ERASMUS UNIVERSITY ROTTERDAM
Erasmus School of Economics

Reducing simulation bias for exotic options with different price processes[1]

Master Thesis Quantitative Finance

Thomas Geissler (445907)
445907tg@student.eur.nl

Supervisor:
Dr. R. Lange

Second assessor:
Prof. Dr. C. Zhou

Date final version:
October 4, 2020

Abstract

This thesis proposes the Brownian bridge maximum method (B2M2) for the simulation of option prices as an alternative to more traditional Monte-Carlo (MC) simulation methods. Traditional MC methods poorly estimate path-dependent quantities, such as extremal values, by restricting simulated values to discrete time points. Conversely, B2M2 explicitly simulates each sub-interval's extremal value, thereby obtaining more accurate representations of e.g. barrier-crossing probabilities, even for relatively coarse time grids. Indeed, I find that B2M2 outperforms traditional MC methods in computing option prices for a wide range of underlying price processes, option types, and simulation times; furthermore, it does so efficiently, in the sense that both the ultimate error as well as the error per unit of (simulation) time is reduced. This feature makes the proposed method especially attractive in practice when speedy computations are required.

[1]The views stated in this thesis are those of the author and not necessarily those of Erasmus School of Economics or Erasmus University Rotterdam.

Contents

1 Introduction

The financial derivatives market is enormous, and the gross market value of all contracts was estimated at \$12.7 trillion in 2018, according to the Bank for International Settlements. Given this massive amount of money in the derivatives market, the valuation of the derivatives must be fast and accurate. Under certain assumptions, such as constant volatility and constant interest rate, we can price a standard European style option analytically using, for example, the Black-Scholes model. If we relax these assumptions (since they do not hold empirically) and look into more complicated options, no closed-form solution exists for its valuation. Instead, we simulate the stock price, volatility, interest rate, and other important drivers of the price process to calculate the option's price.

Monte-Carlo (MC) simulation is often the preferred simulation method as it can handle path-dependent options, such as a barrier, and early exercise options, like American options. However, due to the assumption of continuous price processes, the simulation is based on discretized intervals, which may lead to a bias in the option price if the interval is too large. One way to reduce this bias is to make the time interval as tiny as possible, but this comes at a price since simulation at smaller intervals takes much longer. Furthermore, MC leads to bias in exotic options that depend on the minimum or maximum stock price. Due to discretizing, we lose information on the price process between the discretized intervals. Suppose the stock price is a and we find a simulated stock price of b for the next interval, then we have no information on the process between a and b. Instead of decreasing the interval size, we can construct a Brownian bridge that connects a with b to get the information about the process between the two points.

Beaglehole, Dybvig, and Zhou (1997) propose an alternative method that uses Brownian bridges to simulate options based on the extreme value of the underlying. They find that the Brownian bridge maximum method (B2M2) is a more efficient solution for reducing the pricing bias than to decrease the interval size. They show that the bias for a Russian call option worth \$4.52 goes from 42 cents to only 2.5 cents. Clearly, in a market of more than twelve trillion dollars, such bias reductions help make the market more efficient. The downside to this approach is that the simulation procedure takes longer, but according to Beaglehole et al. (1997), the reduced bias more than makes up for the extra time. Furthermore, simulating with Brownian bridges converges to the analytical price much quicker than a standard simulation.

However, Beaglehole et al. (1997) do not go into detail about the generalized processes that are most often used in practice, such as stochastic volatility and interest rates. This paper focuses on the efficiency of B2M2 in a practical setting by applying this method to generalized processes and a much more extensive set of performance measures. I look into the geometric Brownian motion, the Cox, Ingersoll Jr, and Ross (1985, CIR) model, the Heston (1993) model, and the multivariate geometric Brownian motion. I investigate each of these processes by evaluating the bias, mean squared error, and necessary computation time. As stated before, for more general assumptions on the price processes and more complicated options, a closed-form solution does not always exist. Therefore, I start the analysis of each option with the geometric Brownian motion as we can use the Black-Scholes model to find the analytical price. After analyzing the geometric Brownian motion, I simulate the prices for the other cases and compare them to a converged value as if that is the actual price.

The results in this paper show that B2M2 reduces the bias for all options that I consider. Moreover, for the geometric Brownian motion, the CIR model, and the multivariate geometric Brownian motion, the method produces unbiased prices. The MC simulation consistently undervalues call options that depend on maximum stock prices, such as the Russian option, while it overvalues call options that depend on minimum stock prices, such as the down-and-out option. This method produces sample paths that do not reach the extremal values due to the processes' discretization. Although the MC simulation produces option prices more quickly, when we set the simulation time equal for both methods, B2M2 always has a lower bias. Furthermore, the mean squared error is consistently lower for B2M2. These statistics show that options that depend on extreme prices should always be priced using B2M2 in a practical application.

Using MC simulation to price options dates back to Boyle (1977). He shows that this approach leads to unbiased option prices but a high variance for standard call and put options. Papers such as Clewlow and Carverhill (1994) and Kemna and Vorst (1990) show how to efficiently reduce the variance using antithetic and control variates, while Barraquand (1995) introduced moment matching to reduce the variance. The idea behind antithetic variates is that the random draws are negatively correlated, such that the total variance shrinks. A control variate is a different random variable but with a known mean, such as the pay-off of a standard European call option, which is correlated to the random variable we are trying to estimate. A linear combination of the control variates and random variables results in an unbiased estimator with a smaller variance. Moment matching is based on obtaining the empirical mean and covariance matrix, which by the law of large numbers implies that they are arbitrarily close to the real mean and covariance matrix. A linear combination of the random draws, empirical mean, and covariance matrix results in an estimate with a significantly lower standard error.

More recent variance reduction techniques are based on (randomized) quasi-Monte Carlo (QMC) and exact discretization schemes. Whereas the standard MC simulation is based on random draws with the possibility of two draws being very close to each other, the QMC simulation is deterministic, and each draw is from a low-discrepancy sequence, which means that points are more uniformly distributed. Joy, Boyle, and Tan (1996) show that QMC converges to the exact price much quicker than standard MC. However, we lose randomness due to the low-discrepancy sequence, and it is complicated to obtain an error estimate. Randomized QMC shifts the low-discrepancy sequence randomly such that we regain the randomness and error estimations. Avramidis and L'Ecuyer (2006) show that their randomized QMC method significantly reduces the variance but increases the bias. Exact discretization schemes are models that capture continuous processes in discrete time. A well-known example of such a model is the Euler-Maruyama method to approximate stochastic differential equations. Andersen (2008) presents multiple algorithms to simulate price paths from a Heston model and shows that his quadratic-exponential (QE) scheme significantly reduces bias and variance while also being less computationally expensive.

While there has been much research on reducing the variance in MC simulation, the pricing of American and exotic options remains challenging as the standard simulation leads to biased estimates. Tilley (1993) is the first paper to use MC for American options, and it highlights the bias of this type of simulation. Beaglehole et al. (1997) introduce the use of Brownian bridges to decrease the bias for exotic options. Metwally and Atiya (2002) show that using Brownian bridges also leads to a much faster convergence and more accurate prices than MC. Carrière (1996) uses regression functions to approximate the option price and removes the bias through linear combinations of biased estimators. Longstaff and Schwartz (2001) generalize the methods of Carrière (1996) to least-squares Monte-Carlo (LSM). However, this estimation method leads to higher variance. Broadie and Glasserman (1995) use non-parametric bootstrapping to reduce the bias, but it comes at the cost of significantly increased computation time.

The remainder of this paper is organized as follows. Section 2 shows the different price processes and the corresponding assumptions of these processes. Section 3 explains the different types of exotic options and displays the corresponding analytical price when available. Section 4 presents the two simulation techniques that I examine in this paper. Section 5 gives an overview of how I compare the simulation techniques. Finally, section 6 presents the results of this paper, and section 7 concludes.

2 Underlying price processes

In this paper, I use four different underlying price processes: a standard geometric Brownian motion, the CIR model with a stochastic interest rate, the Heston model with stochastic volatility, and a multivariate geometric Brownian motion. I start by explaining the most basic process, the geometric Brownian motion. I continue by presenting the CIR and Heston model. Finally, I explain the multivariate geometric Brownian motion. Since we need to price options, we simulate the stock price under the risk-neutral measure.

2.1 Geometric Brownian motion

The geometric Brownian motion is the most widely used stochastic differential equation for stock prices. For example, the Black-Scholes model assumes that the prices follow a geometric Brownian motion. I assume that the stock price is log-normal, the risk-free rate is constant, and the volatility is proportional to the stock price. Under the risk-neutral measure, the price follows

$$\mathrm{d}S_t = rS_t\mathrm{d}t + \sigma S_t\mathrm{d}W_t, \tag{2.1}$$

where r is the risk-free rate, σ the volatility, and W_t a standard Brownian motion. Furthermore, I assume that at time $t = 0$, the stock price is $S_0 = s$.

2.2 Cox-Ingersoll-Ross model

One of the drawbacks of modeling a price process with a standard geometric Brownian motion is that the assumption of a constant risk-free rate is not realistic. Therefore, we need to relax this assumption and use a time-varying short rate. The CIR model is a one-factor model for the interest rate evolution, and it ensures mean-reversion and positive interest rates. The short rate process follows

$$\mathrm{d}r_t = \theta(\mu - r_t)\mathrm{d}t + \xi\sqrt{r_t}\mathrm{d}W_t^r, \tag{2.2}$$

where $\theta > 0$ corresponds to the speed of mean reversion, μ to the long-term mean of the short rate, ξ to the volatility of the interest rate process, and W_t^r to a standard Brownian motion. The new stock price process follows

$$\mathrm{d}S_t = r_tS_t\mathrm{d}t + \sigma S_t\mathrm{d}W_t^S, \tag{2.3}$$

where the parameters have the same interpretation as in section 2.1. The processes $\mathrm{d}W_t^r$ and $\mathrm{d}W_t^S$ are dependent with correlation ρ. I assume that at time $t = 0$, the interest rate is equal to $r_0 = \zeta$ and the stock price equal to $S_0 = s$.

2.3 Heston model

Another drawback of modeling a price process with a standard geometric Brownian motion is that the assumption of volatility proportional to the stock price is not realistic. One of the stylized facts of stock returns is that variances are not constant and are usually clustered. This clustering means that large changes usually follow more large changes, and small changes usually follow more small changes. A stochastic volatility model captures this stylized fact. The Heston model is a well-known model that describes the evolution of the volatility. Heston (1993) shows how the inclusion of stochastic volatility affects option prices compared to the usual Black-Scholes approach with a geometric Brownian motion. Following Heston (1993), the variance process evolves according to

$$d\nu_t = \theta\left(\mu - \nu_t\right) dt + \xi\sqrt{\nu_t} dW_t^{\nu}, \tag{2.4}$$

where $\theta > 0$ governs the speed of mean-reversion, μ determines the long-term mean of the variance, ξ is the volatility of the variance, and W_t^{ν} is a standard Brownian motion. The parameters θ, μ, and ξ are constants and are restricted to $2\theta\mu > \xi^2$ to ensure that the variance does not become negative. The price process now evolves according to

$$dS_t = rS_t dt + \sqrt{\nu_t} S_t dW_t^S, \tag{2.5}$$

where the parameters have the same interpretation as in section 2.1. The processes dW_t^{ν} and dW_t^S are dependent with correlation ρ. I assume that at time $t = 0$, the volatility is equal to $\nu_0 = v$ and the stock price equal to $S_0 = s$.

2.4 Multivariate geometric Brownian motion

Finally, we generalize the geometric Brownian motion to a multivariate version of the process. This version allows us to model multiple correlated stock price processes. I assume that the risk-free rate r, the volatility of each stock σ_i, and the covariance of asset i and j are constant for all $i, j = 1, \ldots, n$. Hence, the price process of n risky assets follows

$$d\begin{pmatrix} S_{1,t} \\ S_{2,t} \\ \vdots \\ S_{n,t} \end{pmatrix} = \begin{pmatrix} r & 0 & \ldots & 0 \\ 0 & r & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & r \end{pmatrix} \begin{pmatrix} S_{1,t} \\ S_{2,t} \\ \vdots \\ S_{n,t} \end{pmatrix} dt + \begin{pmatrix} \sigma_1 & \sigma_{12} & \ldots & \sigma_{1n} \\ \sigma_{21} & \sigma_2 & \ldots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \ldots & \sigma_n \end{pmatrix} d\begin{pmatrix} W_{1,t} \\ W_{2,t} \\ \vdots \\ W_{n,t} \end{pmatrix}, \tag{2.6}$$

with $W_{1,t}, \ldots, W_{n,t}$ independent standard Brownian motions. Thus each asset has a drift term equal to the risk-free rate and is exposed to shocks through $\sigma_{i1}, \ldots, \sigma_{in}$. In matrix form, the process follows

$$d\boldsymbol{S}_t = r\boldsymbol{I}_n\boldsymbol{S}_t dt + \boldsymbol{\Sigma} d\boldsymbol{W}_t, \tag{2.7}$$

where $\boldsymbol{I}_n$ is the $n \times n$ identity matrix and $\boldsymbol{W}_t$ an $n \times 1$ vector of independent standard Brownian motions. Finally, I assume that at $t = 0$, each asset's stock price is equal to $S_{i,0} = s_i$.

