An Algorithm for Simulating Bermudan Option Prices on Simulated Asset Prices

Brian Norsk Huge and Niels Rom-Poulsen

Brian Norsk Huge
is chief analyst of quantitative research at Danske Bank in Copenhagen, Denmark.
brno@danskebank.dk

Niels Rom-Poulsen
is senior analyst at quantitative research at Danske Bank and Copenhagen Business School in Copenhagen, Denmark.
nrp@danskebank.dk

We present an algorithm that combines the Longstaff and Schwartz [2001] simulation algorithm and the bias reduction technique developed in Huge and Rom-Poulsen [2004] to simulate Bermudan option prices on securities so complex that their prices must be found by Monte Carlo simulations. This includes compound options and options on callable mortgage-backed bonds. Bias reduction is needed because using simulated prices of the underlying security to compute option pay-offs causes an upward bias in the option price. We prove consistency of the option price computed by the algorithm.

We propose a method for pricing Bermudan options on assets which are priced by Monte Carlo simulations. Generally, the approach can be used in situations where the price of the underlying is given by an unbiased Monte Carlo estimate. Examples include compound options like callable lookback options, where Monte Carlo simulations are needed to price the lookback option, and Bermudan options on Bermudan swaptions priced in a Libor Market Model. For the latter, the underlying Bermudan swaption can be priced with Monte Carlo simulations using a parameterized exercise strategy.[1] The proposed algorithm was originally developed for a similar problem, namely, pricing a Bermudan option on a callable mortgage-backed bond (MBS). The MBS can be viewed as a long position in a non-callable bond and a short position in a Bermudan swaption. Due to the non-optimal behavior of debtors, an exogenous estimated exercise strategy is used, and our specific parametrization yields a path-dependent pricing problem forcing us to use Monte Carlo simulations for valuing the MBS.

When prices of the underlying security are determined by Monte Carlo simulations, an upward bias in the Bermudan option price will emerge. The intuitive reason for this bias is that the Monte Carlo simulated price of the underlying security is only an estimate of the true asset price, and thus this estimate includes a stochastic error term with mean zero and strictly positive variance. This makes simulated prices of the underlying security more volatile than true prices and the result is an upward bias in the option price. The bias can be controlled by putting more computational effort into finding the price of the underlying security, but this poses a huge computational task, especially if there are many exercise dates. However, by using the bias reduction technique from Huge and Rom-Poulsen [2004], the computational task can be reduced considerably.

It is a complicated task to simulate prices of Bermudan style options. The reason is that a simulation routine works forward in the time dimension whereas the Bermudan option price must be found by dynamic programming, which is a backward running routine. Dynamic programming basically finds the exercise boundary endogenously, but if the exercise boundary is specified exogenously, the backward routine is unnecessary and only the forward running Monte Carlo simulations need to be performed to determine the value of the Bermudan option. This approach has been taken in Omberg [1987], who uses an exponential exercise boundary and in Bossaerts [1989], who uses polynomials to represent the exercise boundary. Another approach is to use dynamic programming to determine the exercise boundary endogenously using backward induction on the simulated state space. This was done in Tilley [1993], Barraquand and Martineau [1995], and Carriere [1996] and has recently been followed in the method of Longstaff and Schwartz [2001], Tsitsiklis and Van Roy [2001], in the random tree method of Broadie and Glasserman [1997], and in the stochastic mesh method of Broadie and Glasserman [2004]. A third approach is presented in Rogers [2002], in which the Bermudan option price is given as a minimization problem yielding an upper bound on the option value. A survey of American option pricing using Monte Carlo simulations can be found in Fu et al. [2001] and Chapter 8 in Glasserman [2004].

Most of the above-mentioned methods use dynamic programming to determine the exercise boundary endogenously. The procedure is as follows: first, the state space is simulated using an (in time) forward running routine. Second, the dynamic programming problem is solved using a backward running routine. The backward running routine is a recursive procedure that is initialized at option expiration by setting the option value equal to the option pay-off. It then recursively works back to time 0 by, at each exercise date, setting the option value equal to the maximum of the immediate exercise value and the present value of the expected future option value, also known as the continuation value. That is, at each exercise date a decision is made whether to exercise the option or to hold it for another period.

To use the dynamic programming approach, it is therefore necessary to be able to compute option payoffs at each exercise date. However, if the price of the underlying security must be computed by simulations, the problem of computing option pay-offs poses a huge computational burden. For example, consider a Bermudan option with L exercise dates. Using U paths to simulate the state space and N paths to simulate prices of the underlying security at each exercise date results in a total of $L \times U \times N$ path generations of the underlying security. However, by using the bias reduction technique in Huge and Rom-Poulsen [2004], the number of simulations used to compute prices of the underlying security, N, can be reduced considerably. In fact, it is shown that using one single simulation, $N = 1$, can be sufficient to find prices of the underlying security and still get acceptable option pay-offs. However, the number of state space simulations, U, must increase if N is to be low, and a good functional relationship between the price of the underlying security and the state variables has to be established. We utilize this to develop a simulation algorithm that prices Bermudan options on complex securities.

In the next section, we give a short description of the types of biases that are present in algorithms used for simulating Bermudan option prices. Also, it is illustrated that when the prices of the underlying security are simulated, an upward bias in the simulated option price emerges. The following section contains the model setup. The dynamic programming principle is presented for the original Bermudan pricing problem and for the Least-Squares Monte Carlo approximation. In addition, we introduce our proposed extension, constructed to handle cases in which the value of the underlying security must be found by Monte Carlo simulations. We present our numerical test cases, along with a short description of the Hull-White interest rate model. An explicit expression for a class of basis functions in the Hull-White model is also developed for a plain vanilla coupon bond. We follow with the results of the test cases, and our conclusions. Finally, the appendices contain proofs and the pseudo code for the developed algorithm respectively.

OPTION PRICE BIASES

Two sources of bias arise when simulating Bermudan option prices: a high bias resulting from using information about the future in making decisions whether to exercise or not, and a low bias coming from applying a suboptimal exercise strategy. The high bias is a result of applying the dynamic programming principle to simulated paths. At each exercise date, the dynamic programming recursion states that the state dependent option value is equal to the maximum of immediate exercise and the estimated continuation value. But the estimated continuation value is found using a backward recursion on the future simulated paths and future information thus influences the current exercise decision. To remove the high bias, it is necessary to separate the exercise decision from the continuation value. One way is to simulate a new continuation value in case of no exercise. An example of this can be found in Broadie and Glasserman [1997] resulting in an option price estimator that is biased low. In the original Longstaff and Schwartz [2001] algorithm, the two sources of bias are mixed. The state dependent option value is equal to the maximum of the intrinsic value and the discounted value of the one period ahead option value. This is a special case of simulating a new continuation value in which the existing simulated paths are used to evaluate the new continuation value. In practice, the Longstaff and Schwartz [2001] algorithm thus produces low-biased option price estimates (see Glasserman [2004] chapter 8). Our algorithm contains a third source of bias, which we know is a high bias, but since the original Longstaff and Schwartz [2001] algorithm mixes high and low biases, the bias in the resulting option price cannot be determined.

As mentioned above, a third bias is introduced when the underlying security has to be simulated. In the following, we give a short description of this option price bias problem; for a detailed discussion we refer to the section on Results. To illustrate the problem, we consider the two first test cases—Testcase1 and Testcase2 from the section on Results. In both test cases, the underlying security is a plain vanilla fixed coupon bond with a coupon of 4%, one yearly payment and maturity on June 1, 2034 (see Exhibit 1). We want to value a Bermudan put option on this coupon bond and for simplicity we only have two exercise dates. The Hull-White interest rate model is used and thus we simulate the spot rate forward in time. At each point in time, we know the entire yield curve since it is a known function of the spot rate, and we are able to calculate the true state dependent value of the coupon bond. This is done in Testcase1, in which the option payoffs at both exercise dates is computed using these true prices, resulting in no option price bias. In Testcase2, we also simulate the spot rate, but for each simulated interest rate path, at both exercise dates, we initiate a sub-simulation in order to simulate the state dependent value of the coupon bond. We then calculate option pay-offs using these simulated prices of the underlying security and since these simulated prices contain a stochastic error term, a bias in the simulated option value emerges. The results are shown in Exhibit 2 and Exhibit 3, which shows the relative error of the option price as a function of the number of simulated spot rate paths. For Testcase1, we see that the simulated option price contains no bias, but for Testcase 2, we see a large upward bias. It is also clear that the bias does not disappear by simply increasing the number of spot rate paths. If the option price bias is to be reduced, more computational power has to be used in the sub simulation at each exercise date to produce more precise estimates for the price of the underlying security.

THE MODEL

Define a time line $S = \{s_0, s_1, \ldots, s_M\}$ and let X_{s_0}, $X_{s_1}, \ldots, X_{s_M}$ be a $\mathbb{R}^d$-valued Markov chain with X_{s_0} fixed. X_{s_i}, $i = 1, \ldots, M$, represents the state variables which must be simulated and upon which the underlying security depends. S contains both exercise dates and dates influencing the future price of the underlying security.[2] It is necessary to include non-exercise dates in S because, for path dependent securities, variables that are important for the underlying security, may need to be updated between exercise dates for future option pay-off computations to be possible. We also define $\mathcal{T} = \{t_0, t_1, \ldots, t_L\} \subseteq S$, $t_0 = s_0$ and $t_L < s_M$, as the set of stopping times. t_L is the expiration date of the option. Let $h_{t_i}(X_{t_i})$, $t_i \in \mathcal{T}$ be the t_0 value of the pay-off at time t_i. Since the pay-off function only influences the option value on exercise dates, we put $h_{s_i}(X_{s_i}) = 0$ for all non-exercise dates $s_i \in S$. This formulation of including the discounting into the pay-off function is discussed in Glasserman [2004] Section 8.1. To lighten notation, we let the subscript i indicate time index t_i or s_i.

