssrn.com/abstract=1331904.

[3] DUFFY, D; KIENITZ, J.: Monte Carlo Frameworks. Building customizable high performance applications in C++, Wiley, Chichester, 2009

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

[5] FRIES, CHRISTIAN P.; JOSHI, MARK S.: Partial Proxy Simulation Schemes for Generic and Robust Monte-Carlo Greeks. *Journal of Computational Finance*, 12-1. (2008).
http://www.christian-fries.de/finmath/proxyscheme.

[6] FRIES, CHRISTIAN P.; JOSHI, MARK S.: Conditional Analytic Monte Carlo Pricing Scheme for Auto-Callable Products.
http://www.christian-fries.de/finmath/montecarlo4trigger.

[7] FRIES, CHRISTIAN P.; KAMPEN, JÖRG: Proxy Simulation Schemes for generic robust Monte Carlo sensitivities, process oriented importance sampling and high accuracy drift approximation. *Journal of Computational Finance*, 10-2, p. 97-128. (2006).
http://www.christian-fries.de/finmath/proxyscheme

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

[9] GLASSERMAN, PAUL; STAUM, JEREMY: Conditioning on one-step survival in barrier option simulations. *Operations Research*, 49:923?937, 2001.

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

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

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

[13] LORD, ROGER; KOEKKOEK, REMMERT;VAN DIJK, DICK: A comparison of biased simulation schemes for stochastic volatility models. *Quantitative Finance*, vol. 10, no. 2, pp. 177-194.

[14] LESNIEWSKI, A.: Notes on the CEV model, Working paper http://lesniewski.us/papers/working/NotesOnCEV.pdf

[15] MADAN, D.; SENETA, E.: The V.G. model for share market returns, Journal of Business, 63, 511-524 (1990)

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

Notes

Suggested Citation

FRIES, CHRISTIAN P.; KIENITZ, JÖRG: Monte-Carlo Simulation with Boundary Conditions. (2010).
http://www.christian-fries.de/finmath/montecarlowithboundary

Classification

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

Keywords: Monte Carlo Simulation, Valuation, Stress Test, Variance Reduction, Boundary Conditions, Numerical Schemes, CEV, Variance Gamma

40 pages. 17 figures. 1 tables.