3 Option types

In this section, I present the options I investigate. I consider a Russian option, a knock-out option, and a knock-in option on a basket. These options depend on extreme price movements and should thus be of great interest when we want to reduce bias.

3.1 Russian options

A Russian option is a lookback option with a strike price of K, no maturity date, and it can be exercised at every price the stock has traded at. This specification means that a Russian call option can be seen as an option on the stock's maximum price, and a Russian put option as an option on the minimum price of the stock. Because it is impossible to simulate a time horizon that goes to infinity, the Russian options that I use have a specified maturity date $t = T$. With a specified maturity date, the pay-off function of a Russian call option at maturity for the stock price $V(S_T, T)$ is

$$V(S_T, T) = \max\left\{ \sup_{t\in(0,T)} (S_t - K), 0 \right\}. \tag{3.1}$$

If the process of S_t follows a geometric Brownian motion as in equation (2.1), it is possible to calculate the analytical option price. Conze and Viswanathan (1991) show that the price of an option based on the maximum value of the stock price is

$$C_{\max} = S_0\Phi(d_1) - e^{-rT}K\Phi(d_2) + e^{-rT}\frac{\sigma^2}{2r}S_0\left[-\left(\frac{S_0}{K}\right)^{\frac{-2r}{\sigma^2}}\Phi\left(d_1 - \frac{2r\sqrt{T}}{\sigma}\right) + e^{rT}\Phi(d_1)\right], \tag{3.2}$$

where

$$d_1 = \left(\ln\frac{S_0}{K} + rT + \frac{1}{2}\sigma^2 T\right)/\sigma\sqrt{T}$$
$$d_2 = d_1 - \sigma\sqrt{T},$$

and $\Phi(\cdot)$ is the standard normal cumulative distribution function. Thus the price of a Russian option is the value of a standard European call option plus a term that captures the value of the maximum at any point of the option's existence. To the best of my knowledge, analytical prices for a Russian option with an underlying CIR or Heston process are not available. Therefore, prices need to be calculated numerically. In this paper, I focus on the call option.

3.2 Knock-out options

The knock-out option is a group term for options that immediately become worthless when the underlying stock has exceeded a certain threshold. For example, the down-and-out option gives the holder the right to buy a stock for the strike price at the agreed-upon date of maturity, given that the stock price has not gone under a specific threshold. The value of the down-and-out call option with strike K and threshold $H < S_0$ at maturity T is

$$V(S_T, T) = \max\{S_T - K, 0\}\mathbb{I}\left[\inf_{t\in(0,T)} S_t > H\right], \tag{3.3}$$

where $\mathbb{I}[\cdot]$ is the indicator function and is 1 when the stock price did not reach the threshold H at any time, but is equal to 0 when at some point the price did go under the threshold. Suppose the price process follows a geometric Brownian motion as in equation (2.1), then it is possible to obtain an analytical option price. Rubinstein and Reiner (1991) show that the analytical price of a down-and-out call option is

$$C_{do} = S_0\Phi(x) - Ke^{-rT}\Phi(x - \sigma\sqrt{T}) - S_0\left(\frac{H}{S}\right)^{2\lambda}\Phi(y) - Ke^{-rT}\left(\frac{H}{S}\right)^{2\lambda-2}\Phi(y - \sigma\sqrt{T}), \tag{3.4}$$

with

$$\begin{aligned}
\lambda &= 1 + \frac{\mu}{\sigma^2} \\
\mu &= \ln(1 + r) - \frac{1}{2}\sigma^2 \\
x &= \left[\ln\left(\frac{S_0}{K}\right)/(\sigma\sqrt{T})\right] + \lambda\sigma\sqrt{T} \\
y &= \ln\left(\frac{H^2}{S_0 K}\right)/(\sigma\sqrt{T}) + \lambda\sigma\sqrt{T},
\end{aligned}$$

and $\Phi(\cdot)$ is the standard normal cumulative distribution function. Upon further inspection of this pricing formula, we see that the first part is the Black-Scholes price of a regular call option, and the second part is a correction for the possibility of the option becoming worthless. To the best of my knowledge, analytical prices for a down-and-out option with an underlying CIR or Heston process are not available. In this paper, I focus on the call option.

3.3 Knock-in basket options

Finally, the knock-in option on a basket is a group term for options that only have value when the weighted sum of the underlying stock prices exceeds a specific threshold. For example, an up-and-in basket option gives the owner the right to buy the specified assets for the strike price at the agreed-upon date of maturity, given that the weighted sum of stock prices has exceeded a specific price. The value of the up-and-in basket call option on a specified set of n assets with strike K and threshold $H > S_0$ at maturity T is

$$V(S_{1,T}, S_{2,T}, \ldots, S_{n,T}, T) = \max\left\{\sum_{i=1}^{n} w_i S_{i,T} - K, 0\right\}\mathbb{I}\left[\sup_{t\in(0,T)}\sum_{i=1}^{n} w_i S_{i,t} > H\right], \tag{3.5}$$

where w_i is the weight of asset i, and $\mathbb{I}[\cdot]$ is the indicator function and is 1 when the weighted sum of the stock prices did exceed the threshold H at least once, but is equal to 0 when the weighted sum of prices remained under the threshold. To the best of my knowledge, analytical prices for basket options do not exist and have to be calculated numerically. However, there are closed-form approximations for standard call options such as Gentle (1993) and Ju (2002). In this paper, I focus on the call option for $n = 3$ risky assets. For simplicity, I set the weighting function w_i equal to 1 for each asset, such that the basket option depends on the sum of all asset prices.

4 Simulation set-up

I perform two types of simulations; I start by explaining the standard Monte-Carlo simulation for the geometric Brownian motion, the CIR and Heston process, and the multivariate geometric Brownian motion. Next, I explain the simulation with Brownian bridges for each of the underlying price processes. For both simulations, I follow the methodology of Beaglehole et al. (1997).

4.1 Monte-Carlo simulation

Suppose the underlying asset follows a geometric Brownian motion as in equation (2.1), and we want to obtain the price of an option that expires at maturity date T. First, we have to divide the time-to-maturity into tiny intervals to get as close to a continuous-time process as possible. Suppose we want to divide the whole interval into I smaller intervals, then we have

$$0 = t_0 < t_1 < t_2 < \ldots < t_I = T, \tag{4.1}$$

where each sub-interval has length $\Delta t \equiv t_{i+1} - t_i = T/I$ for $i = 1, \ldots, I$. The integer I can be arbitrarily large, but a larger I leads to more accurate results at the expense of computational time. Next, we discretize the geometric Brownian motion:

$$S_{t_{i+1}} - S_{t_i} = rS_{t_i}\Delta t + \sigma S_{t_i}(W_{t_{i+1}} - W_{t_i}), \tag{4.2}$$

where $W_{t_{i+1}} - W_{t_i}$ is simulated from a normal distribution with mean 0 and variance Δt. With the assumption that $S_0 = s$, we can obtain the full sample-path and find the option's price for this path. For example, if we want to price the Russian option, we calculate the pay-off as in equation (3.1) and discount it by multiplying with e^{-rT}. If we simulate these paths enough times and calculate the option prices for these paths, the average simulated option price converges to the analytical price. The blue line in figure 1 shows a simulated sample-path using this method.

The MC simulation for the CIR and Heston process is very similar to the simulation

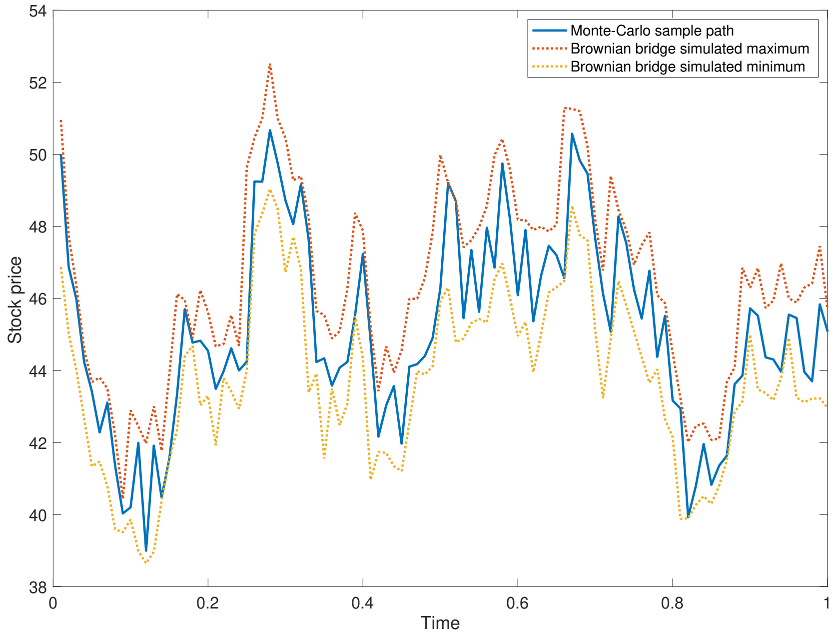


Figure 1: A simulated sample-path with corresponding Brownian bridge maxima and minima. The sample-path is constructed by discretizing a geometric Brownian motion with $T = 1$, $I = 100$, $r = 0.10$, $\sigma = 0.40$, and $S_0 = 50$. The minima (maxima) are constructed by drawing a minimum (maximum) of a Brownian bridge over each sub-interval.

for a geometric Brownian motion. For brevity, I only explain the methodology for the CIR process as the techniques between the two processes are the same. Suppose that the underlying price process follows equation (2.2) and equation (2.3), and we divide the interval from now to maturity date T as before, then we can discretize the model as follows:

$$r_{t_{i+1}} - r_{t_i} = \theta(\mu - r_{t_i})\Delta t + \xi\sqrt{r_{t_i}}(W^r_{t_{i+1}} - W^r_{t_i}) \tag{4.3}$$

$$S_{t_{i+1}} - S_{t_i} = r_{t_i}S_{t_i}\Delta t + \sigma S_{t_i}(W^S_{t_{i+1}} - W^S_{t_i}), \tag{4.4}$$

where $W^r_{t_{i+1}} - W^r_{t_i}$ and $W^S_{t_{i+1}} - W^S_{t_i}$ are simulated from a multivariate normal distribution with

$$\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Delta t & \rho\Delta t \\ \rho\Delta t & \Delta t \end{pmatrix}.$$

With the assumption that $S_0 = s$ and $r_0 = \zeta$, we can obtain the full sample-path and calculate the option price by calculating the option pay-off and discounting with $\exp\left\{-\Delta t \sum_{i=1}^{I} r_{t_i}\right\}$. Note that for the Heston model, we discount with e^{-rT} as the interest rate is constant in this model.

Finally, suppose that the underlying price process of n risky assets follows the multivariate geometric Brownian motion as in equation (2.6) and that the total interval is divided into the same sub-intervals as for the other processes, then we can discretize the model as follows:

$$\boldsymbol{S}_{t_{i+1}} - \boldsymbol{S}_{t_i} = r\boldsymbol{I}_n\boldsymbol{S}_{t_i}\Delta t + \boldsymbol{\Sigma}(\boldsymbol{W}_{t_{i+1}} - \boldsymbol{W}_{t_i}), \tag{4.5}$$

where $W_{t_{i+1}} - W_{t_i}$ is simulated from a multivariate normal distribution with mean $\boldsymbol{\mu} = \boldsymbol{0}_n$, and covariance matrix $\boldsymbol{V} = \Delta t\boldsymbol{I}_n$. With the assumption that $S_{i,0} = s_i$ for every $i = 1, \ldots, n$, we can obtain the full sample-paths of all assets and calculate the option price in the same manner as before.

4.2 Brownian bridge maximum method

Next, I look into the simulation with Brownian bridges. For options that depend on the stock reaching a specific price, the maximum stock price is crucial. The simulated path from the MC simulation, as described in the previous section, will always have a lower maximum than the true continuous-time path, according to Beaglehole et al. (1997). Suppose that the simulated path reaches its maximum at S_{t_i}, then we know that at time t_{i-1}, the price is lower. However, in the small time frame between t_{i-1} and t_i, the price could have reached a higher level than S_{t_i}. The lower maximum implies that MC simulation always underestimates the true option price. This problem could be solved by looking at the consecutive sub-intervals as a Brownian bridge that connects the previous price with the new simulated price. This simulation method is a significant improvement since we can draw a maximum of the Brownian bridge and find the maximum price during each sub-interval.