Bermudan Option Pricing and Optimal Stopping

In this section we present the original Bermudan option pricing problem in which no approximations are made. The Bermudan option price is given as the value of the option if optimal exercise behaviour is followed by the holder of the option (see Duffie [1996]). The optimal exercise strategy amounts to choosing the time of exercise (stopping time) that maximizes the value of the option today. Let $V_i^*(X_i)$ denote the value of the option at exercise date $t_i \in \mathcal{T}$ in state X_i. The superscript * indicates that this is the option price in the problem where no approximations are made. The value of the option is found by choosing, from the set of stopping times $\mathcal{T}_i = \{i, i+1, \ldots, L\}$, $\tau \in \mathcal{T}_i$, that maximizes $V_i^*(X_i)$. i.e.,

$$V_i^*(X_i) = \max_{\tau \in \mathcal{T}_i} \mathbb{E}_i^{\mathbb{Q}}[h_\tau(X_\tau)] \tag{1}$$

EXHIBIT 1
Test Cases

Id	Underlying security	Description
Testcase1	4% coupon bond, maturity:01/06/2034, yearly payment	Put option, 2 exercise dates, true price for the underlying security
Testcase2	4% coupon bond, maturity:01/06/2034, yearly payment	Put option, 2 exercise dates, simulated price for the underlying security, no bias reduction
Testcase3	4% coupon bond, maturity:01/06/2034, yearly payment	Put option, 2 exercise dates, simulated price for the underlying security + bias reduction
Testcase4	4% coupon bond, maturity:01/06/2034, yearly payment	Put option, 104 exercise dates, simulated price for the underlying secutity + bias reduction
Testcase5	Mortgage backed bond, 6% maturity:01/10/2038, quarterly payments	Put option, 104 exercise dates, simulated price for the underlying secutity + bias reduction

$V_i^*(X_i)$ is the value of the option given that the option was not exercised before date t_i. We are interested in computing $V_0^*(X_0)$, the value of the option today. The optimal stopping time is found using dynamic programming, and $V_0^*(X_0)$ is found recursively as follows:

$$V_L^*(X_L) = h_L(X_L) \tag{2}$$

$$V_i^*(X_i) = \max(h_i(X_i), \mathbb{E}_i^{\mathbb{Q}}[V_{i+1}^*(X_{i+1})]), t_i \in \mathcal{T}, i < L \tag{3}$$

At option expiration the value of the option is equal to its exercise value given by the function $h_L(X_L)$. On any exercise date i before expiration, the option holder has to decide whether to hold the option for another period, which has the value $\mathbb{E}_i^{\mathbb{Q}}[V_{i+1}^*(X_{i+1})]$, or to exercise immediately yielding a pay-off of $h_i(X_i)$. Working backwards iteratively, this procedure gives the solution to the pricing problem in Equation (1).

The Bermudan option valuation problem can also be formulated using continuation values. Define

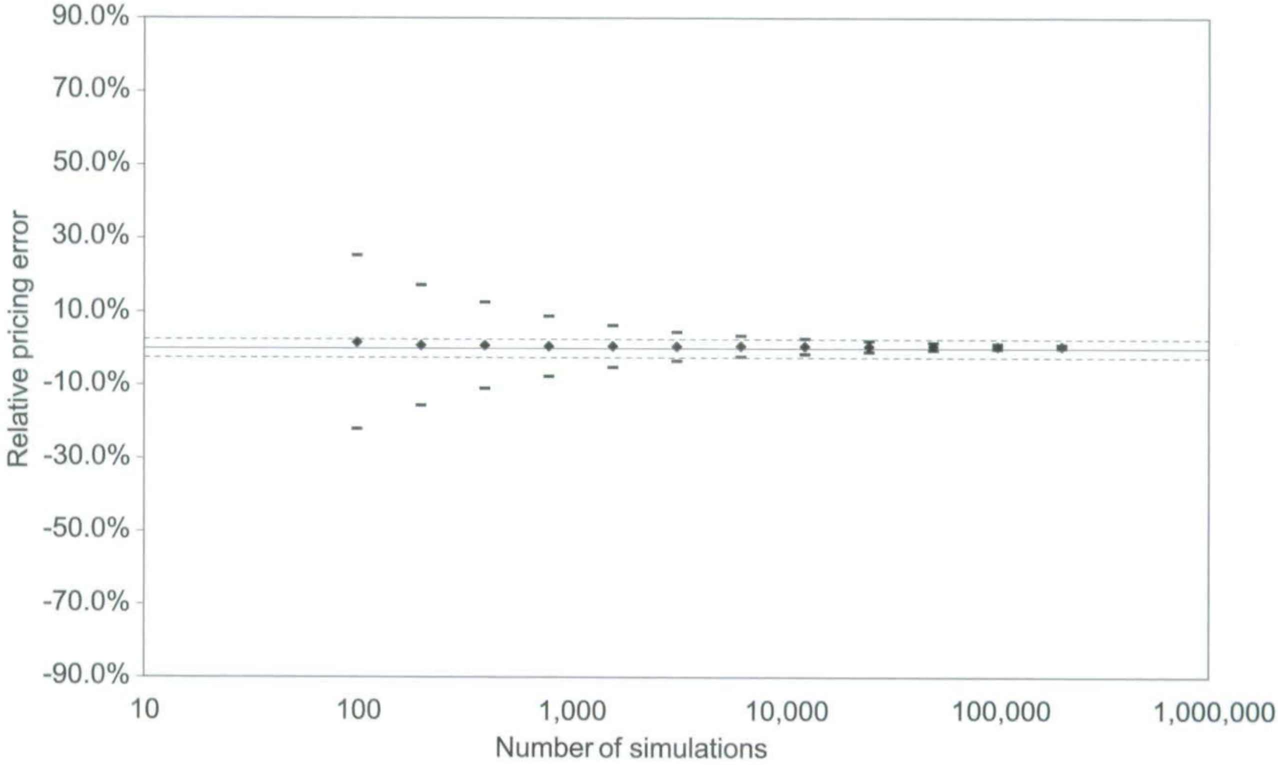
$$C_i^*(X_i) = \mathbb{E}_i^{\mathbb{Q}}[V_{i+1}^*(X_{i+1})] \tag{4}$$

and the recursion procedure in (2) and (3) now becomes

EXHIBIT 2

Testcase1—Convergence of Least Squares Monte Carlo

Diamonds represent the average relative pricing error. The upper and lower bar show ±2 standard deviation of the pricing error respectively. The dotted lines indicate ±2.5% deviations from the PDE computed option price. The sample size is 1000.



$$C_L^*(X_L) \equiv 0 \tag{5}$$

$$C_i^*(X_i) = \mathbb{E}_i^{\mathbb{Q}}[V_{i+1}^*(X_{i+1})],\, t_i \in \mathcal{T}, i < L \tag{6}$$

$$V_i^*(X_i) = \max(h_i(X_i), C_i^*(X_i)),\, t_i \in \mathcal{T} \tag{7}$$

Obviously, the continuation value is 0 at option expiration. At any date before option expiration, the continuation value is equal to the present value of the expected option value one period ahead. The option value is then given as the maximum of immediate exercise and the continuation value.

As can be seen from (2) and (3) or (5), (6), and (7), the dynamic programming routine works backward in the time dimension, and solving it using a forward working Monte Carlo simulation therefore seems impossible. However, in recent years, methods that combine Monte Carlo simulations and the dynamic programming problem have been proposed.

Least Squares Monte Carlo Simulations (LSMC)

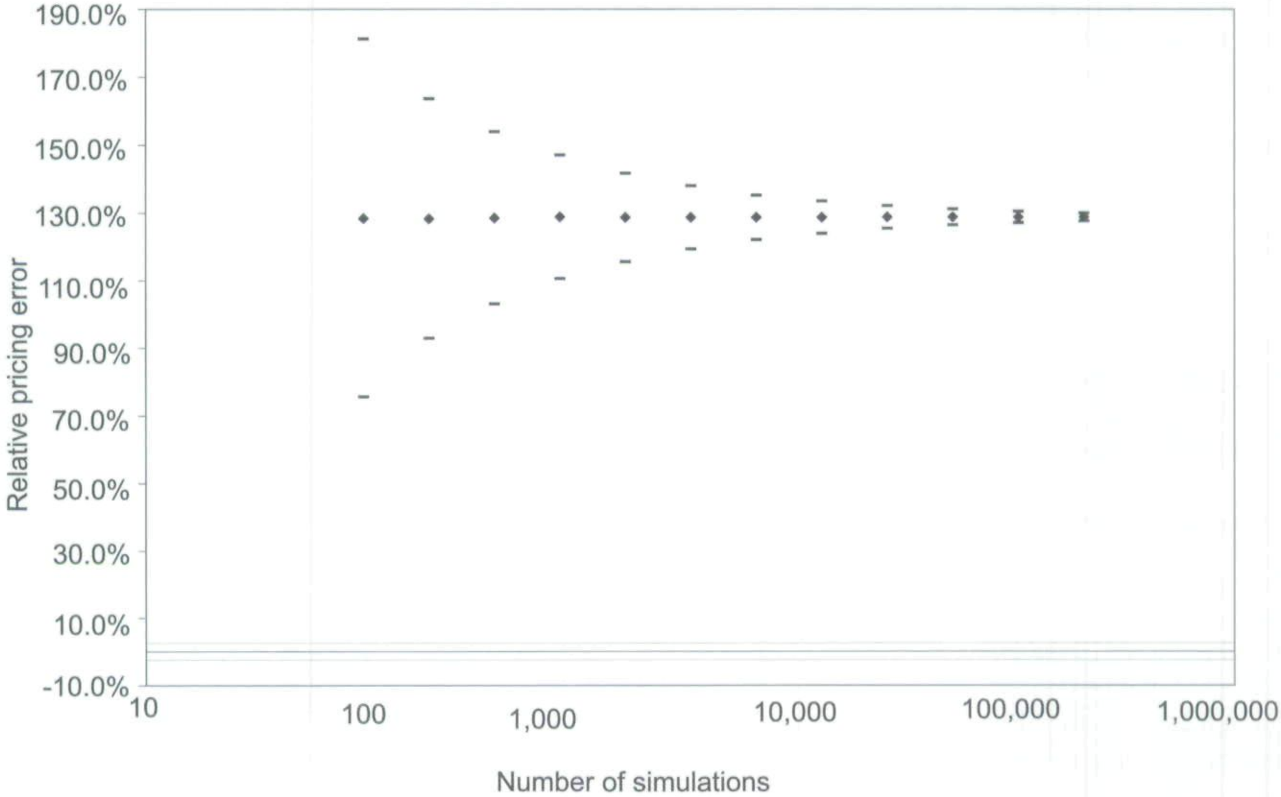
In order to distinguish approximated functions from the original problem presented in the prior section, we will use the notational convention that approximated functions do not have the * superscript as opposed to functions in the original problem.

Simulation routines for pricing Bermudan type options approximate the continuation value in (4). This is the case in both the Tsitsiklis and Van Roy [2001] and the Longstaff and Schwartz [2001] approaches. In both algorithms, the continuation value in (4) is approximated by a linear function of the form

EXHIBIT 3

Testcase2—Convergence of Least Squares Monte Carlo

Diamonds represent the average relative pricing error. The upper and lower bar show ±2 standard deviation of the pricing error respectively. The dotted lines indicate ±2.5% deviations from the PDE computed option price. The sample size is 1,000.



$$C_i^*(X_i) \approx \sum_{r=1}^{R} \zeta_{ir} \psi_{ir}(X_i) \equiv C_i(X_i) \tag{8}$$

where $\psi_i(X_i) = (\psi_{i0}(X_i), \ldots, \psi_{iR}(X_i))^{\mathsf{T}}$ is the value of the basis functions at time t_i in state X_i and $\zeta_i = (\zeta_{i0}, \ldots, \zeta_{iR})$ are the coefficients to the basis functions. $^{\mathsf{T}}$ denotes the matrix transpose operator. ζ_i is found by projecting the basis functions onto the continuation values, i.e.,

$$\zeta_i = \left(\mathbb{E}_i^{\mathbb{Q}}\left[\psi_i(X_i)\psi_i(X_i)^{\mathsf{T}}\right]\right)^{-1} \mathbb{E}_i^{\mathbb{Q}}\left[\psi_i(X_i)V_{i+1}(X_{i+1})\right] \tag{9}$$

The Tsitsiklis and Van Roy [2001] and the Longstaff and Schwartz [2001] algorithms differ in the way the value in (7) is approximated. In Tsitsiklis and Van Roy [2001], the value is approximated by

$$V_i(X_i) = \max(h_i(X_i), C_i(X_i)) \tag{10}$$

and in Longstaff and Schwartz [2001] it is approximated by

$$V_i(X_i) = \begin{cases} h_i(X_i) & h_i(X_i) \geq C_i(X_i) \\ V_{i+1}(X_{i+1}) & h_i(X_i) < C_i(X_i) \end{cases} \tag{11}$$

In the Tsitsiklis and Van Roy [2001] algorithm, the option pay-off is directly determined by the approximated continuation value. In the Longstaff and Schwartz [2001] algorithm, the approximated continuation value is only used to determine the exercise boundary and does not enter directly into the option pay-off.

Using the approximation in (8), the dynamic programming problem can now be written as

$$C_L(X_L) \equiv 0 \tag{12}$$

$$C_i(X_i) = \sum_{r=1}^{R} \zeta_{ir} \psi_{ir}(X_i), t_i \in \mathcal{T},\ i < L \tag{13}$$

and $V_i(X_i)$ equal to either (10) or (11).

In the rest of the article we will focus on the Longstaff and Schwartz [2001] algorithm only.

Extension of the LSMC Method

As dictated by the dynamic programming routine, we must be able to compute the immediate exercise value $h_i(X_i)$. When the price of the underlying security itself must be computed by simulations, an additional approximation is made. In that case, the option pay-off $h_i(X_i)$ is replaced by the approximation

$$\hat{h}_i(X_i) = f\left(\hat{P}_i(X_i)\right) \tag{14}$$

where $\hat{P}_i(X_i)$ is the price estimate of the underlying security in state X_i and $f(P_i(X_i))$ is the pay-off function, e.g., for a call/put, taking as input the state dependent price of the underlying security at exercise date t_i. $\hat{P}_i(X_i)$ could be a crude Monte Carlo estimate, but as shown in Huge and Rom-Poulsen [2004] this creates a bias in the option price. To reduce this bias much computational effort has to be allocated to find $\hat{P}(X_i)$. Huge and Rom-Poulsen [2004] propose an alternative price estimator for the price of the underlying security. This estimator uses crude Monte Carlo estimates to construct a new estimator that is less dependent of the computational effort put into generating the crude Monte Carlo estimates. The study in Huge and Rom-Poulsen [2004] only considers European options but in this section their arguments are extended so that the bias reduction method can be applied to price Bermudan options as well.

We will make the following assumptions:

Assumption 1. *Assume that the true price of the underlying security, P_i, $t_i \in \mathcal{T}$, is given as a finite linear combination of E basis functions, i.e.,*

$$P_i(X_i) = \sum_{j=1}^{E} \upsilon_{ij} \phi_{ij}(X_i) \tag{15}$$

Assumption 2. *For each exercise date $t_i \in \mathcal{T}$, along any two paths m, n, an unbiased crude Monte Carlo estimate, with independent error terms, for the price of the underlying security can be computed, i.e.,*

$$\hat{P}_i^{MC,n}(X_i) = P_i(X_i) + \epsilon_i^{MC,n} \tag{16}$$

$$\mathbb{E}_i^{\mathbb{Q}}\left[\epsilon_i^{MC,n}\right] = 0 \tag{17}$$

$$Var_i^{\mathbb{Q}}\left[\epsilon_i^{MC,n}\right] = \sigma_\epsilon^2 \tag{18}$$

$$Cov_i^{\mathbb{Q}}\left[\epsilon_i^{MC,m}, \epsilon_i^{MC,n}\right] = 0,\ \forall m, n \tag{19}$$

If Assumptions 1 and 2 are fulfilled, a better estimate for the price of the underlying security is given by

$$\hat{P}_i^{LS}(X_i) = \sum_{j=1}^{E} \hat{\upsilon}_{ij} \phi_{ij}(X_i) \tag{20}$$

where $\hat{\upsilon}_i = (\hat{\upsilon}_{i1} \ldots, \hat{\upsilon}_{iE})^\top$ is computed by

$$\hat{\upsilon}_i = \left(\mathbb{E}_i^{\mathbb{Q}}[\phi_i(X_i)\phi_i(X_i)^\top]\right)^{-1} \mathbb{E}_i^{\mathbb{Q}}\left[\phi_i(X_i)\hat{P}_i^{MC}(X_i)\right] \tag{21}$$

i.e., given Assumption 1 and Assumption 2, noise is removed by projecting the crude Monte Carlo estimates of the price of the underlying security onto a set of basis functions.

We denote functions that use the approximations (14) and (20) by superscript ^{LS}. The dynamic programming problem can now be stated as

$$C_L^{LS}(X_L) \equiv 0 \tag{22}$$

$$C_i^{LS}(X_i) = \sum_{r=1}^{R} \zeta_{ir}^{LS} \psi_{ir}(X_i), t_i \in \mathcal{T}, i < L \tag{23}$$

$$V_i^{LS}(X_i) = \begin{cases} \hat{h}_i^{LS}(X_i), & \hat{h}_i^{LS}(X_i) \geq C_i^{LS}(X_i), t_i \in \mathcal{T}, i < L \\ V_{i+1}^{LS}(X_{i+1}), & \hat{h}_i^{LS}(X_i) < C_i^{LS}(X_i), t_i \in \mathcal{T} \end{cases} \tag{24}$$

with $\hat{h}_i^{LS}(X_i) = f(\hat{P}_i^{LS}(X_i))$ and $P_i^{LS}(X_i)$ given by (20).

Solving with Simulations

In a practical implementation of the LSMC method from the section on LSMC simulations, the expectations in (9) are replaced by their sample counterparts thereby introducing one more approximation. Assume that U independent replications of the Markov chain have been sampled and denote them by $X_{1j}, \ldots, X_{Mj}, j = 1, \ldots, U$. Then the coefficients in (9) are estimated by

$$\hat{\zeta}_i = \left(\frac{1}{U}\sum_{j=1}^{U}\psi_i(X_{ij})\psi_i(X_{ij})^{\mathsf{T}}\right)^{-1} \left(\frac{1}{U}\sum_{j=1}^{U}\psi_i(X_{ij})V_{i+1}(X_{i+1,j})\right) \tag{25}$$

The dynamic programming problem can now be written for $j = 1, \ldots, U$

$$\hat{C}_{Lj}(X_{Lj}) \equiv 0 \tag{26}$$

$$\hat{C}_{ij}(X_{ij}) = \sum_{r=1}^{R}\hat{\zeta}_{ir}\psi_{ir}(X_{ij}), t_i \in \mathcal{T}, i < L \tag{27}$$

where (11) is given by

$$\hat{V}_{ij}(X_{ij}) = \begin{cases} \hat{h}_{ij}(X_{ij}) & \hat{h}_{ij}(X_{ij}) \geq \hat{C}_{ij}(X_{ij}) \\ \hat{V}_{i+1,j}(X_{i+1,j}) & \hat{h}_{ij}(X_{ij}) < \hat{C}_{ij}(X_{ij}) \end{cases} \tag{28}$$

Because X_0 is fixed, the price at time 0 is given by

$$\hat{V}_0(X_0) = \frac{1}{U}\sum_{j=1}^{U}\hat{V}_{1j}(X_{1j}) \tag{29}$$

Clemente, Lamberton, and Protter [2002] show that as the number of replications, U, increases and for $R < \infty$ fixed, the option price estimator in Equation (29) approaches the value of the approximative option price defined by relation (8). They also show that as $R \to \infty$ the stopping problem defined by the approximation (8) approaches the original optimal stopping problem.

In the extended LSMC approach, we further have to approximate the price of the underlying security. The expectations in (21) are also found using the simulated paths and the coefficients can thus be computed by

$$\hat{\upsilon}_i = \left(\frac{1}{U}\sum_{j=1}^{U}\phi_i(X_{ij})\phi_i(X_{ij})^{\mathsf{T}}\right)^{-1} \left(\frac{1}{U}\sum_{j=1}^{U}\phi_i(X_{ij})\hat{P}_{ij}^{MC}(X_{ij}; N)\right) \tag{30}$$

where N is the number of simulations used to generate the crude Monte Carlo prices.

The option price estimator is now defined as

$$\hat{V}_0^{LS} = \frac{1}{U}\sum_{j=1}^{U}\hat{V}_{1j}^{LS}(X_{1j}) \tag{31}$$

We will now show that using least squares price estimates, $\hat{P}_i^{LS}$, when computing option pay-offs yields an option price estimator that converges to the option price computed using least squares Monte Carlo as the number of paths U increases. The result does not depend on the number of simulations, N, used to generate the crude Monte Carlo estimates, $\hat{P}_i^{MC}(X_i)$, and is therefore valid for any such number.