Suppose that the underlying asset follows a geometric Brownian motion as in equation (2.1), then the discretized process is the process in equation (4.2), and with the assumption that $S_0 = s$, we can obtain the full sample-path as usual. Using proposition 3.6 of Karatzas and Shreve (1988a), the continuous-time price process between t_i and t_{i+1} can be approximated by

$$S_t = S_{t_i} + \sigma S_{t_i} B_t, \tag{4.6}$$

where B_t is a standard Brownian bridge that connects 0 with $a = (S_{t_{i+1}} - S_{t_i})/\sigma S_{t_i}$, and $t_i \le t \le t_{i+1}$. The next step is to find the maximum price during the sub-interval; thus, we need to find the Brownian bridge's maximum between t_i and t_{i+1}. Karatzas and Shreve (1988b) show that the distribution of the maximum is

$$\mathbb{P}\left[\max_{t_i \le t \le t_{i+1}} B_t \le x \mid B_{t_{i+1}} = a\right] = 1 - e^{-2x(x-a)/\Delta t}. \tag{4.7}$$

We can now draw u from a standard uniform distribution and obtain a draw from the Brownian bridge maximum using the inverse transform method:

$$B_t^{\max} = \frac{1}{2}\left(a + \sqrt{a^2 - 2\Delta t \log(1-u)}\right). \tag{4.8}$$

Combining equation (4.6) and equation (4.8), the maximum stock price from t_i to t_{i+1} is equal to

$$S_{t_{i+1}}^{\max} = S_{t_i} + \sigma S_{t_i} B_t^{\max}. \tag{4.9}$$

If we do this for each i, we obtain a full sample-path of maxima, which we can use to price the options. Similarly, we can construct a full sample-path of minima using the reflection principle. The only difference is that we need to draw the maximum from a Brownian bridge that connects 0 with $-a$ and multiply this number with -1. This simulation algorithm is the same for the CIR as equation (4.8) and equation (4.9) do not depend on the interest rate. For the Heston process, I assume that the variance in a sub-interval is equal to the variance at the start of that sub-interval. That is, I assume that the transition of variance at time t_i to t_{i+1} is smooth. Therefore, the Brownian bridge is independent from the variance, and we only need to change σ to σ_{t_i} in equation (4.9). Figure 1 shows simulated Brownian bridge maxima and minima that correspond to a MC sample-path. For the multivariate geometric Brownian motion, equation (4.6) changes to:

$$\sum_{i=1}^{n} w_i S_{i,t} = \sum_{i=1}^{n} w_i S_{i,t_i} + \sigma_b \sum_{i=1}^{n} w_i S_{i,t_i} B_t, \tag{4.10}$$

where σ_b is the standard deviation of the basket, and B_t is a standard Brownian bridge that connects 0 with $a = (\sum_{i=1}^{n} w_i S_{i,t_{i+1}} - \sum_{i=1}^{n} w_i S_{i,t_i})/\sigma_b \sum_{i=1}^{n} w_i S_{i,t_i}$. The remainder of the simulation is the same.

5 Performance measures

The two simulation methods are compared by looking at the bias, the standard deviation per iteration, the mean squared error (MSE), and the computational time per error (MSE $\times$ time per simulation). For the Russian and down-and-out option with a geometric Brownian motion, I calculate the bias by comparing the simulated option price with the analytical price described in section 3. For the other processes, I simulate the option prices 10,000,000 times on a very fine grid ($I = 8T$, or once per trading hour) and take this as a converged estimate for the true price. I use the same analytical prices and converged estimates to calculate the mean squared error as follows:

$$\text{MSE} = \mathbb{E}_{C_{\text{option}}}\left[(\hat{C} - C_{\text{option}})^2\right] = \mathbb{V}_{C_{\text{option}}}(\hat{C}) + \mathbb{E}_{C_{\text{option}}}\left[\hat{C} - C_{\text{option}}\right]^2, \tag{5.1}$$

where C_{option} is the analytical or converged price of the specified option, and $\hat{C}$ is the simulated price. Since simulation with Brownian bridges requires extra steps, it is a slower method. Therefore, if simulation with Brownian bridges indeed reduces the bias, we need to decide if the bias reduction justifies the extra computational time. To decide this objectively, I multiply the MSE with the simulation time for each type of simulation.

6 Results

This section presents the results of the standard MC simulation and B2M2. I denote the time to compute a full simulation as CPU time, and I generate the results using a 2.4 GHz Quad-Core Intel Core i5 processor with 8 GB 2133 MHz LPDDR3 RAM on MacOS Catalina (10.15.4). The simulations are programmed in Matlab 2019a.

6.1 Russian option

The Russian option gives the holder of the contract the right to receive the maximum price that the stock has reached during the contract's lifetime minus the strike price. This section evaluates the efficiency of the different methods for this type of option. First, we discuss the geometric Brownian motion. We continue by inspecting the Cox-Ingersoll-Ross process and conclude with the Heston process. For each process, the stock price and strike price are equal and set at $S_0 = K = 50$.

6.1.1 Geometric Brownian motion

Table 1 shows the simulation bias for a Russian style option with $r = 0.10$, $\sigma = 0.25$, and two maturities: $T = 0.25$ (three months) and $T = 1$ (one year). The analytical price of the option with a maturity of three months is 5.771. The standard MC simulation consistently undervalues the option with prices around 5.20 and 5.37 for 45 and 90 sub-intervals, respectively. As the bias does not disappear when increasing the number of draws, we conclude that the MC simulation itself is biased. B2M2 produces prices that are much closer to the analytical price. The largest error is 0.171 for 1,000 simulated sample-paths and 90 sub-intervals, which is already much closer to the analytical price than the standard simulation technique. Furthermore, seven of the eight prices are not significantly different from the analytical price.

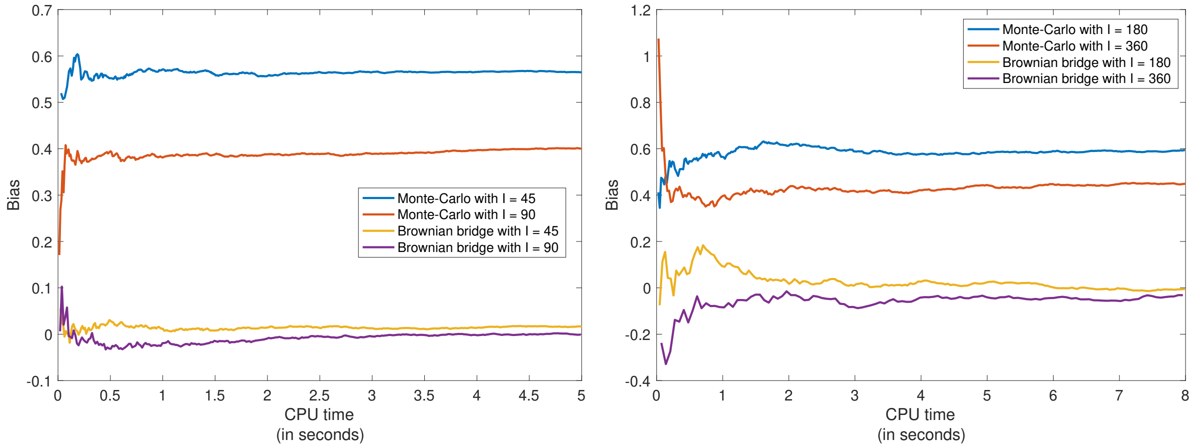
The analytical price of the Russian option with a maturity of one year is 12.891. We observe the same results as for the option with a maturity of three months. The standard MC simulation significantly undervalues the option with prices converging to 12.24 and 12.46 for 180 and 360 sub-intervals respectively. The standard MC simulation with 1,000 simulated draws is the only setting where the price is not significantly different from the analytical price. A high standard deviation causes the bias to be insignificant. When we increase the number of simulations, the bias changes relatively little, but the standard deviation becomes much smaller. As such, the bias is significantly different from zero for the larger number of simulations. B2M2 again produces prices that are much closer to the analytical price. None of the prices are significantly different from zero; thus, this method is unbiased with a geometric Brownian motion price process.

Table 1: Simulation bias of a Russian call option with a geometric Brownian motion price process. The time-to-maturity is denoted as T, the number of intervals is I, furthermore, the price process has the following parameters: $S_0 = 50$, $r = 0.10$, $\sigma = 0.25$, and $K = 50$. The bias is obtained by creating a specified number of sample-paths and calculating the average discounted pay-off of the option which is then subtracted from the analytical price. Bold faced numbers indicate that the simulated option price is not significantly different from the analytical price.

		Number of simulations				
T	I	1,000	10,000	100,000	1,000,000	Analytical price
Standard Monte-Carlo simulation						
0.25	45	0.593	0.623	0.585	0.565	5.771
		(0.143)	(0.046)	(0.014)	(0.005)	
0.25	90	0.374	0.365	0.391	0.396	5.771
		(0.144)	(0.046)	(0.014)	(0.005)	
1	180	**0.702**	0.739	0.641	0.625	12.891
		(0.368)	(0.114)	(0.036)	(0.011)	
1	360	**0.406**	0.397	0.503	0.453	12.891
		(0.375)	(0.114)	(0.036)	(0.011)	
Brownian bridge maximum simulation						
0.25	45	**0.070**	**0.015**	**-0.012**	0.011	5.771
		(0.145)	(0.046)	(0.015)	(0.005)	
0.25	90	**0.171**	**0.003**	**0.004**	**0.000**	5.771
		(0.142)	(0.047)	(0.015)	(0.005)	
1	180	**-0.078**	**0.133**	**0.027**	**-0.025**	12.891
		(0.346)	(0.113)	(0.036)	(0.012)	
1	360	**-0.317**	**0.027**	**-0.008**	**0.013**	12.891
		(0.339)	(0.115)	(0.036)	(0.011)	

Figure 2 plots the simulation time against the pricing bias produced by each method. The significant bias of the standard MC simulation is even more pronounced in this figure. As the simulation time increases, we observe a convergence of bias for all four simulation types, but the MC simulations do not converge to zero. For the option with a three-month maturity, the bias converges to 0.55 for $I = 45$ and 0.4 for $I = 90$. For the option with a one-year maturity, the bias converges to 0.6 for $I = 180$ and 0.4 for $I = 360$. As expected, the bias decreases somewhat by increasing the number of sub-intervals, but a significant amount of bias remains. For the three-month and one-year option, the Brownian bridge simulations both converge to zero; the convergence takes approximately one second for $T = 0.25$ and three seconds for $T = 1$.

Table 2 shows the summary statistics of the simulations with 1,000,000 replications. The massive bias of the MC method is again highlighted, with prices 40 cents below the analytical price. Although the standard deviation per iteration is slightly lower when



(a) $T = 0.25$ **(b)** $T = 1$

Figure 2: Simulation bias compared to simulation time for the Monte-Carlo and Brownian bridge simulation for a Russian call option with a geometric Brownian motion price process. The time-to-maturity is denoted as T, the number of intervals is I, furthermore, the price process has the following parameters: $S_0 = 50$, $r = 0.10$, $\sigma = 0.25$, and $K = 50$.

performing the standard MC simulation, the MSE per million iterations is much higher than for the Brownian bridge maximum simulation. This difference is huge for the option with a maturity of three months, where the MSE of the standard method is more than 50,000 times larger than B2M2. As noted in Beaglehole et al. (1997), the extra step in computing the maximum over each sub-interval increases the CPU time. The extra step results in an increase of approximately 4.5 seconds for $T = 0.25$ and 20 seconds for $T = 1$. To measure whether the added CPU time is worth reducing the bias, I look at the MSE multiplied by the CPU time. Using this statistic, we conclude that B2M2 is the preferred method as the statistic is more than 1,000 times smaller for $T = 0.25$ and more than 300 times smaller for $T = 1$. Thus, the significant reduction of bias is worth the added computational time.

Table 2: Summary statistics of a Russian call option with a geometric Brownian motion price process for 1,000,000 replications. The price process has the following parameters: $S_0 = 50$, $r = 0.10$, $\sigma = 0.25$, and $K = 50$. The standard deviation is per iteration, while the MSE, CPU, and MSE $\times$ CPU are per million iterations. The CPU time is presented in seconds.

Simulation method	Bias	Std. Dev.	MSE	CPU	MSE $\times$ CPU
Monte-Carlo ($T = 0.25$)	0.396	4.563	0.157	3.332	0.524
Brownian bridge ($T = 0.25$)	0.000	4.595	0.003×10^{-3}	8.802	0.247×10^{-3}
Monte-Carlo ($T = 1$)	0.453	11.401	0.186	9.017	1.678
Brownian bridge ($T = 1$)	0.013	11.481	0.165×10^{-3}	29.171	0.005

6.1.2 Cox-Ingersoll-Ross process

Next, we relax the assumption that the risk-free rate is constant. Instead, we assume that the interest rate is stochastic and follows a Cox-Ingersoll-Ross process. Table 3 shows the simulation biases of a Russian call option with $\sigma = 0.25$, $\theta = 0.70$ $\mu = 0.05$, $\xi = 0.15$, and $\rho = -0.20$ for an option with $T = 0.25$ (three months) and $T = 1$ (one year). Since the analytical price of a Russian option with a CIR process does not exist, we look at a converged price estimate. The converged price is based on 10,000,000 simulations on a very fine grid ($I = 8T$, once per trading hour), using the Brownian bridge approach since the previous section shows the accuracy of this method compared to standard MC. For the option with a maturity of three months, the converged price is 5.748. The standard MC simulation undervalues the option with prices around 5.20 and 5.35 for 45 and 90 sub-intervals, respectively. These results are very similar to the geometric Brownian motion results, and once again, we conclude that the standard MC simulation is biased. B2M2 produces prices that are much closer to the converged price. The largest error is 0.147 for 1,000 simulated sample-paths, which is already much closer to the analytical price than the standard simulation technique. None of the simulated prices are significantly different from the converged price, which means that this method is unbiased.