Proposition 1 *Let U be the number of crude Monte Carlo estimates used to estimate the coefficients in (30) and let N be any positive integer. Assume that $\forall i$*

$$\psi(X_i)V_{i+1}^{LS}(X_{i+1}; U) \xrightarrow{d} \psi(X_i)V_{i+1}(X_{i+1}) \text{ as } U \to \infty \tag{32}$$

$$\sup_{U \geq 1}\mathbb{E}_i^{\mathbb{Q}}\left[|\psi(X_i)V_{i+1}^{LS}(X_{i+1}; U)|^{\delta}\right] < \infty \tag{33}$$

for some $\delta > 1$. Given Assumptions 1 and 2, the option price computed using (22), (23), and (24) converge to the least squares Monte Carlo computed option price from the section on LSMC Simulations as U approaches infinity, i.e.,

$$V_0^{LS}(X_0) \to V_0(X_0) \text{ as } U \to \infty$$

Proof: *See Appendix A.*

An implication of proposition 1 is that the estimator (31) will approach the estimator (29) as $U \to \infty$.

TEST CASES

In this section, we describe our computational test cases. Since our underlying interest rate model is the Hull-White model, we first present well known results for this model.

The Hull-White Model

In the Hull-White model, the $\mathbb{Q}$-dynamics of the spot rate is given by

$$dr_s = (\Theta(s) - \kappa r_s)ds + \sigma dW_s \tag{34}$$

where r_s is the spot rate, κ is the mean-reversion rate, σ is the spot rate volatility, and W_s is a Brownian motion. $\Theta(s)$ is found so that the initial term structure of market interest rates is matched and is given by

$$\Theta(s) = \frac{\partial f^M(0,s)}{\partial s} + \kappa f^M(0,s) + \frac{\sigma^2}{2\kappa}(1 - e^{-2\kappa s}) \tag{35}$$

where $f^M(0, s)$ is the observed market term structure of forward rates.

Applying Ito's lemma on $e^{\kappa s} r_s$ yields the following solution to the stochastic differential equation in (34)

$$\begin{aligned} r_S &= r_s e^{-\kappa(S-s)} + \gamma(S) - e^{-\kappa(S-s)}\gamma(s) + \sigma \int_s^S e^{-\kappa(S-u)} dW_u \\ \gamma(s) &= f^M(0, s) + \frac{\sigma^2}{2\kappa^2}(1 - e^{-\kappa s})^2 \end{aligned} \tag{36}$$

Since the stochastic integral $\sigma \int_s^S e^{-\kappa(S-u)} dW_u$ is normally distributed with mean zero and variance $\sigma^2 \int_s^S e^{-2\kappa(S-u)} du$ (see Björk [1998] Lemma 3.15), the conditional expected spot rate is given by

$$\begin{aligned} \mathbb{E}_s^{\mathbb{Q}}[r_S] &= r_s e^{-\kappa(S-s)} + \gamma(S) - e^{-\kappa(S-s)}\gamma(s) \\ \gamma(s) &= f^M(0, s) + \frac{\sigma^2}{2\kappa^2}(1 - e^{-\kappa s})^2 \end{aligned} \tag{37}$$

and has conditional variance

$$\mathrm{Var}_s^{\mathbb{Q}} = \frac{\sigma^2}{2\kappa}(1 - e^{-2\kappa(S-s)}) \tag{38}$$

Prices of zero coupon bonds are given by

$$P^{zcb}(s, S) = e^{\alpha(s,S) + \beta(s,S) r_s} \tag{39}$$

where $\alpha(s, S)$ and $\beta(s, S)$ are given by

$$\begin{aligned} \beta(s, S) &= \frac{1}{\kappa}(e^{-\kappa(S-s)} - 1) \\ \alpha(s, S) &= -f^M(0, s)\beta(s, S) + \ln\left(\frac{P^{zcb}(0,S)}{P^{zcb}(0,s)}\right) \\ &\quad + \frac{\sigma^2}{4\kappa}\beta^2(s, S)(e^{-2\kappa s} - 1) \end{aligned}$$

Basis functions in the Hull-White model Choosing a set of basis functions in (15) can be a difficult task. In this section we will choose a set of basis function that can be computed explicitly. To do this we let the underlying security be a plain vanilla coupon bond.

Start by writing the exponential function as the infinite sum

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \tag{40}$$

This leads to the following expression for a zero coupon bond in the Hull-White model

$$\begin{aligned} P^{zcb}(t, T) &= e^{\alpha(t,T)} e^{\beta(t,T) r_t} \\ &= e^{\alpha(t,T)} \sum_{n=0}^{\infty} \frac{(\beta(t,T) r_t)^n}{n!} \end{aligned} \tag{41}$$

The value at time $t \le t_1$ for a security with non-stochastic payments $b_1, \ldots, b_N$ at the dates $t_1, \ldots, t_N$ where $b_i = b$ for $i = 1, \ldots, N-1$ and $b_N = 1 + b$ can be written as

$$\begin{aligned} P(t) &= P^{zcb}(t, t_N) + b\sum_{i=1}^{N} P^{zcb}(t, t_i) \\ &= \sum_{i=1}^{N} b_i P^{zcb}(t, t_i) \\ &= \sum_{i=1}^{N} b_i e^{\alpha(t,t_i)} \sum_{n=0}^{\infty} \frac{(\beta(t, t_i) r_t)^n}{n!} \end{aligned}$$

$$
\begin{aligned}
&= \sum_{n=0}^{\infty} \sum_{i=1}^{N} b_i e^{\alpha(t,t_i)} \frac{(\beta(t,t_i) r_t)^n}{n!} \\
&= \sum_{n=0}^{\infty} \left(\sum_{i=1}^{N} b_i e^{\alpha(t,t_i)} \frac{\beta(t,t_i)^n}{n!} \right) r_t^n \\
&= \sum_{n=0}^{\infty} \upsilon_n r_t^n
\end{aligned} \tag{42}
$$

Line 1, 2 and 6 in Equation (42) all define a set of basis functions. Choosing the basis functions $P^{zcb}(t, t_N)$, $\Sigma_{i=1}^{N} P^{zcb}(t, t_i)$ will lead to coefficients 1 and b. Choosing the basis functions $P^{zcb}(t, t_1), \ldots, P^{zcb}(t, t_N)$ will lead to coefficients $b_1, \ldots, b_N$. Both sets fulfill Assumption 1 but these choices would require information about the pricing function $P(t)$. Thus, these choices do not seem realistic so instead we have chosen powers of the spot rate as basis functions. In this case Assumption 1 is usually violated. However, we can choose an arbitrary number of basis functions to get sufficiently close to the true price of the underlying security. This choice will also give us an indication of how serious it would be to violate Assumption 1.

By this choice of basis functions, the coefficients to the basis functions are given by

$$
\upsilon_n = \sum_{i=1}^{N} b_i e^{\alpha(t,t_i)} \frac{\beta(t,t_i)^n}{n!} \tag{43}
$$

Basis Functions in Practice

In Testcase1 to Testcase4, we use the first nine terms from Equation (42) as basis functions. In Testcase5, in which a callable mortgage backed security (MBS) is the underlying security, the basis functions are constructed using the following variables

$$
r, p, \omega_1, \omega_2, r^2, p^2, \omega_1^2, \omega_2^2, rp, r\omega_1, r\omega_2, p\omega_1, p\omega_2, \omega_1\omega_2
$$

where r is the spot rate, p is the MBS pool factor, ω_1 is MBS tranche weight 1 and ω_2 is MBS tranche weight 2[3]. The basis functions are now constructed from Chebyshev polynomials taking as input one of the 14 variables. Input variable x is assigned the basis function $T_1(x) = x$, x^2 is assigned the basis function $T_2(x) = 2x^2 - 1$, and xy is assigned $T_1(x)T_1(y) = xy$. Since we also include a constant, we end up with a total of 15 basis functions. For more details we refer to Huge and Rom-Poulsen [2004].

In Longstaff and Schwartz [2001], Laguerre polynomials are used as basis functions and in Glasserman and Yu [2004], the basis functions are build from Hermite polynomials. The basis functions are built from the simulated variables in the same manner we build basis functions from Chebyshev polynomials in the MBS case.

Test Cases

In Exhibit 1 we have listed the test cases that will be investigated in the section on Results below. Testcase1, Testcase2, and Testcase3 compute the Bermudan put option price on a coupon bond. To keep the computational burden to a minimum, we only have two exercise dates. Testcase1 covers the case where the price of the underlying security, at each exercise date, is given as a known function of the state variables (spot rate), meaning that the true state dependent price of the underlying can be easily computed from the simulated spot rate (state variable). This corresponds to the algorithm in Longstaff and Schwartz [2001] and is basically Algorithm 1 from Appendix B without bias reduction (without line 15 to 18). In Testcase2, we apply Algorithm 1 without bias reduction to illustrate that this creates a high bias in the Bermudan put option price. The bias arises because highly uncertain crude Monte Carlo simulated price estimates of the underlying security are used to compute the value of immediate exercise in the backward recursion of the dynamic programming algorithm. Testcase3 is Testcase2 plus bias reduction. Testcase3 should thus be compared with Testcase1 in order to evaluate the effectiveness of our proposed algorithm. In Testcase4, we compute the price of a 2-year Bermudan option with weekly exercise giving a total of 104 exercise dates. The underlying bond is the same as for Testcase1, Testcase2, and Testcase3. This test case is included to study how effective the algorithm is in approximating American option prices. For Testcase1 to Testcase4, we compare the option price computed with Algorithm 1 with a PDE[4] computed option price. Testcase5 is similar to Testcase4 except that the underlying security is a callable mortgage-backed security instead of a coupon bond. This test case is included to study the effectiveness of the algorithm when the underlying security is complex, and Testcase5 should thus be viewed as the prototype pricing problem that our algorithm is tailored to solve.

The parameters in the HW-model κ, σ, are chosen to be equal to 0.05 and 0.01 respectively and the model is calibrated to a flat yield curve of 5% by computing $\Theta(s)$ according to Equation (35). The trading date is April 6, 2004, and for Testcase1, Testcase2 and Testcase3 the exercise dates are April 6, 2005, and April 6, 2006, respectively. For Testcase4 and Testcase5, we have weekly exercise starting at April 13, 2004, and ending at April 4, 2006. The infinite sum in (42) is approximated using the first nine terms yielding the following approximation for the coupon bond.

$$P(t) \approx \sum_{n=0}^{8} \upsilon_n r_t^n \tag{44}$$

Note that using only a finite number of terms in Equation (44) is a violation of Assumption 1 and thus one more approximation is introduced. Since the underlying security is a coupon bond, the payments received each year at April 1 are given by

$$b_i = \begin{cases} 0.04 & t_i < 01/06/2034 \\ 1.04 & t_i = 01/06/2034 \end{cases} \tag{45}$$

RESULTS

In this section, we present the results of the test cases described in Exhibit 1. Exhibit 2 shows the convergence of the least squares Monte Carlo computed option price as the number of simulations increases and when true prices of the underlying security are used to compute option pay-offs. This case corresponds to the traditional least squares Monte Carlo algorithm as described in the section on LSMC Simulations. In the exhibit, ± 2 standard deviations of the option price estimate are also shown. The dashed lines indicate $\pm 2.5\%$ deviations from the PDE computed option price. As can be seen, the least squares Monte Carlo computed option price converges to the PDE computed option price as the number of simulations increases. Also, not surprisingly, as the number of simulations increases, the uncertainty of the price estimate decreases. It is also worth noting that all the price estimates are almost identical to the PDE computed option price, even for a small number of simulations. This would probably not be the case if we were to price an option on a more complex underlying than the option on a coupon bond we investigate in Exhibit 2.

In Exhibit 3, the convergence of the least squares Monte Carlo computed option price for Testcase2 is shown. In this test case, true prices of the underlying security have been replaced with crude Monte Carlo simulated prices when option pay-offs are computed. No bias reduction is performed. Recall that when option pay-offs are computed, Algorithm 1 only uses the current path to compute the state dependent price of the underlying security. Of course this price estimate will be highly uncertain, since only one single path has been used to simulate the price. This leads to a high bias in the option pay-off, and we would expect the resulting least squares Monte Carlo computed option price to be biased high as well. Also, the bias cannot be removed no matter how many simulations we put into computing the option price. The only way to reduce the bias is to lower the uncertainty of the crude Monte Carlo estimates of the underlying, which only can be achieved by increasing the work that is put into generating these prices. In many cases, this is very computationally burdensome and Algorithm 1 is constructed in a way that makes it impossible to use more than one path in the simulation of the underlying.

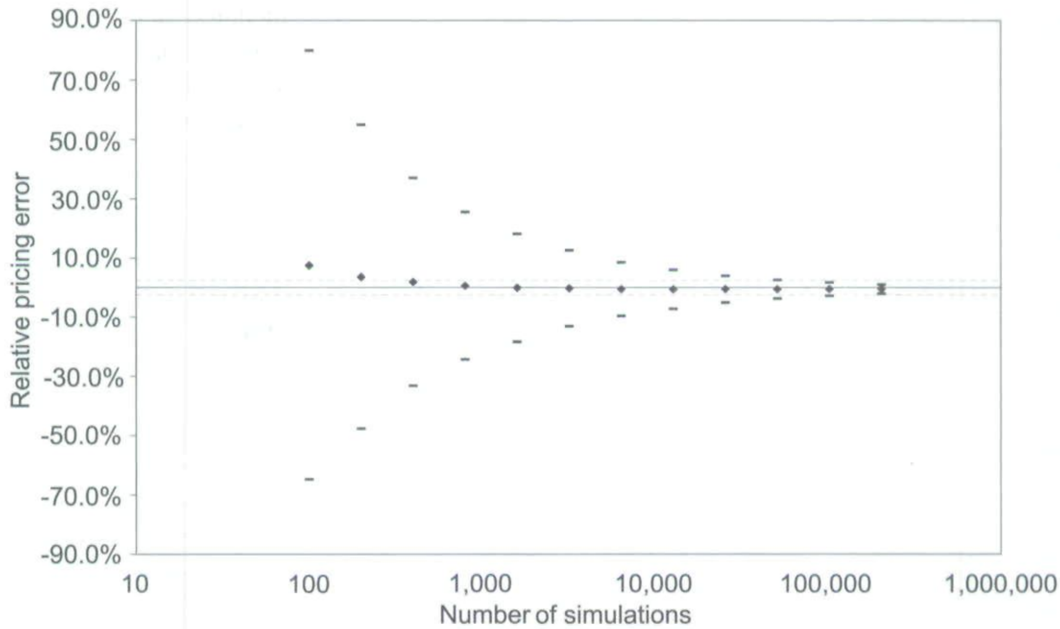
When we add bias reduction to Testcase2, we get Testcase3. The convergence result for this test case is shown in Exhibit 4. For a few simulations, the simulated option price is biased high. But as the number of simulations increases, the bias is reduced and the option price becomes almost equal to the PDE option price. Recall that we are only using nine basis functions in the polynomial approximating the underlying bond price, but this does not seem to be any problem. Another attractive feature of the algorithm is that both bias reduction and accuracy are improved by increasing the number of simulations. This is in contrast to Testcase2 in which only accuracy, but not bias reduction, is improved by increasing the number of simulations. Comparing with Testcase1, in which the true price of the underlying security was used to compute option pay-offs, the option prices computed in Testcase3 are very close to the ones computed in Testcase1. Only for 100, 200, and 400 simulations is there a small difference between the option prices in the two test cases. However, the uncertainty induced by using simulated prices of the underlying security when computing option pay-offs also turns up in higher uncertainty on the simulated option price and this higher uncertainty persists even for 204,800 simulations.

In Exhibit 5 the price of a Bermudan option with 104 exercise dates has been simulated. The effect of

EXHIBIT 4

Testcase3—Convergence of Least Squares Monte Carlo

Diamonds represent the average relative pricing error. The upper and lower bar show ±2 standard deviation of the pricing error respectively. The dotted lines indicate ±2.5% deviations from the PDE computed option price. The sample size is 1000.



increasing the number of exercise dates from 2 to 104 is higher bias and higher standard errors, especially for a low number of simulations. But as the number of simulations increases, both the bias and the standard errors are reduced and for approximately 12,800 simulations they are at the same low level as in Testcase3.

Finally, for Testcase5 we have simulated the price of a Bermudan option with 104 exercise dates on a MBS. The results are shown in Exhibit 6. We have not computed a PDE price for Testcase5, so the values shown are the simulated option prices. The conclusions are the same as for the previous test cases—both bias and standard errors are reduced when the number of simulations increases.

Estimated Price Relation

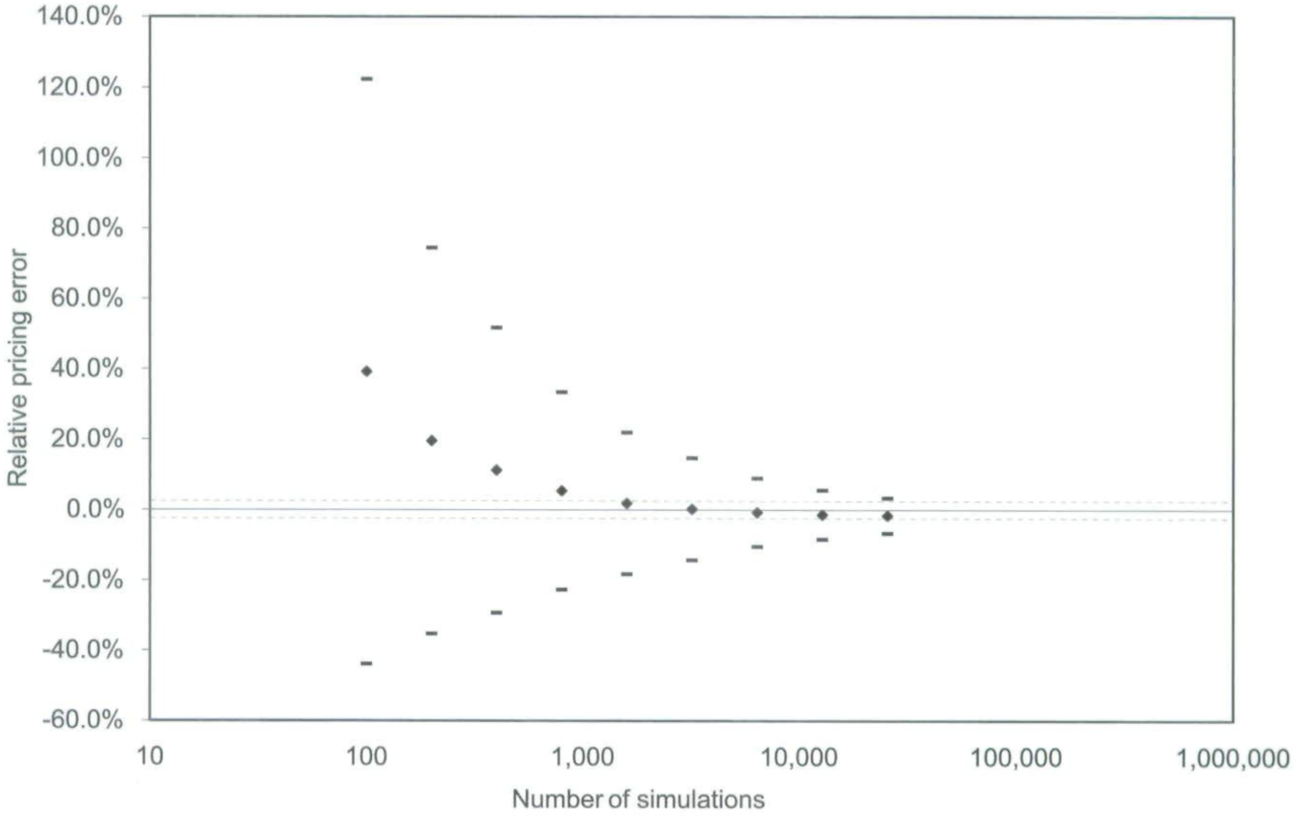
In this section we will investigate the estimated price relations of the underlying security at the exercise dates for Testcase3. These estimated relations are used to compute option pay-offs and it is therefore crucial that they are good approximations of the true relation between the price of the underlying security and the state variables.