For the option with a maturity of one year, the converged price is 12.638. We observe the same results as for the option with a maturity of three months. The standard MC simulation significantly undervalues the option with prices converging to 12.0 and 12.2 for 180 and 360 sub-intervals, respectively. B2M2 produces much more accurate prices with the largest error equal to 0.114 for 1,000 simulated sample-paths. Again, none of the simulated prices is significantly different from the converged price. Hence, this method produces unbiased Russian call option prices.

Figure 3 plots the simulation time against the pricing bias produced by each method. The large bias of the standard MC simulation is again very pronounced in this figure. As the simulation time increases, we observe a convergence of bias for all four simulation types, but the MC simulations do not converge to zero. For the option with a three-month maturity, the bias converges to 0.55 for $I = 45$ and 0.4 for $I = 90$. For the option with a one-year maturity, the bias converges to 0.6 for $I = 180$ and 0.4 for $I = 360$. As expected, the bias decreases somewhat by increasing the number of sub-intervals, but a significant amount of bias remains. For both the three month and one year option, the Brownian bridge simulations both converge to zero; the convergence takes approximately four seconds for the three-month option and five second for the one-year option.

Table 4 shows the summary statistics of the simulations with 1,000,000 replications. The results are very similar to the geometric Brownian motion statistics: we observe a bias of approximately 40 cents for the MC simulations and a bias of less than a cent for B2M2. Furthermore, the standard deviation per iteration is slightly lower for MC, but

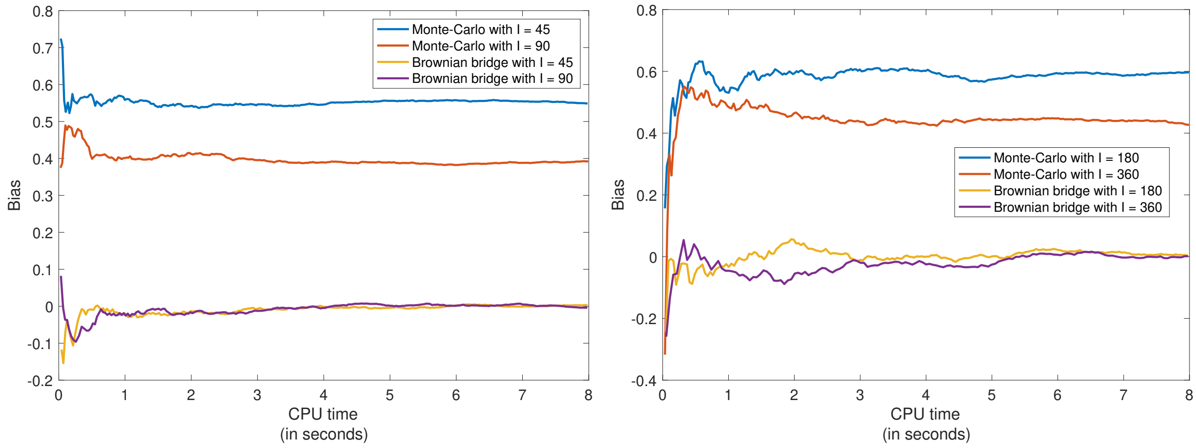
Table 3: Simulation bias of a Russian call option with a Cox-Ingersoll-Ross price process. The time-to-maturity is denoted as T, the number of intervals is I, furthermore, the price process has the following parameters: $S_0 = 50$, $r_0 = 0.10$, $\sigma = 0.25$, $\theta = 0.70$, $\mu = 0.05$, $\xi = 0.15$, $\rho = -0.20$, and $K = 50$. The bias is obtained by creating a specified number of sample-paths and calculating the average discounted pay-off of the option which is then subtracted from the converged price. Bold faced numbers indicate that the simulated option price is not significantly different from the converged price.

		Number of simulations				
T	I	1,000	10,000	100,000	1,000,000	Converged price
Standard Monte-Carlo simulation						
0.25	45	0.631	0.625	0.552	0.560	5.748
		(0.133)	(0.043)	(0.014)	(0.004)	
0.25	90	0.510	0.348	0.382	0.398	5.748
		(0.142)	(0.044)	(0.014)	(0.004)	
1	180	0.610	0.605	0.648	0.627	12.638
		(0.308)	(0.102)	(0.032)	(0.010)	
1	360	0.814	0.373	0.438	0.455	12.638
		(0.314)	(0.101)	(0.032)	(0.010)	
Brownian bridge maximum simulation						
0.25	45	**0.147**	**0.016**	**0.003**	**0.003**	5.748
		(0.140)	(0.045)	(0.014)	(0.004)	
0.25	90	**0.037**	**-0.033**	**0.001**	**0.005**	5.748
		(0.141)	(0.044)	(0.014)	(0.004)	
1	180	**0.066**	**0.045**	**-0.022**	**0.015**	12.638
		(0.334)	(0.103)	(0.032)	(0.010)	
1	360	**0.114**	**0.016**	**0.010**	**-0.018**	12.638
		(0.315)	(0.101)	(0.032)	(0.010)	

the MSE per million iterations is significantly lower for B2M2. The CPU time is increased for both simulations compared to the geometric Brownian motion due to the extra step of simulating the risk-free rate. Again, we observe a higher CPU time for B2M2, but the relative difference between the two methods is considerably smaller than in table 2. B2M2 is around 1.25 and 1.67 times slower for the option with a maturity of three months and one year, respectively. The MSE $\times$ CPU statistic shows that the increased CPU time of B2M2 is worth the decrease in bias as the statistic is drastically higher for the MC simulation.

6.1.3 Heston process

Next, we assume that the risk-free rate is constant again, but the volatility is no longer constant. We assume that the volatility is stochastic and follows a Heston process. Table 5 shows the simulation biases of a Russian call option with $r = 0.10$, $\theta = 2.00$ $\mu = 0.20$, $\xi = 0.40$, and $\rho = -0.30$ for an option with $T = 0.25$ (three months) and $T = 1$ (one



(a) $T = 0.25$ **(b)** $T = 1$

Figure 3: Simulation bias compared to simulation time for the Monte-Carlo and Brownian bridge simulation for a Russian call option with a Cox-Ingersoll-Ross price process. The time-to-maturity is denoted as T, the number of intervals is I, furthermore, the price process has the following parameters: $S_0 = 50$, $r_0 = 0.10$, $\sigma = 0.25$, $\theta = 0.70$, $\mu = 0.05$, $\xi = 0.15$, $\rho = -0.20$, and $K = 50$.

Table 4: Summary statistics of a Russian call option with a Cox-Ingersoll-Ross price process for 1,000,000 replications. The price process has the following parameters: $S_0 = 50$, $r_0 = 0.10$, $\sigma = 0.25$, $\theta = 0.70$, $\mu = 0.05$, $\xi = 0.15$, $\rho = -0.20$, and $K = 50$. The standard deviation is per iteration, while the MSE, CPU, and MSE $\times$ CPU are per million iterations. The CPU time is presented in seconds.

Simulation method	Bias	Std. Dev.	MSE	CPU	MSE $\times$ CPU
Monte-Carlo ($T = 0.25$)	0.398	4.420	0.159	19.563	3.108
Brownian bridge ($T = 0.25$)	0.005	4.450	0.392×10^{-4}	25.829	0.001
Monte-Carlo ($T = 1$)	0.455	10.137	0.207	30.768	6.381
Brownian bridge ($T = 1$)	-0.018	10.217	0.001	52.301	0.022

year). Since the analytical price of a Russian option with a Heston process does not exist, we look at a converged price estimate. The converged price is based on 10,000,000 simulations on a very fine grid ($I = 8T$, once per trading hour), using the Brownian bridge approach since the previous sections show the accuracy of this method compared to standard Monte-Carlo. For the option with a maturity of three months, the converged price is 10.532. The standard MC simulation undervalues the option with prices around 9.6 and 9.92 for 45 and 90 sub-intervals, respectively. These results are very similar to the results of the geometric Brownian motion, and once again, we conclude that the standard MC simulation is biased. B2M2 produces prices closer to the converged price; however, they are still biased, with prices converging to 9.98 and 10.17 for 180 and 360 sub-intervals, respectively.

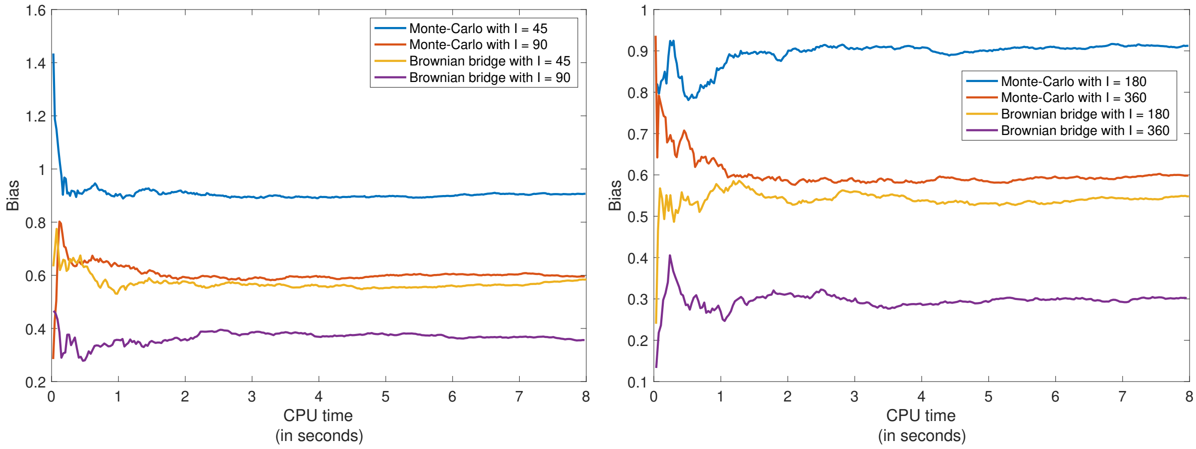
For the option with a maturity of one year, the converged price is 21.858. We observe the same results as for the option with a maturity of three months. The standard MC simulation and B2M2 significantly undervalue the option with prices converging to 21.19 for Monte-Carlo and 21.52 for the Brownian bridge with 360 sub-intervals. We conclude that for both simulation procedures, they produce unreliable option prices as they are significantly different from the converged price and are thus biased.

Table 5: Simulation bias of a Russian call option with a Heston price process. The time-to-maturity is denoted as T, the number of intervals is I, furthermore, the price process has the following parameters: $S_0 = 50$, $r = 0.10$, $\sigma_0 = 0.25$, $\theta = 2.00$, $\mu = 0.20$, $\xi = 0.40$, $\rho = -0.30$, and $K = 50$. The bias is obtained by creating a specified number of sample-paths and calculating the average discounted pay-off of the option which is then subtracted from the converged price. Bold faced numbers indicate that the simulated option price is not significantly different from the converged price.

		Number of simulations				
T	I	1,000	10,000	100,000	1,000,000	Converged price
Standard Monte-Carlo simulation						
0.25	45	1.164	1.039	0.927	0.937	10.532
		(0.282)	(0.090)	(0.029)	(0.009)	
0.25	90	0.726	0.455	0.600	0.605	10.532
		(0.284)	(0.092)	(0.029)	(0.009)	
1	180	**1.258**	0.817	0.922	0.993	21.858
		(0.728)	(0.228)	(0.073)	(0.023)	
1	360	**1.292**	0.768	0.668	0.671	21.858
		(0.728)	(0.230)	(0.072)	(0.023)	
Brownian bridge maximum simulation						
0.25	45	**0.571**	0.592	0.552	0.555	10.532
		(0.299)	(0.092)	(0.029)	(0.009)	
0.25	90	0.600	0.281	0.368	0.360	10.532
		(0.272)	(0.092)	(0.029)	(0.009)	
1	180	**0.323**	0.782	0.690	0.620	21.858
		(0.685)	(0.224)	(0.072)	(0.023)	
1	360	**0.649**	**0.296**	0.374	0.400	21.858
		(0.716)	(0.228)	(0.073)	(0.023)	

Figure 4 plots the simulation time against the pricing bias produced by each method. We observe that all four simulation types converge to a positive bias, which indicates that all methods undervalue the option. However, at every length of simulation time, we see that B2M2 has a lower bias than the standard MC method. For the three-month option, the standard MC technique's bias converges to 0.9 for $I = 45$ and 0.6 for $I = 90$, while B2M2 converges to 0.6 for $I = 45$ and 0.3 for $I = 90$. For the one-year option, the standard MC bias converges to 1.1 for $I = 180$ and 0.8 for $I = 360$, whereas B2M2 converges to 0.6 for $I = 180$ and 0.3 for $I = 360$. As expected, the bias decreases somewhat by increasing the number of sub-intervals, but a significant amount of bias remains.

Table 6 shows the summary statistics of the simulations with 1,000,000 replications. We observe a bias of approximately 60 cents for the MC simulations and a bias of around



(a) $T = 0.25$ **(b)** $T = 1$

Figure 4: Simulation bias compared to simulation time for the Monte-Carlo and Brownian bridge simulation for a Russian call option with a Heston price process. The time-to-maturity is denoted as T, the number of intervals is I, furthermore, the price process has the following parameters: $S_0 = 50$, $r = 0.10$, $\sigma_0 = 0.25$, $\theta = 2.00$, $\mu = 0.20$, $\xi = 0.40$, $\rho = -0.30$, and $K = 50$.