In Exhibit 7 the true relation between the price of the coupon bond and the spot rate is shown at the first exercise date, April 6, 2005. The true price relation can easily be found in the Hull-White model since, in each future state, the entire yield curve is just an exponential affine function of the spot rate. Thus, the true price relation fulfills Equation 42. Exhibit 7 also shows three estimated relations between the price of the underlying security and the spot rate. These relations are found by projecting the crude Monte Carlo simulated price estimates of the underlying security onto powers of the spot rate. That is, they are approximations of the expression given by Equation (42). The approximations are found using 100, 6,400, and 204,800 simulations, respectively. As can be seen, the approximation becomes more accurate as the number of simulations increases. For only 100 simulations, the estimated price for high spot rates is completely wrong, whereas it is close to the true price relation for low and middle ranged spot rates. The reason for this divergence is that when we only use 100 simulations, there is little or no price information in the high spot rate region of the state space. The probability of getting a spot

Exhibit 5

Testcase4—Convergence of Least Squares Monte Carlo

Diamonds represent the average relative pricing error. The upper and lower bar show ±2 standard deviation of the pricing error respectively. The dotted lines indicate ±2.5% deviations from the PDE computed option price. The sample size is 1000.



rate, at exercise date 1, above 8% is only 0.11%, meaning that on average only 0.11 out of 100 simulated spot rates will be above 8%. Thus, the probability of getting a crude Monte Carlo estimate of the underlying security in this part of the state space is very low, making the estimated price relation very difficult to infer. The bias reduction method takes advantages of a diversification effect in the sense that if one has enough observations, the error term embedded in each observation can be filtered away. But since the elapsed time from today until the first exercise date is relatively short, the simulated spot rates are not very dispersed and will be concentrated around its conditional mean at exercise date 1. This can also be seen in Exhibit 7, namely in regions of the state space where there are many observations, the estimated price relations are very accurate. Looking at Exhibit 4, we see that the inaccurate estimated price relation of the underlying results in an inaccurate and biased option price. The reason the computed option prices are not that inaccurate in spite of the divergence in the estimated price relation of the underlying is that the weight (probability) which these pay-offs get is relatively low. Increasing the number of simulations to 6,400 and 204,800 removes the divergence of the estimated price relation, and with 204,800 simulations the estimated price relation is relatively close to the true price relation for all levels of the spot rate. This can also be seen in the option prices as a more accurate option price estimate (see Exhibit 4).

Now turn to exercise date 2. In Exhibit 8, the relation between the price of the underlying and the spot rate is shown for the same models as in Exhibit 7. For 6,400 and 204,800 simulations, the conclusions are the same as for exercise date 1—that the estimated price relations are good at approximating the true price relation

EXHIBIT 6

Testcase5—Convergence of Least Squares Monte Carlo

Diamonds represent the average option price. The upper and lower bar shows ±2 standard deviation of the pricing respectively. The sample size is 1,000.

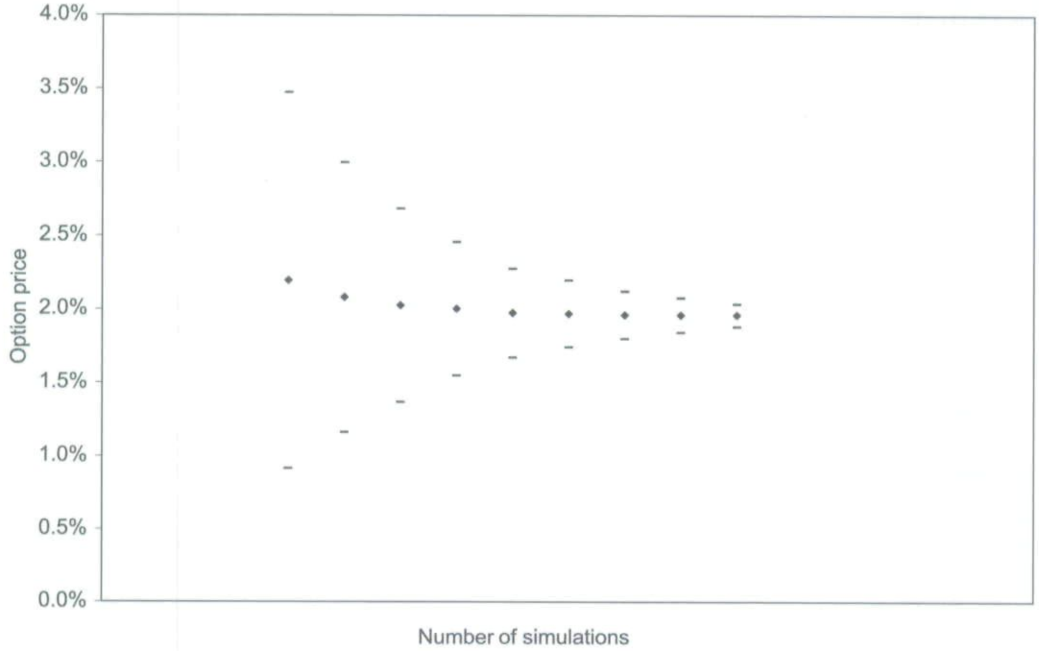
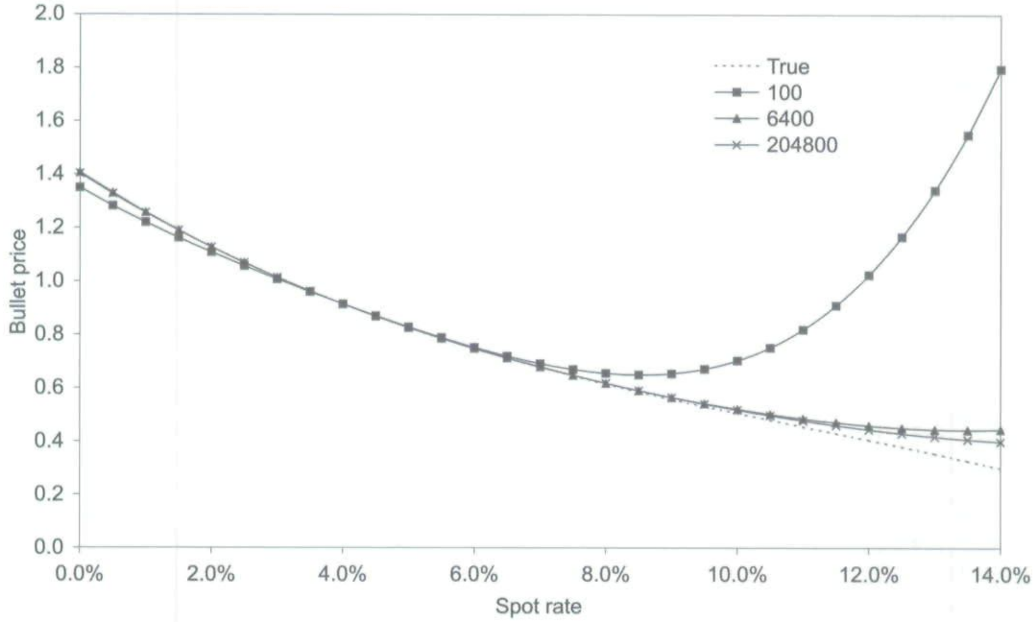


EXHIBIT 7

Testcase3 Model Price-Spot Rate Relation and Estimated Price-Spot Rate Relation of the Underlying Coupon Bond at Exercise Data 1. Initial Spot Rate is 5%.



over a large range of the state space. For 100 simulations, however, we have a much better approximation of the true price relation of the underlying. The reason is the longer time period between today and exercise date 2, which results in more dispersed spot rates over the state space than for exercise date 1. At exercise date 2, the probability of getting a spot rate above 8% is now 1.34%, more than 10 times higher than for exercise date 1.

Estimated Price Coefficients

In an earlier section, we derived an expression for the coefficients to the basis functions in the Hull-White model for a coupon bond. In this section, we investigate the first five estimated coefficients at exercise dates 1 and 2. The true coefficients are computed by Equation (43) and are summarized in Exhibit 9.

In Exhibit 10, the bars show the relative errors on the estimated constant coefficient, ν_0, as the number of simulations increases. In addition, the standard deviations of the relative errors are displayed as lines with values on the right hand axis. Relative errors and standard deviations of the relative errors are shown for both exercise date 1 and exercise date 2. As can be seen, both the relative error and the standard deviation of the relative error decrease and approach zero as the number of simulations increases. It can also be seen that the relative errors are lower for exercise date 2 than for exercise date 1, and that the standard deviation of the relative errors are lower on exercise date 2 than on exercise date 1. The only major exception for this pattern is the case with 100 simulations, where the relative error of ν_0 is lower on exercise date 1 than on exercise date 2. It is also worth nothing that, as the number of simulations increases, the differences in terms of relative errors and standard deviations of the relative errors between exercise date 1 and exercise date 2, vanish. The reason for this phenomenon is that for a small number of simulations, the spot rates are more dispersed throughout the state space on exercise date 2 than they are on exercise date 1. This is illustrated in Exhibit 11 where the solution to the spot rate process in Equation (36) has been simulated on exercise date 1 and exercise date 2 respectively. Note also that the standard deviations of the relative errors are very high for a small number of simulations.

EXHIBIT 8

Testcase3—Model Price-Spot Rate Relation and Estimated Price-Spot Rate Relation of the Underlying Coupon Bond at Exercise Data 2. Initial Spot Rate is 5%.