40 cents for B2M2. Furthermore, the standard deviation per iteration is slightly lower for the MC method, but the MSE per million iterations is significantly lower for B2M2. The CPU time is increased for both simulations compared to the geometric Brownian motion due to the extra step of simulating the volatility. Again, we observe a higher CPU time for B2M2, but the relative difference is smaller for $T = 0.25$ and larger for $T = 1$. B2M2 is around 1.25 and 1.67 times slower for the option with a maturity of three months and one year, respectively. The MSE $\times$ CPU statistic shows that the increased CPU time of B2M2 is worth the decrease in bias as the statistic is almost sixty percent lower than MC for $T = 0.25$ and forty percent smaller for $T = 1$.

Table 6: Summary statistics of a Russian call option with a Heston price process for 1,000,000 replications. The price process has the following parameters: $S_0 = 50$, $r = 0.10$, $\sigma_0 = 0.25$, $\theta = 2.00$, $\mu = 0.20$, $\xi = 0.4$, $\rho = -0.30$, and $K = 50$. The standard deviation is per iteration, while the MSE, CPU, and MSE $\times$ CPU are per million iterations. The CPU time is presented in seconds.

Simulation method	Bias	Std. Dev.	MSE	CPU	MSE $\times$ CPU
Monte-Carlo ($T = 0.25$)	0.605	9.121	0.366	18.949	6.938
Brownian bridge ($T = 0.25$)	0.360	9.131	0.130	24.284	3.147
Monte-Carlo ($T = 1$)	0.671	22.883	0.451	30.702	13.853
Brownian bridge ($T = 1$)	0.400	22.933	0.161	51.235	8.237

6.2 Knock-out option

The knock-out option gives the holder of the contract the right to buy a stock at maturity for the strike price, but if the price exceeds a certain threshold during the contract's lifetime, the option becomes worthless. We inspect the down-and-out option, which means that the option ceases to exist if the price goes below the threshold. This section evaluates the efficiency of the different methods for this type of option. First, we discuss the geometric Brownian motion. We continue by inspecting the Cox-Ingersoll-Ross process and conclude with the Heston process. For each process, the stock price and strike price are equal and set at $S_0 = K = 50$, and the barrier is equal to 45.

6.2.1 Geometric Brownian motion

Table 7 shows the simulation bias of a down-and-out call option for $r = 0.10$ and $\sigma = 0.25$ for an option with $T = 0.25$ (three months) and $T = 1$ (one year). The analytical price of the option with a maturity of three months is 4.048. The standard MC simulation consistently overvalues the option with prices around 4.47 and 4.35 for 45 and 90 sub-intervals, respectively. As the bias does not disappear when increasing the number of draws or increasing the number of sub-intervals, we conclude that the MC simulation is biased itself. B2M2 produces prices that are much closer to the analytical price. The largest error is 0.159 for 1,000 simulated sample-paths, which is already much closer to the analytical price than the standard simulation technique. For both $I = 45$ and $I = 90$, all simulations produce prices that are not significantly different from the analytical price.

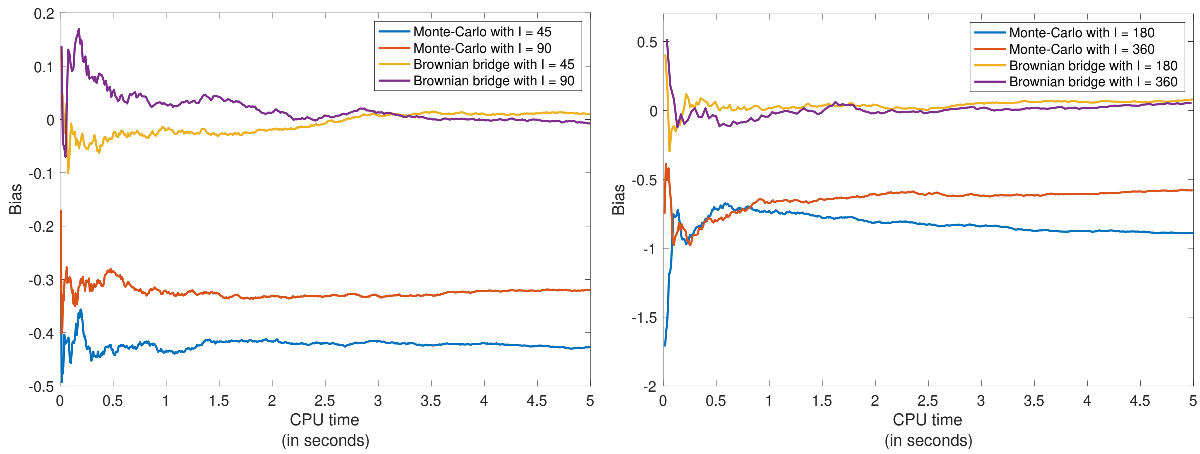
The analytical price of the down-and-out option with a maturity of one year is 5.509. We observe the same results as for the option with a maturity of three months. The standard MC simulation significantly overvalues the option with prices converging to 6.37 and 6.15 for 180 and 360 sub-intervals, respectively. B2M2 produces prices that quickly converge to the analytical price, with all biases not significantly different from zero. For $T = 0.25$ and $T = 1$, we conclude that the Brownian bridge maximum simulation produces unbiased prices for the down-and-out call option.

Figure 5 plots the simulation time against the pricing bias produced by each method. The large bias of the standard MC simulation is even more pronounced in this figure. As the simulation time increases, we observe a convergence of bias for all four simulation types, but the MC simulations do not converge to zero. For the option with a three-month maturity, the bias converges to -0.45 for $I = 45$ and -0.3 for $I = 90$. For the option with a one-year maturity, the bias converges to -0.9 for $I = 180$ and -0.6 for $I = 360$. As expected, the bias decreases somewhat by increasing the number of sub-intervals, but a significant amount of bias remains. For both the three-month and one-year option, B2M2 both converge to zero; the convergence takes approximately three seconds for $T = 0.25$ and five seconds for $T = 1$.

Table 8 shows the summary statistics of the simulations with 1,000,000 replications. The significant bias of MC is again highlighted, with prices at least 30 cents above the analytical price. The standard deviation per iteration is slightly smaller for B2M2, something we did not observe for the Russian call option. The MSE per million iterations

Table 7: Simulation bias of a down-and-out call option with a geometric Brownian motion price process. The time-to-maturity is denoted as T, the number of intervals is I, furthermore, the price process has the following parameters: $S_0 = 50$, $r = 0.10$, $\sigma = 0.50$, $K = 50$, and $H = 45$. The bias is obtained by creating a specified number of sample-paths and calculating the average discounted pay-off of the option which is then subtracted from the analytical price. Bold faced numbers indicate that the simulated option price is not significantly different from the analytical price.

		Number of simulations				
T	I	1,000	10,000	100,000	1,000,000	Analytical price
Standard Monte-Carlo simulation						
0.25	45	**-0.370**	-0.396	-0.446	-0.419	4.048
		(0.272)	(0.088)	(0.028)	(0.009)	
0.25	90	**-0.384**	-0.215	-0.318	-0.316	4.048
		(0.274)	(0.089)	(0.028)	(0.009)	
1	180	**-0.593**	-0.862	-0.952	-0.863	5.509
		(0.587)	(0.201)	(0.065)	(0.020)	
1	360	**-1.180**	-0.514	-0.707	-0.616	5.509
		(0.633)	(0.202)	(0.063)	(0.020)	
Brownian bridge maximum simulation						
0.25	45	**-0.066**	**0.019**	**-0.009**	**0.011**	4.048
		(0.280)	(0.087)	(0.028)	(0.008)	
0.25	90	**0.159**	**-0.071**	**0.019**	**0.006**	4.048
		(0.309)	(0.087)	(0.028)	(0.009)	
1	180	**-0.026**	**0.047**	**-0.006**	**0.008**	5.509
		(0.584)	(0.200)	(0.061)	(0.019)	
1	360	**0.324**	**0.260**	**-0.050**	**-0.015**	5.509
		(0.557)	(0.189)	(0.061)	(0.019)	



(a) $T = 0.25$ **(b)** $T = 1$

Figure 5: Simulation bias compared to simulation time for the Monte-Carlo and Brownian bridge simulation for a down-and-out call option with a geometric Brownian motion price process. The time-to-maturity is denoted as T, the number of intervals is I, furthermore, the price process has the following parameters: $S_0 = 50$, $r = 0.10$, $\sigma = 0.50$, $K = 50$, and $H = 45$.

is also much smaller for B2M2. The extra step for the Brownian bridge results in an increased CPU time of approximately 5.4 seconds for $T = 0.25$ and 20 seconds for $T = 1$. As we combine the MSE and CPU time, we conclude that B2M2 is the preferred choice for both options as the decrease in bias outweighs the increase in CPU time.

Table 8: Summary statistics of a down-and-out call option with a geometric Brownian motion price process for 1,000,000 replications. The price process has the following parameters: $S_0 = 50$, $r = 0.10$, $\sigma = 0.50$, $K = 50$, and $H = 45$. The standard deviation is per iteration, while the MSE, CPU, and MSE $\times$ CPU are per million iterations. The CPU time is presented in seconds.

Simulation method	Bias	Std. Dev.	MSE	CPU	MSE $\times$ CPU
Monte-Carlo ($T = 0.25$)	-0.316	8.815	0.094	3.353	0.314
Brownian bridge ($T = 0.25$)	-0.006	8.683	0.150×10^{-3}	8.728	0.001
Monte-Carlo ($T = 1$)	-0.616	20.084	0.416	9.509	3.960
Brownian bridge ($T = 1$)	-0.015	19.246	0.001	29.868	0.017

6.2.2 Cox-Ingersoll-Ross process

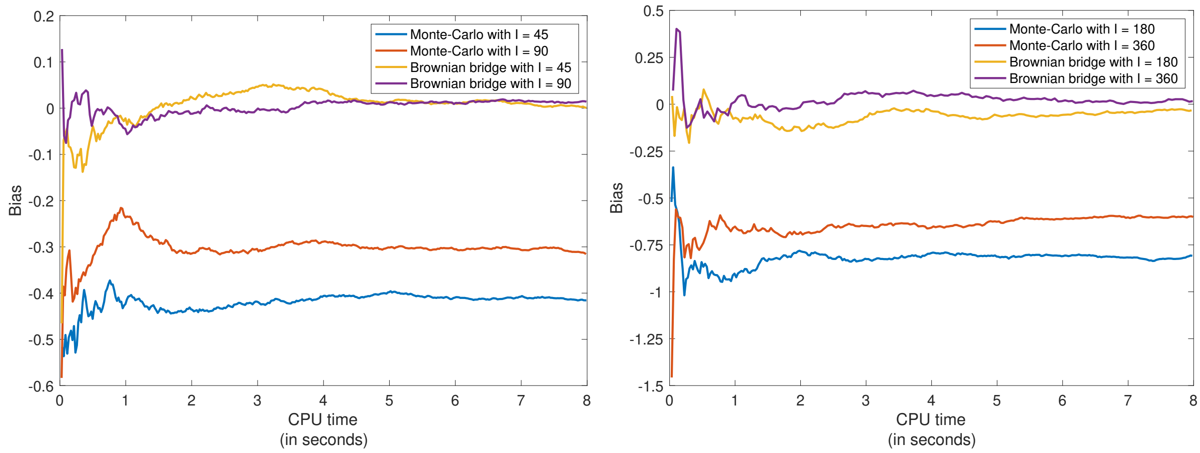
Next, we relax the assumption that the risk-free rate is constant. Instead, we assume that the interest rate is stochastic and follows a Cox-Ingersoll-Ross process. Table 9 shows the simulation bias of a down-and-out call option with $\sigma = 0.50$, $\theta = 0.70$ $\mu = 0.05$, $\xi = 0.15$, and $\rho = -0.20$ for an option with $T = 0.25$ (three months) and $T = 1$ (one year). Since the analytical price of a down-and-out option with a CIR process does not exist, we look at a converged price estimate. The converged price is based on 10,000,000 simulations on a very fine grid ($I = 8T$, once per trading hour), using the Brownian bridge approach since the previous section shows the accuracy of this method compared to standard Monte-Carlo. The converged price of the option with a maturity of three months is 4.030. The standard MC simulation consistently overvalues the option with prices around 4.45 and 4.35 for 45 and 90 sub-intervals, respectively. As the bias does not disappear when increasing the number of draws or number of sub-intervals, we conclude that the MC simulation is biased itself. B2M2 produces prices that are much closer to the analytical price. The largest error is 0.187 for 1,000 simulated sample-paths and $I = 90$, which is already much closer to the converged price than the standard simulation technique. Moreover, the biases produced by B2M2 are all not significantly different from zero, while the MC simulation only has six biases that are significantly different from zero.