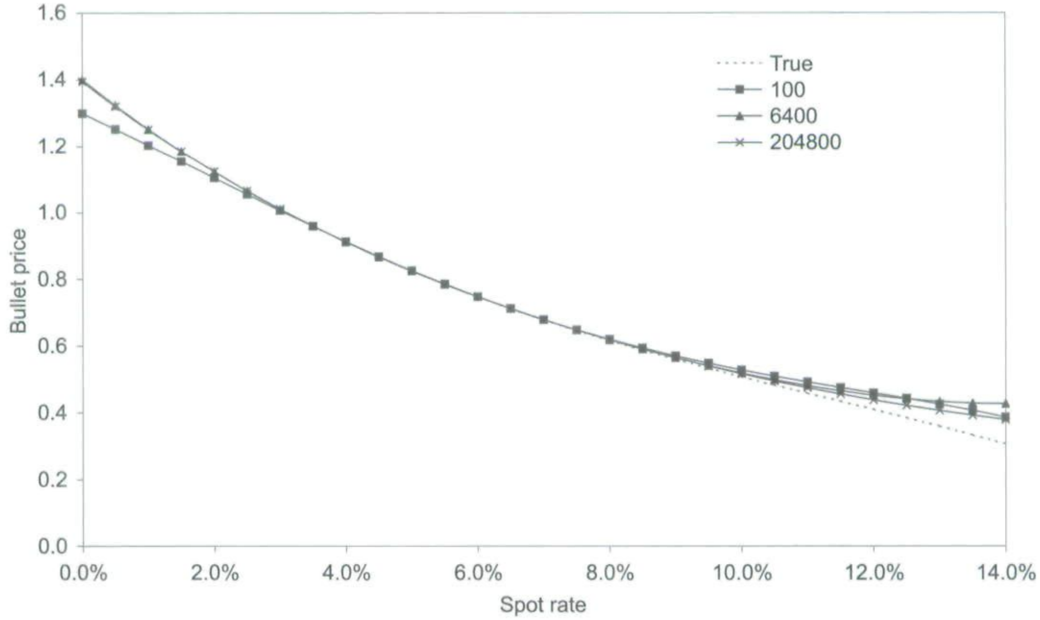


EXHIBIT 9

True Coefficients

Coefficient	Spot rate	Value at exercise date1	Value at exercise date 2
v_0	r^0	1.4046	1.3955
v_1	r^1	-15.6877	-15.4399
v_2	r^2	102.4994	99.7405
v_3	r^3	-475.1468	-456.6217
v_4	r^4	1704.1267	1,616.1862

EXHIBIT 10

Testcase 3—Estimated v_0 Coefficient

Estimated constant at exercise date 1 and 2. The bars show the average relative error and are related to the left axis. The lines show the standard deviation of the estimate and are related to the right axis. The sample size is 1,000.

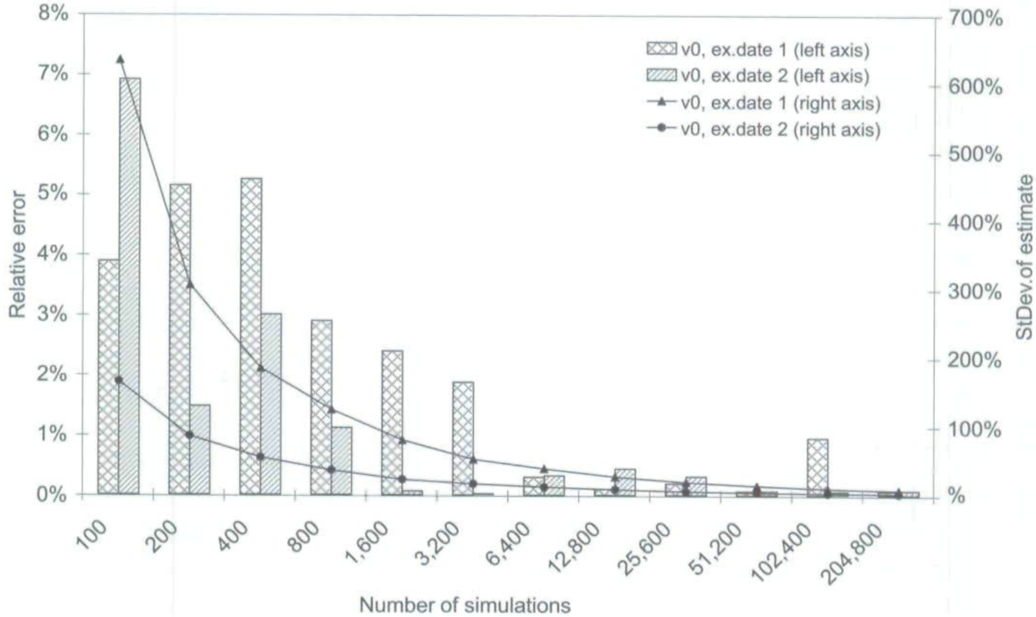
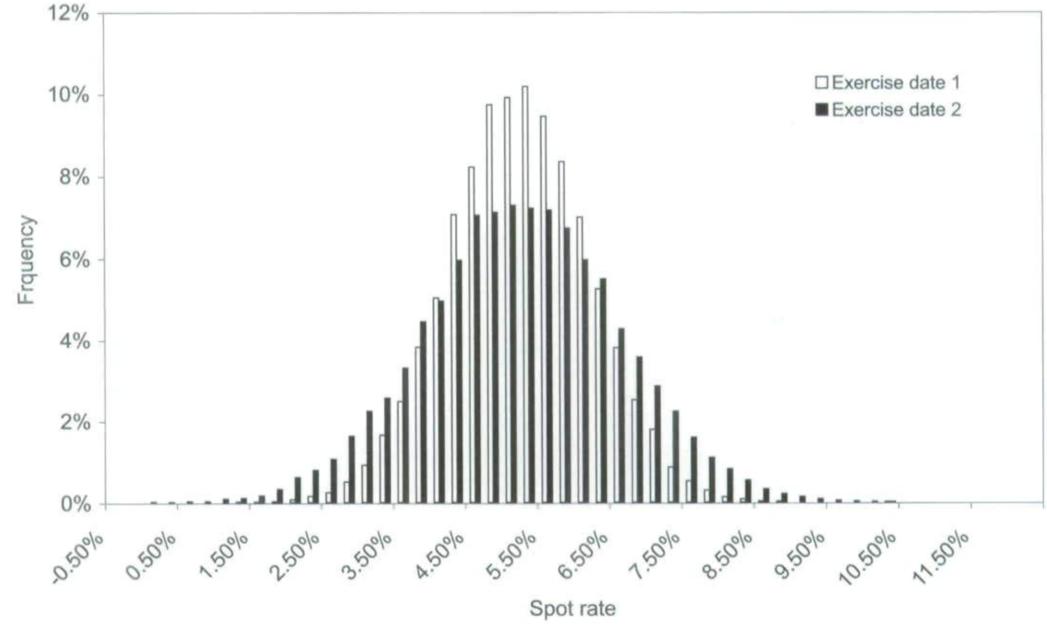


EXHIBIT 11
Simulated Hull—White Spot Rate Distribution

Simulated spot rate distribution at exercise date 1 and exercise date 2 respectively for the parameter values in Testcase3. 30.000 simulations have been used.



The conclusions made for ν_0 also apply for the estimated coefficients ν_1, ν_2, ν_3, and ν_4. This is illustrated in Exhibit 12, Exhibit 13, Exhibit 14, and Exhibit 15. The differences are that the level for both the relative error and the standard deviation of the relative error is much higher, the higher the power of the spot rate. This suggests that a large part of variations on prices of the underlying coupon bond at exercise date 1 and exercise date 2 can be explained by the first basis functions in the approximation of the price given by (44) for this simple underlying security. This would probably not be the case if the underlying security were more complex, as would be the case for a mortgage-backed bond. However, as the number of simulations increases, the estimates become more and more accurate, because the spot rates become more dispersed throughout the state space. Our results are consistent with the result in Glasserman and Yu [2004], saying that the number of simulations needed grows exponentially in the number of basis functions.

CONCLUSION

In this article we develop an algorithm aimed at simulating Bermudan option prices written on securities that must also be priced by simulations. The method extends the algorithm developed in Longstaff and Schwartz [2001] by applying the bias reduction technique from Huge and Rom-Poulsen [2004] to reduce the bias in the option price induced by using noisy price estimates of the underlying security in the dynamic programming recursion. We show that if the price of the underlying security belongs to a space spanned by some basis functions, the result of the algorithm approaches the solution to the programming problem in Longstaff and Schwartz [2001]. Clemente, Lamberton, and Protter [2002] prove that this solution approaches the true option price as the number of basis functions and paths approaches infinity. However, generally the spanning assumption is not fulfilled, and the bias reduction thus introduces an additional approximation besides those introduced using the algorithm in Longstaff and Schwartz [2001].

Generally, the method can be applied when an unbiased estimate for the value of the underlying security exists. Thus, the method can also be used for pricing compound options since, in these cases, the underlying option may be simulated and an unbiased estimate exists.

Exhibit 12

Testcase3—Estimated v_1 Coefficient

Estimated v_1 at exercise date 1 and 2. The bars show the average relative error and are related to the left axis. The lines show the standard deviation of the estimate and are related to the right axis. The sample size is 1,000.

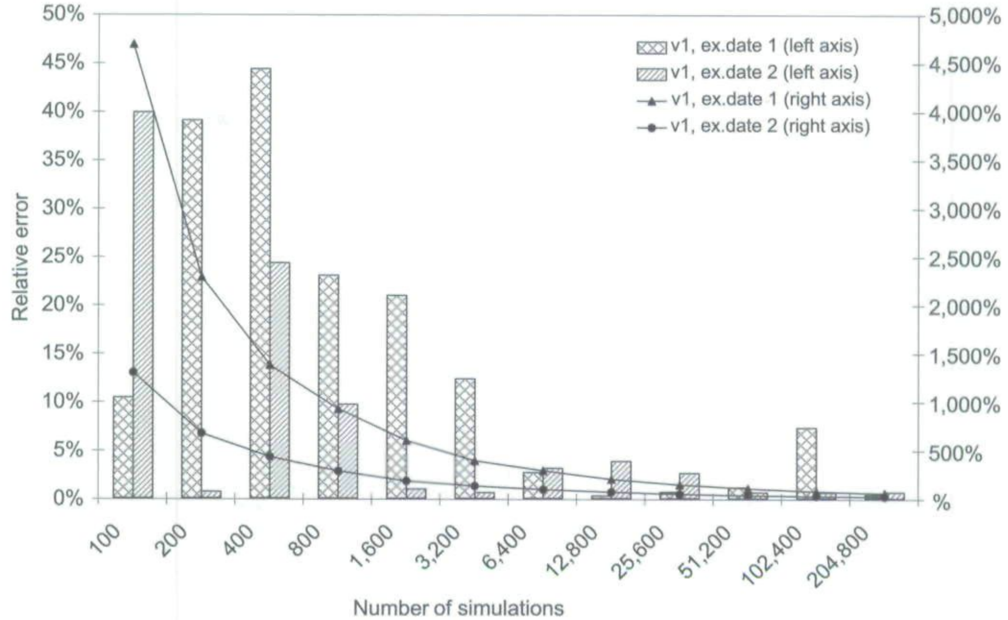


Exhibit 13

Testcase3—Estimated v_2 Coefficient

Estimated v_2 at exercise date 1 and 2. The bars show the average relative error and are related to the left axis. The lines show the standard deviation of the estimate and are related to the right axis. The sample size is 1,000.