The converged price of the down-and-out option with a maturity of one year is 5.374. We observe the same results as for the option with a maturity of three months. The standard MC simulation significantly overvalues the option with prices converging to 6.19 and 5.97 for 180 and 360 sub-intervals, respectively. Again, the bias does not disappear when increasing the number of simulations or sub-intervals, which means that the simulation procedure is biased. B2M2 produces prices that converge to the analytical price quickly with a bias of -0.003 ($I = 180$) and -0.005 ($I = 360$) for the simulation with 1,000,000 sample-paths. Thus, B2M2 produces unbiased down-and-out call option prices for both $T = 0.25$ and $T = 1$.

Table 9: Simulation bias of a down-and-out call option with a Cox-Ingersoll-Ross price process. The time-to-maturity is denoted as T, the number of intervals is I, furthermore, the price process has the following parameters: $S_0 = 50$, $r_0 = 0.10$, $\sigma = 0.50$, $\theta = 0.70$, $\mu = 0.05$, $\xi = 0.15$, $\rho = -0.20$, $K = 50$, and $H = 45$. The bias is obtained by creating a specified number of sample-paths and calculating the average discounted pay-off of the option which is then subtracted from the analytical price. Bold faced numbers indicate that the simulated option price is not significantly different from the analytical price.

		Number of simulations				
T	I	1,000	10,000	100,000	1,000,000	Converged price
Standard Monte-Carlo simulation						
0.25	45	**-0.083**	-0.580	-0.461	-0.418	4.030
		(0.252)	(0.087)	(0.027)	(0.008)	
0.25	90	**-0.492**	-0.317	-0.311	-0.303	4.030
		(0.268)	(0.087)	(0.027)	(0.009)	
1	180	**-0.861**	-1.050	-0.781	-0.820	5.374
		(0.612)	(0.187)	(0.057)	(0.018)	
1	360	**-1.034**	-0.483	-0.556	-0.594	5.374
		(0.577)	(0.172)	(0.056)	(0.018)	
Brownian bridge maximum simulation						
0.25	45	**0.166**	**-0.031**	**0.025**	**0.003**	4.030
		(0.250)	(0.085)	(0.027)	(0.008)	
0.25	90	**0.187**	**-0.012**	**0.015**	**-0.003**	4.030
		(0.261)	(0.083)	(0.027)	(0.008)	
1	180	**-0.008**	**0.112**	**-0.150**	**-0.003**	5.374
		(0.521)	(0.172)	(0.055)	(0.017)	
1	360	**0.260**	**0.005**	**0.054**	**-0.005**	5.374
		(0.554)	(0.176)	(0.054)	(0.017)	

Figure 6 plots the simulation time against the pricing bias produced by each method. Similar to the geometric Brownian motion, the large bias of MC is highlighted. As the simulation time increases, we observe a convergence of bias for all four simulation types, but the MC simulations do not converge to zero. For the option with a three-month maturity, the bias converges to -0.4 for $I = 45$ and -0.3 for $I = 90$. For the option with a one-year maturity, the bias converges to -0.8 for $I = 180$ and -0.6 for $I = 360$. As expected, the bias decreases somewhat by increasing the number of sub-intervals, but a significant amount of bias remains. For both the three month and one year option, the Brownian bridge simulations both converge to zero; the convergence takes approximately three seconds for $T = 0.25$ and four seconds for $T = 1$. Since the Brownian bridge method's absolute bias is always lower than the MC method at any simulation length, we conclude that simulating with a Brownian bridge is the preferred method.



(a) $T = 0.25$ **(b)** $T = 1$

Figure 6: Simulation bias compared to simulation time for the Monte-Carlo and Brownian bridge simulation for a down-and-out call option with a Cox-Ingersoll-Ross price process. The time-to-maturity is denoted as T, the number of intervals is I, furthermore, the price process has the following parameters: $S_0 = 50$, $r_0 = 0.10$, $\sigma = 0.50$, $\theta = 0.70$, $\mu = 0.05$, $\xi = 0.15$, $\rho = -0.20$, $K = 50$, and $H = 45$.

Table 10 shows the summary statistics of the simulations with 1,000,000 replications. The results are very similar to the geometric Brownian motion statistics: the MC simulation produces a bias of roughly 30 and 59 cents for the three-month and one-year option, respectively. B2M2 reduces the bias significantly since the option with a time-to-maturity of three months has a bias of three-tenths of a cent, while the option with expiration in one year has a bias of five-tenths of a cent. The standard deviation per iteration is slightly smaller for the Brownian bridge, which, again, is something we did not observe for the Russian call option. The MSE per million iterations is as expected from the previous statistics much smaller for B2M2. The extra step of simulating the risk-free rate increases the CPU time for both methods; however, the relative difference is smaller. B2M2 is around 1.38 and 1.70 times slower for the option with a maturity of three months and one year, respectively. As we combine the MSE and CPU time, we conclude that for both options, B2M2 is the preferred choice as the decrease in bias outweighs the increase in CPU time.

6.2.3 Heston process

Next, we assume that the risk-free rate is constant again, but the volatility is no longer constant. We assume that the volatility is stochastic and follows a Heston process. Table 11 shows the prices of a down-and-out call option with $r = 0.10$, $\theta = 2.00$ $\mu = 0.20$, $\xi = 0.40$, and $\rho = -0.30$ for an option with $T = 0.25$ (three months) and $T = 1$ (one year). Since the analytical price of a down-and-out option with a Heston process does not exist, we look at a converged price estimate. The converged price is based on 10,000,000 simulations on a very fine grid ($I = 8T$, once per trading hour), using the Brownian

Table 10: Summary statistics of a down-and-out call option with a Cox-Ingersoll-Ross price process for 1,000,000 replications. The price process has the following parameters: $S_0 = 50$, $r_0 = 0.10$, $\sigma = 0.25$, $\theta = 0.7$, $\mu = 0.05$, $\xi = 0.15$, $\rho = -0.20$, $K = 50$, and $H = 45$. The standard deviation is per iteration, while the MSE, CPU, and MSE $\times$ CPU are per million iterations. The CPU time is presented in seconds.

Simulation method	Bias	Std. Dev.	MSE	CPU	MSE $\times$ CPU
Monte-Carlo ($T = 0.25$)	-0.303	8.583	0.101	19.482	1.964
Brownian bridge ($T = 0.25$)	-0.003	8.449	0.244×10^{-3}	27.042	0.007
Monte-Carlo ($T = 1$)	-0.594	17.925	0.372	27.590	10.264
Brownian bridge ($T = 1$)	-0.005	17.154	0.295×10^{-3}	47.092	0.014

bridge approach since the previous sections show the accuracy of this method compared to standard MC. For the option with a maturity of three months, the converged price is 4.054. The standard MC simulation overvalues the option with prices around 4.41 and 4.28 for 45 and 90 sub-intervals, respectively. Seven out of eight prices are significantly different from the converged price; hence we conclude that this method produces biased results. B2M2 produces prices that are closer to the converged price. However, they are still not accurate, with prices converging to 4.25 and 4.20 for 45 and 90 sub-intervals, respectively. In conclusion, both simulation types are biased for the Heston model.

For the option with a maturity of one year, the converged price is 5.641. We observe the same results as for the option with a maturity of three months. The standard MC simulation and B2M2 significantly overvalue the option with prices converging to 6.10 for MC and 5.90 for B2M2 with 360 sub-intervals. For both simulation procedures, we conclude that they produce unreliable option prices as they are biased.

Figure 7 plots the simulation time against the pricing bias produced by each method. We observe that all four simulation types converge to a negative bias, which indicates that all methods overvalue the option. However, at every length of simulation time, we see that B2M2 has a lower absolute bias than the standard MC method. For the option with a three-month maturity, the bias of the standard MC technique converges to -0.3 for $I = 45$ and -0.25 for $I = 90$, while B2M2 converges to -0.35 for $I = 45$ and -0.18 for $I = 90$. For the option with a one-year maturity, the bias of standard MC converges to -0.8 for $I = 180$ and -0.5 for $I = 360$, whereas B2M2 converges to -0.7 for $I = 180$ and -0.4 for $I = 360$. As expected, the bias decreases somewhat by increasing the number of sub-intervals, but a significant amount of bias remains.

Table 12 shows the summary statistics of the simulations with 1,000,000 replications. We observe a bias of approximately 23 cents for the three-month option and 45 cents for the one-year option when applying the standard MC technique. B2M2 produces a bias of roughly 15 cents for the three-month option and 26 cents for the one-year option. Furthermore, the standard deviation per iteration is slightly higher for MC. Therefore, the MSE per million iterations is significantly lower for the Brownian bridge. The CPU time

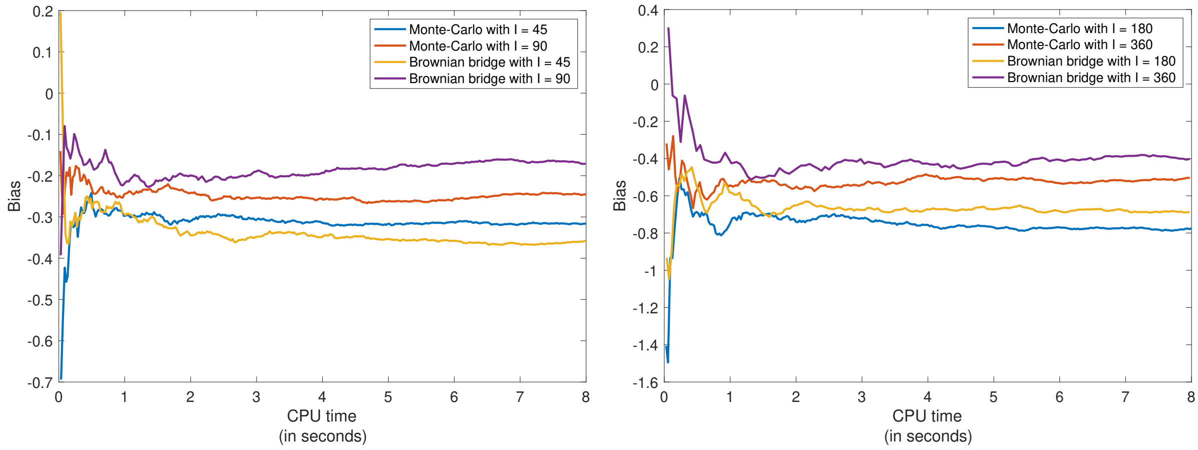
Table 11: Simulation bias of a down-and-out call option with a Heston price process. The time-to-maturity is denoted as T, the number of intervals is I, furthermore, the price process has the following parameters: $S_0 = 50$, $r = 0.10$, $\sigma_0 = 0.25$, $\theta = 2.00$, $\mu = 0.20$, $\xi = 0.40$, $\rho = -0.30$, $K = 50$, and $H = 45$. The bias is obtained by creating a specified number of sample-paths and calculating the average discounted pay-off of the option which is then subtracted from the analytical price. Bold faced numbers indicate that the simulated option price is not significantly different from the analytical price.

		Number of simulations				
T	I	1,000	10,000	100,000	1,000,000	Converged price
Standard Monte-Carlo simulation						
0.25	45	-0.820	-0.174	-0.313	-0.356	4.054
		(0.256)	(0.083)	(0.026)	(0.008)	
0.25	90	**-0.189**	-0.246	-0.257	-0.225	4.054
		(0.279)	(0.083)	(0.027)	(0.008)	
1	180	**0.054**	-0.705	-0.687	-0.691	5.641
		(0.610)	(0.186)	(0.058)	(0.018)	
1	360	**-0.101**	-0.499	-0.450	-0.454	5.641
		(0.514)	(0.180)	(0.057)	(0.018)	
Brownian bridge maximum simulation						
0.25	45	**0.023**	**-0.165**	-0.193	-0.199	4.054
		(0.268)	(0.085)	(0.026)	(0.008)	
0.25	90	**-0.254**	**-0.081**	-0.156	-0.147	4.054
		(0.275)	(0.082)	(0.026)	(0.008)	
1	180	**-0.380**	-0.976	-0.402	-0.408	5.641
		(0.557)	(0.189)	(0.057)	(0.018)	
1	360	**-1.051**	**-0.055**	-0.207	-0.260	5.641
		(0.581)	(0.168)	(0.055)	(0.018)	

is increased for both simulations compared to the geometric Brownian motion due to the extra step of simulating the volatility. Again, we observe a higher CPU time for B2M2, but the relative differences are smaller than in table 8. B2M2 is around 1.25 and 1.67 times slower for the option with a maturity of three months and one year, respectively. The MSE $\times$ CPU statistic shows that the increased CPU time of B2M2 is worth the decrease in bias as the statistic is at least twice as high for the Monte-Carlo simulation for both options.

6.3 Knock-in basket option

The knock-in basket option gives the holder of the contract the right to buy an agreed-upon set of stocks at maturity for the strike price, but only if the cumulative prices have exceeded a specified threshold during the contract's lifetime. In this case, we examine the up-and-in option, which means that the option expires worthless if the stocks' cumulative price did not exceed this threshold. This section evaluates the standard MC simulation's



(a) $T = 0.25$ **(b)** $T = 1$

Figure 7: Simulation bias compared to simulation time for the Monte-Carlo and Brownian bridge simulation for a down-and-out call option with a Heston price process. The time-to-maturity is denoted as T, the number of intervals is I, furthermore, the price process has the following parameters: $S_0 = 50$, $r = 0.10$, $\sigma_0 = 0.25$, $\theta = 2.00$, $\mu = 0.20$, $\xi = 0.40$, $\rho = -0.30$, $K = 50$, and $H = 45$.

Table 12: Summary statistics of a down-and-out call option with a Heston price process for 1,000,000 replications. The price process has the following parameters: $S_0 = 50$, $r = 0.10$, $\sigma_0 = 0.25$, $\theta = 2.00$, $\mu = 0.20$, $\xi = 0.40$, $\rho = -0.30$, $K = 50$, and $H = 45$. The standard deviation is per iteration, while the MSE, CPU, and MSE $\times$ CPU are per million iterations. The CPU time is presented in seconds.

Simulation method	Bias	Std. Dev.	MSE	CPU	MSE $\times$ CPU
Monte-Carlo ($T = 0.25$)	-0.225	8.313	0.051	20.026	1.014
Brownian bridge ($T = 0.25$)	-0.147	8.286	0.022	26.026	0.563
Monte-Carlo ($T = 1$)	-0.454	17.854	0.206	29.791	6.149
Brownian bridge ($T = 1$)	-0.260	17.693	0.068	49.674	3.383

efficiency against B2M2 for a basket of three stocks.

Table 13 shows the simulation bias of an up-and-in basket call option for an option with $T = 0.25$ (three months) and $T = 1$ (one year). This option has $n = 3$ underlying assets with the following parameters: $S_{0,1} = 50$, $S_{0,2} = 40$, $S_{0,3} = 60$, $r = 0.10$, $\sigma_1 = 0.25$, $\sigma_2 = 0.125$, $\sigma_3 = 0.4$, $\rho_{12} = 0.5$, $\rho_{13} = -0.3$, $\rho_{23} = -0.1$. The strike price is equal to 150 and the barrier is set at 175. The converged price of the option with a maturity of three months is 2.753. The standard MC simulation consistently undervalues the option with prices around 2.44 and 2.54 for 45 and 90 sub-intervals, respectively. As the bias does not disappear when increasing the number of draws or sub-intervals, we conclude that the MC simulation is biased. B2M2 produces prices that are much closer to the analytical price. The most substantial error is 0.148 for 1,000 simulated sample paths, which is already much closer to the analytical price than almost all standard simulation techniques. However, we see that a small amount of bias remains, which means that for

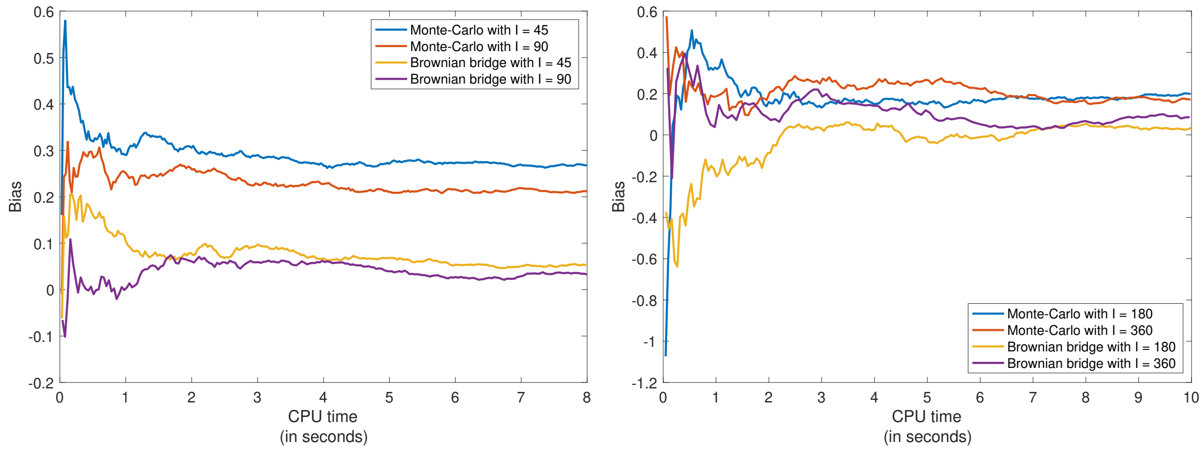
1,000,000 simulations, the option price is significantly different from the converged price.

The converged price of the up-and-in basket option with a maturity of one year is 16.583. We observe the same results as for the option with a maturity of three months. The standard MC simulation significantly undervalues the option with prices converging to 16.34 and 16.42. The standard MC simulation with 10,000 simulated draws and 360 sub-intervals has a much lower bias than the simulations with an increased number of draws. Moreover, it has much less bias than the Brownian bridge simulation with 1,000 draws. However, this result is likely a coincidence since B2M2 prices for 10,000, 100,000, and 1,000,000 sample-paths are again much closer to the converged price.

Table 13: Simulation bias of an up-and-in call option with a multivariate geometric Brownian motion price process. The time-to-maturity is denoted as T, the number of intervals is I, furthermore, the price process has the following parameters: $S_{0,1} = 50$, $S_{0,2} = 40$, $S_{0,3} = 60$, $r = 0.10$, $\sigma_1 = 0.25$, $\sigma_2 = 0.125$, $\sigma_3 = 0.4$, $\rho_{12} = 0.5$, $\rho_{13} = -0.3$, $\rho_{23} = -0.1$, $K = 150$, and $H = 175$. The bias is obtained by creating a specified number of sample-paths and calculating the average discounted pay-off of the option which is then subtracted from the analytical price. Bold faced numbers indicate that the simulated option price is not significantly different from the analytical price.

		Number of simulations				
T	I	1,000	10,000	100,000	1,000,000	Converged price
Standard Monte-Carlo simulation						
0.25	45	0.985	0.351	0.305	0.311	2.753
		(0.226)	(0.082)	(0.026)	(0.008)	
0.25	90	**0.370**	**0.121**	0.254	0.214	2.753
		(0.254)	(0.085)	(0.026)	(0.008)	
1	180	**-0.189**	**-0.076**	0.265	0.237	16.583
		(0.834)	(0.252)	(0.079)	(0.025)	
1	360	2.323	**-0.025**	**0.081**	0.167	16.583
		(0.719)	(0.250)	(0.079)	(0.025)	
Brownian bridge maximum simulation						
0.25	45	**-0.038**	**0.080**	0.081	0.068	2.753
		(0.270)	(0.086)	(0.027)	(0.009)	
0.25	90	**0.148**	**0.019**	**0.054**	0.047	2.753
		(0.271)	(0.086)	(0.027)	(0.009)	
1	180	**-0.517**	**0.028**	**0.118**	**0.034**	16.583
		(0.772)	(0.248)	(0.079)	(0.025)	
1	360	1.455	**0.233**	**-0.031**	**0.026**	16.583
		(0.697)	(0.245)	(0.079)	(0.025)	

Figure 8 plots the simulation time against the pricing bias produced by each method. The large bias of the standard MC simulation is again clearly visible in this figure. As the simulation time increases, we observe a convergence of bias for all four simulation types, but the MC simulations do not converge to zero. For the option with a three-month maturity, the bias converges to 0.25 for $I = 45$ and 0.2 for $I = 90$. For the option with a one-year maturity, the bias converges to 0.2 for $I = 180$ and $I = 360$. As expected, the bias decreases somewhat by increasing the number of sub-intervals, but a significant amount of bias remains. For the three-month option, the Brownian bridge's simulation bias converges to 0.05 for 45 and 90 sub-intervals. For the option with a maturity of one year, the bias goes to 0 for $I = 180$ and 0.05 for $I = 360$; the convergence takes approximately five seconds for $T = 0.25$ and eight seconds for $T = 1$.



(a) $T = 0.25$ **(b)** $T = 1$

Figure 8: Simulation bias compared to simulation time for the Monte-Carlo and Brownian bridge simulation for a up-and-in call option with a multivariate geometric Brownian motion price process. The time-to-maturity is denoted as T, the number of intervals is I, furthermore, the price process has the following parameters: $S_{0,1} = 50$, $S_{0,2} = 40$, $S_{0,3} = 60$, $r = 0.10$, $\sigma_1 = 0.25$, $\sigma_2 = 0.125$, $\sigma_3 = 0.4$, $\rho_{12} = 0.5$, $\rho_{13} = -0.3$, $\rho_{23} = -0.1$, $K = 150$, and $H = 175$.

Table 14 shows the summary statistics of the simulations with 1,000,000 replications. We observe significant bias reduction when using B2M2 as the bias is roughly five times smaller for the three month option and six times smaller for the one year option. Furthermore, the standard deviation per iteration is slightly lower for MC for the three-month option and vice versa for the one-year option. The MSE is much smaller for B2M2 with a value of 0.002 for $T = 0.25$ and 0.001 for $T = 1$. The CPU time is increased for the basket option as we simulate multiple stocks at the same time. Again, we observe a higher CPU time for the Brownian bridge, and it is around 1.30 and 1.40 times slower for the option with a maturity of three months and one year, respectively. The MSE $\times$ CPU statistic shows that the increased CPU time of B2M2 is worth the decrease in bias as the statistic is drastically higher for the MC simulation.

7 Conclusion

This paper proposed the Brownian bridge maximum method to simulate option prices based on extreme values instead of traditional Monte-Carlo simulations. The standard MC technique requires discretization of the underlying price process, after which we can

Table 14: Summary statistics of an up-and-in call option with a multivariate geometric Brownian motion price process for 1,000,000 replications. The price process has the following parameters: $S_{0,1} = 50$, $S_{0,2} = 40$, $S_{0,3} = 60$, $r = 0.10$, $\sigma_1 = 0.25$, $\sigma_2 = 0.125$, $\sigma_3 = 0.4$, $\rho_{12} = 0.5$, $\rho_{13} = -0.3$, $\rho_{23} = -0.1$, $K = 150$, and $H = 175$. The standard deviation is per iteration, while the MSE, CPU, and MSE $\times$ CPU are per million iterations. The CPU time is presented in seconds.

Simulation method	Bias	Std. Dev.	MSE	CPU	MSE $\times$ CPU
Monte-Carlo ($T = 0.25$)	0.214	8.425	0.046	28.770	1.319
Brownian bridge ($T = 0.25$)	0.047	8.583	0.002	37.536	0.085
Monte-Carlo ($T = 1$)	0.167	24.994	0.029	54.076	1.542
Brownian bridge ($T = 1$)	0.026	24.969	0.001	75.716	0.099

generate sample-paths and calculate the average option pay-off to obtain a price. Due to discretization, traditional MC simulation produces inaccurate option prices as information of extremal values in each subinterval is lost. This loss of information can be solved by simulating a Brownian bridge that connects the start- and endpoint of the sub-interval. Next, we draw a maximum from the Brownian bridge's distribution and use this to simulate the maximum or minimum price the underlying stock has reached in the subinterval. This paper examined four underlying price processes: the geometric Brownian motion, the Cox-Ingersoll-Ross process, the Heston process, and the multivariate geometric Brownian motion. For the first three processes, I obtained prices for a Russian call option and a down-and-out call option. The option I evaluated for the multivariate geometric Brownian motion is an up-and-in call option. A Russian call option gives the owner of the contract the right to receive the maximum price that the stock has reached minus the contract's strike price at maturity. A down-and-out call option gives the contract owner the right to buy the underlying stock at maturity for the strike price, but if the stock price goes under a specified threshold, the option becomes worthless. An up-and-in call option gives the owner of the contract a right to buy the underlying stock at maturity for the strike price, but only if the stock price has reached a specified threshold, otherwise the option pay-off is zero.

I evaluated the efficiency of B2M2 in three key statistics: bias, mean squared error, and CPU time. B2M2 produces prices close to the analytical and converged prices of all options and price processes, even for relatively little sub-intervals. This method produces unbiased option prices for the geometric Brownian motion, Cox-Ingersoll-Ross process, and the multivariate geometric Brownian motion for all options. For the Heston process, the bias is reduced compared to standard MC, but a significant amount of bias remains. For all options and all underlying price processes, the MC simulation produces options prices that are significantly different from the analytical and converged prices. This method overvalues options based on minimum prices, such as the down-and-out option, and overvalues options based on maximum prices, such as the Russian and up-and-in option. The standard deviation of the option price is almost identical for the two methods, MC has a slightly smaller standard error for the Russian option, while B2M2 has a slightly smaller error for the down-and-out option. The results are mixed for the up-and-in option. However, due to the similar standard deviations, B2M2 has a much smaller mean squared error for all underlying price processes and option types. B2M2 requires extra steps to generate sample-paths and is thus a slower method to obtain option prices. However, if we keep the CPU time equal for both methods, we observe that B2M2 always has a lower bias than Monte-Carlo. Only when we compare the two methods based on the number of iterations, we see that MC is approximately 28 percent faster. On the other hand, if we combine the MSE and the CPU time per one million iterations, we once again conclude that B2M2 produces better results than traditional MC simulation. In conclusion, B2M2 is the preferred method for obtaining exotic option prices as it reduces the bias and mean squared error, while still delivering fast results.

B2M2 reduces approximately fifty percent of the bias for an option with a Heston price process, but it is the only process for which this method cannot remove the bias. This method might be improved by also simulating the maximum of the stochastic volatility, which influences the maximum and minimum stock price during a sub-interval. Furthermore, other option types that are based on extreme prices, such as a swing option that pays the difference between the maximum and minimum stock price, have to be analyzed, as well as different and more complex underlying price processes, such as a Chen process with stochastic interest rates and volatility.

References

Andersen, L. (2008). Simple and efficient simulation of the Heston stochastic volatility model. *Journal of Computational Finance*, *11*(3), 1–42.

Avramidis, A. N., & L'Ecuyer, P. (2006). Efficient Monte Carlo and quasi-Monte Carlo option pricing under the variance gamma model. *Management Science*, *52*(12), 1930–1944.

Barraquand, J. (1995). Numerical valuation of high-dimensional multivariate European securities. *Management Science*, *41*(12), 1882–1891.

Beaglehole, D. R., Dybvig, P. H., & Zhou, G. (1997). Going to extremes: Correcting simulation bias in exotic option valuation. *Financial Analysts Journal*, *53*(1), 62–68.

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

Broadie, M., & Glasserman, P. (1995). A pruned and bootstrapped American option simulator. In *Winter Simulation Conference Proceedings, 1995.* (pp. 229–235).

Carrière, J. F. (1996). Valuation of the early-exercise price for options using simulations and non-parametric regression. *Insurance: Mathematics and Economics*, *19*(1), 19–30.

Clewlow, L., & Carverhill, A. (1994). On the simulation of contingent claims. *The Journal of Derivatives*, *2*(2), 66–74.

Conze, A., & Viswanathan. (1991). Path dependent options: The case of lookback options. *The Journal of Finance*, *46*(5), 1893–1907.

Cox, J. C., Ingersoll Jr, J. E., & Ross, S. A. (1985). An intertemporal general equilibrium model of asset prices. *Econometrica*, *53*(2), 363–384.

Gentle, D. (1993). Basket weaving. *Risk*, *6*(6), 51–52.

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The Review of Financial Studies*, *6*(2), 327–343.

Joy, C., Boyle, P. P., & Tan, K. S. (1996). Quasi-Monte Carlo methods in numerical finance. *Management Science*, *42*(6), 926–938.

Ju, N. (2002). Pricing Asian and basket options via Taylor expansion. *Journal of Computational Finance*, *5*(3), 79–103.

Karatzas, I., & Shreve, S. E. (1988a). Brownian motion and stochastic calculus. In (2nd ed., pp. 302–303). Springer. (Chapter 5)

Karatzas, I., & Shreve, S. E. (1988b). Brownian motion and stochastic calculus. In (pp. 265–267). Springer. (Chapter 4)

Kemna, A. G., & Vorst, A. C. (1990). A pricing method for options based on average asset values. *Journal of Banking & Finance*, *14*(1), 113–129.

Longstaff, F. A., & Schwartz, E. S. (2001). Valuing American options by simulation: A simple least-squares approach. *The Review of Financial Studies*, *14*(1), 113–147.

Metwally, S. A., & Atiya, A. F. (2002). Using Brownian bridge for fast simulation of jump-diffusion processes and barrier options. *The Journal of Derivatives*, *10*(1), 43–54.

Rubinstein, M., & Reiner, E. (1991). Breaking down the barriers. *Risk*, *4*(8), 28–35.

Tilley, J. A. (1993). Valuing American options in a path simulation model. *Transactions of the Society of Actuaries*, 55–67.

List of Tables

List of Figures

Appendix

Main code Russian option

```
%% Initialize
clear
ttm = 0.25;
noSteps = 90;
noReps = 1000000;
S0 = 50;
r = 0.10;
sigma = 0.25;
strike = S0;
d1 = (log(S0 / strike) + (r + 0.5 * sigma ^ 2) * ttm) / (sigma * sqrt(ttm));
d2 = d1 - sigma * sqrt(ttm);
truePrice = S0 * normcdf(d1) - exp(-r * ttm) * strike * normcdf(d2) + exp(-r * ttm) * sigma ^ 2 / (2 * r) * S0 * (-(S0 / strike) ^ (-2 * r / sigma ^ 2) * normcdf(d1 - 2 * r * sqrt(ttm) / sigma) + exp(r * ttm) * normcdf(d1));

%% Geometric Brownian motion
tic
[MCpriceGBM, MCerrorGBM] = russianGBM_MC(noReps, noSteps, ttm, S0, r, sigma, strike);
timeMC_GBM = toc;
biasMC_GBM = truePrice - MCpriceGBM;
mseMC_GBM = biasMC_GBM ^ 2 + MCerrorGBM ^ 2;
mseTimeMC_GBM = mseMC_GBM * timeMC_GBM;

tic
[BBpriceGBM, BBerrorGBM] = russianGBM_BB(noReps, noSteps, ttm, S0, r, sigma, strike);
timeBB_GBM = toc;
biasBB_GBM = truePrice - BBpriceGBM;
mseBB_GBM = biasBB_GBM ^ 2 + BBerrorGBM ^ 2;
mseTimeBB_GBM = mseBB_GBM * timeBB_GBM;

%% Cox-Ingersoll-Ross process
r0 = 0.10;
theta = 0.7;
mu = 0.05;
xi = 0.15;
corrSr = -0.20;
convSteps = noSteps * 8;

% Find converged price or use pre-calculated price
```

```
convPrice_CIR = 5.748;
%russianCIR_BB(10000000, convSteps, ttm, S0, r0, sigma, theta, mu, xi, corrSr, strike);

tic
[MCpriceCIR, MCerrorCIR] = russianCIR_MC(noReps, noSteps, ttm, S0, r0, sigma, theta, mu,
    xi, corrSr, strike);
timeMC_CIR = toc;
biasMC_CIR = convPrice_CIR - MCpriceCIR;
mseMC_CIR = biasMC_CIR ^ 2 + MCerrorCIR ^ 2;
mseTimeMC_CIR = mseMC_CIR * timeMC_CIR;

tic
[BBpriceCIR, BBerrorCIR] = russianCIR_BB(noReps, noSteps, ttm, S0, r0, sigma, theta, mu,
    xi, corrSr, strike);
timeBB_CIR = toc;
biasBB_CIR = convPrice_CIR - BBpriceCIR;
mseBB_CIR = biasBB_CIR ^ 2 + BBerrorCIR ^ 2;
mseTimeBB_CIR = mseBB_CIR * timeBB_CIR;

%% Heston process
v0 = 0.25;
theta = 2;
mu = 0.2;
xi = 0.4;
corrSv = -0.3;

% Find converged price or use pre-calculated
convPrice = 10.532;
%russianHES_BB(10000000, convSteps, ttm, S0, r, v0, theta, mu, xi, corrSv, strike);

tic
[MCpriceHES, MCerrorHES] = russianHES_MC(noReps, noSteps, ttm, S0, r, v0, theta, mu, xi,
    corrSv, strike);
timeMC_HES = toc;
biasMC_HES = convPrice - MCpriceHES;
mseMC_HES = biasMC_HES ^ 2 + MCerrorHES ^ 2;
mseTimeMC_HES = mseMC_HES * timeMC_HES;

tic
[BBpriceHES, BBerrorHES] = russianHES_BB(noReps, noSteps, ttm, S0, r, v0, theta, mu, xi,
    corrSv, strike);
timeBB_HES = toc;
biasBB_HES = convPrice - BBpriceHES;
mseBB_HES = biasBB_HES ^ 2 + BBerrorHES ^ 2;
mseTimeBB_HES = mseBB_HES * timeBB_HES;
```

Main code knock-out option

```
%% Initialize
clear
ttm = 1;
noSteps = 360;
noReps = 1000000;
S0 = 50;
r = 0.10;
sigma = 0.50;
```

```
strike = S0;
barrier = 45;
mu = log(1 + r) - 0.5 * sigma ^ 2;
lambda = 1 + (mu / sigma ^ 2);
x = log(S0 / strike) / (sigma * sqrt(ttm)) + lambda * sigma * sqrt(ttm);
y = log(barrier ^ 2 / (S0 * strike)) / (sigma * sqrt(ttm)) + lambda * sigma * sqrt(ttm);
p1 = S0 * normcdf(x) - strike * exp(-r * ttm) * normcdf(x - sigma * sqrt(ttm));
p3 = S0 * (barrier / S0) ^ (2 * lambda) * normcdf(y) - strike * exp(-r * ttm) * (barrier
    / S0) ^ (2 * lambda - 2) * normcdf(y - sigma * sqrt(ttm));
truePrice = p1 - p3;

%% Geometric Brownian motion
tic
[MCpriceGBM, MCerrorGBM] = knockoutGBM_MC(noReps, noSteps, ttm, S0, r, sigma, strike,
    barrier);
timeMC_GBM = toc;
biasMC_GBM = truePrice - MCpriceGBM;
mseMC_GBM = biasMC_GBM ^ 2 + MCerrorGBM ^ 2;
mseTimeMC_GBM = mseMC_GBM * timeMC_GBM * 1000;

tic
[BBpriceGBM, BBerrorGBM] = knockoutGBM_BB(noReps, noSteps, ttm, S0, r, sigma, strike,
    barrier);
timeBB_GBM = toc;
biasBB_GBM = truePrice - BBpriceGBM;
mseBB_GBM = biasBB_GBM ^ 2 + BBerrorGBM ^ 2;
mseTimeBB_GBM = mseBB_GBM * timeBB_GBM * 1000;

%% Cox-Ingersoll-Ross process
r0 = 0.10;
theta = 0.6;
mu = 0.05;
xi = 0.15;
corrSr = -0.20;
convSteps = 8 * noSteps;

% Find converged price or use pre-calculated
convPrice = 5.374;
%knockoutCIR_BB(10000000, convSteps, ttm, S0, r0, sigma, theta, mu, xi, corrSr, strike,
    barrier);

tic
[MCpriceCIR, MCerrorCIR] = knockoutCIR_MC(noReps, noSteps, ttm, S0, r0, sigma, theta,
    mu, xi, corrSr, strike, barrier);
timeMC_CIR = toc;
biasMC_CIR = convPrice - MCpriceCIR;
mseMC_CIR = biasMC_CIR ^ 2 + MCerrorCIR ^ 2;
mseTimeMC_CIR = mseMC_CIR * timeMC_HES;

tic
[BBpriceCIR, BBerrorCIR] = knockoutCIR_BB(noReps, noSteps, ttm, S0, r0, sigma, theta,
    mu, xi, corrSr, strike, barrier);
timeBB_CIR = toc;
biasBB_CIR = convPrice - BBpriceHES;
mseBB_CIR = biasBB_CIR ^ 2 + BBerrorHES ^ 2;
mseTimeBB_CIR = mseBB_CIR * timeBB_CIR;

%% Heston process
v0 = 0.25;
```

```
theta = 2;
mu = 0.2;
xi = 0.4;
corrSv = -0.3;

% Find converged price or use pre-calculated
convPrice = 5.641;
%knockoutHES_BB(10000000, convSteps, ttm, S0, r, v0, theta, mu, xi, corrSv, strike,
    barrier);

%% 1000 replications
tic
[MCpriceHES, MCerrorHES] = knockoutHES_MC(noReps, noSteps, ttm, S0, r, v0, theta, mu,
    xi, corrSv, strike, barrier);
timeMC_HES = toc;
biasMC_HES = convPrice - MCpriceHES;
mseMC_HES = biasMC_HES ^ 2 + MCerrorHES ^ 2;
mseTimeMC_HES = mseMC_HES * timeMC_HES;

tic
[BBpriceHES, BBerrorHES] = knockoutHES_BB(noReps, noSteps, ttm, S0, r, v0, theta, mu,
    xi, corrSv, strike, barrier);
timeBB_HES = toc;
biasBB_HES = convPrice - BBpriceHES;
mseBB_HES = biasBB_HES ^ 2 + BBerrorHES ^ 2;
mseTimeBB_HES = mseBB_HES * timeBB_HES;
```

Main code knock-in option

```
% Initialize
clear
ttm = 1;
noSteps = 360;
noReps = 1000000;
S0 = [50; 40; 60];
r = 0.10;
sigma = [0.25; 0.125; 0.4];
stockCorr = [1 0.5 -0.3; 0.5 1 -0.1; -0.3 -0.1 1];
strike = 150;
barrier = 175;
convSteps = noSteps * 8;

% Find converged price or use pre-calculated
convPrice = 16.583;
%knockinMGBM_BB(10000000, convSteps, ttm, S0, r, sigma, stockCorr, strike, barrier);

tic
[MCpriceMGBM, MCerrorMGBM] = knockinMGBM_MC(noReps, noSteps, ttm, S0, r, sigma,
    stockCorr, strike, barrier);
timeMC_MGBM = toc;
biasMC_MGBM = convPrice - MCpriceMGBM;
mseMC_MGBM = biasMC_MGBM ^ 2 + MCerrorMGBM ^ 2;
mseTimeMC_MGBM = mseMC_MGBM * timeMC_MGBM;

tic
```

```
[BBpriceMGBM, BBerrorMGBM] = knockinMGBM_BB(noReps, noSteps, ttm, S0, r, sigma,
    stockCorr, strike, barrier);
timeBB_MGBM = toc;
biasBB_MGBM = convPrice - BBpriceMGBM;
mseBB_MGBM = biasBB_MGBM ^ 2 + BBerrorMGBM ^ 2;
mseTimeBB_MGBM = mseBB_MGBM * timeBB_MGBM;
```