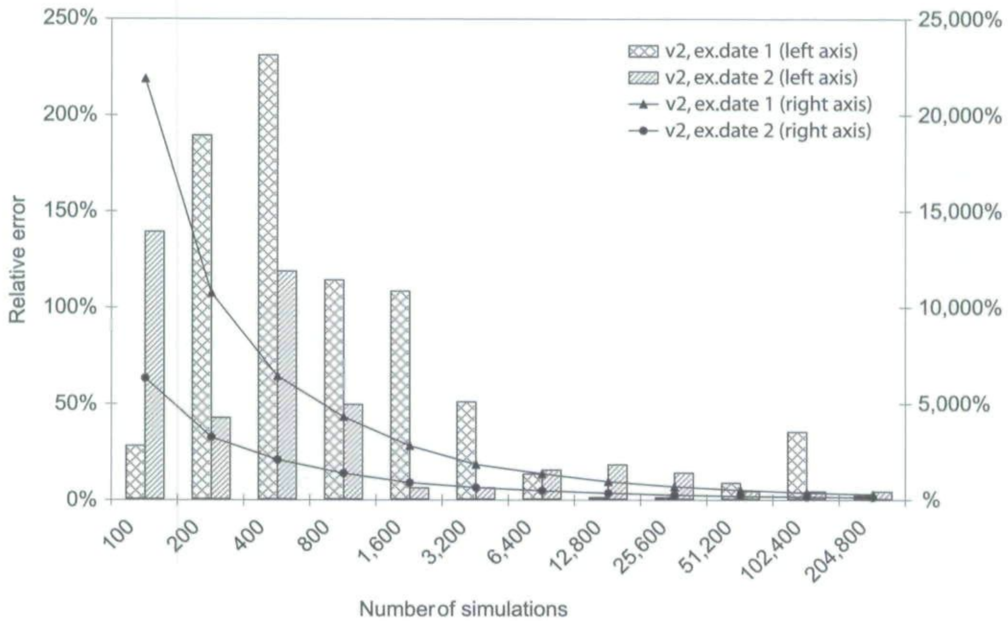


EXHIBIT 14

Testcase3—Estimated v_3 Coefficient

Estimated v_3 at exercise date 1 and 2. The bars show the average relative error and are related to the left axis. The lines show the standard deviation of the estimate and are related to the right axis. The sample size is 1,000.

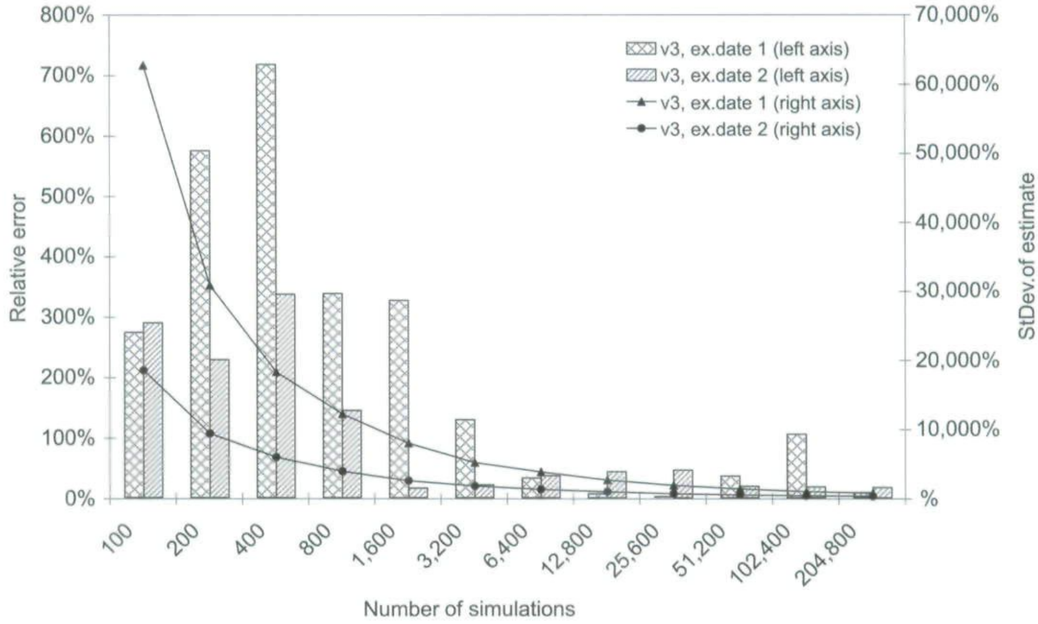
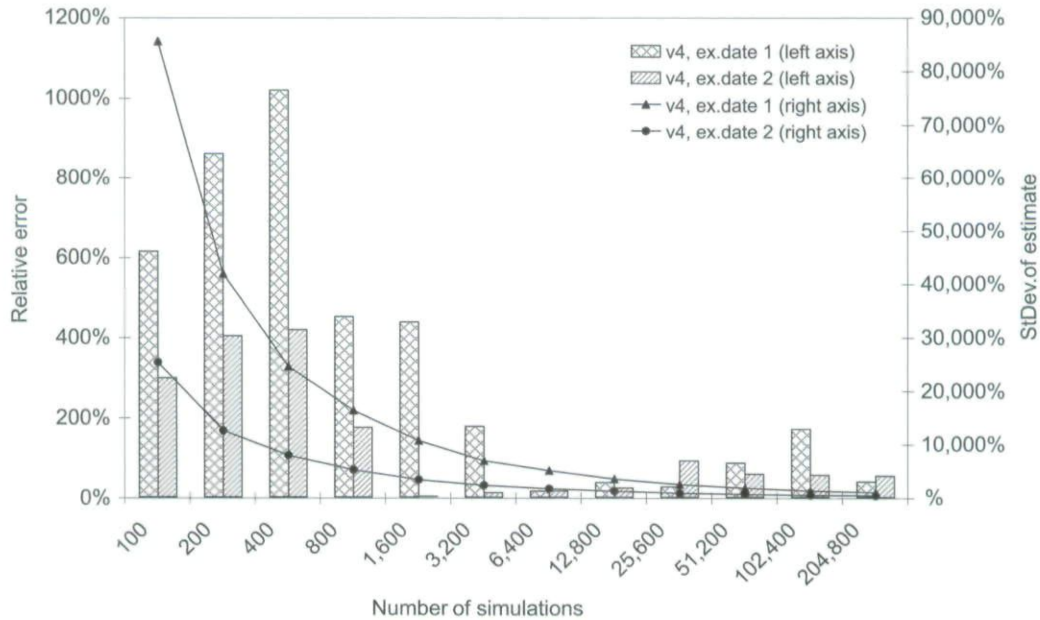


EXHIBIT 15

Testcase3—Estimated v_4 Coefficient

Estimated v_4 at exercise date 1 and 2. The bars show the average relative error and are related to the left axis. The lines show the standard deviation of the estimate and are related to the right axis. The sample size is 1,000.



We demonstrate, using simple test cases, that the algorithm is capable of computing Bermudan option prices that are very close to the option prices computed using a PDE approach.

The estimated coefficients to the basis functions seem to approach their true values, especially for the coefficients multiplying low powers of the spot rate. However, for a small number of simulations, the standard deviation of the estimated coefficient is extremely high. Many simulations are needed in order to get accurate coefficient estimates. This is in line with the findings in Glasserman and Yu [2004].

Using many simulations yields relatively accurate coefficient estimates and this results in relatively good approximations of the relations between the price of the underlying and the spot rate. However, for a small number of simulations and short term exercise dates, the relation between the price of the underlying and the spot rate is only accurate in non-extreme regions of the state space. Thus, one must be sure to use a sufficient number of paths when simulating Bermudan option prices with a strike corresponding to extreme values of the state variables.

APPENDIX A

PROOFS

Proof of Proposition 1

We want to show that the solution to the dynamic programming problem in the section on Extension of the LSMC Method approaches the solution to the problem in the section on LSMC Simulations when the approximation in Equation (30) is used. In the following, we emphasize the dependence of the number of paths U used in the simulation of the state space and thus used in Equation (30) to estimate the coefficients in the least squares price estimate. By Assumptions 1 and 2, the price of the underlying security fulfils $P_i^{LS}(X_i;U) \to P_i(X_i)$ $\forall$ i as U increases. Thus we have $\forall$ i

$$\hat{h}_i^{LS}(X_i;U) \to h_i(X_i) \text{ as } U \to \infty$$

By definition $C_L^{LS}(X_L;U) = C_L(X_L) = 0$ and $V_L^{LS}(X_L;U) = \hat{h}_L^{LS}(X_L;U) \to h_L(X_L) = V_L(X_L)$ as $U \to \infty$. The proof now goes by induction. Assume that $V_j^{LS}(X_j;U) \to V_j(X_{ij})$ as $U \to \infty$, for $j = i+1, \ldots, L$. If we can prove that $C_i^{LS}(X_i;U) \to C_i(X_i)$ as $U \to \infty$, then we have $V_i^{LS}(X_i;U) \to V_i(X_i)$ as $U \to \infty$ and we are done. Now from (23)

$$C_i^{LS}(X_i;U) = \sum_{r=1}^{R} \zeta_{ir}^{LS}(U)\psi_{ir}(X_i)$$

where

$$\zeta_i^{LS}(U) = (\mathbb{E}_i^{\mathbb{Q}}[\psi_i(X_i)\psi_i(X_i)^{\top}])^{-1} \mathbb{E}_i^{\mathbb{Q}}[\psi_i(X_i)V_{i+1}^{LS}(X_{i+1};U)]$$

If

$$\mathbb{E}_i^{\mathbb{Q}}[\psi_i(X_i)V_{i+1}^{LS}(X_{i+1};U)] \to \mathbb{E}_i^{\mathbb{Q}}[\psi_i(X_i)V_{i+1}(X_{i+1})] \text{ as } U \to \infty \tag{46}$$

then $\zeta_i^{LS}(U) \to \zeta_i$ as $U \to \infty$, which implies that $C_i^{LS}(X_i;U) \to C_i(X_i)$ as $U \to \infty$

Since $\psi_i(X_i)V_{i+1}^{LS}(X_{i+1};U)$ defines a sequence of stochastic variables as U increases, a sufficient condition for (46) is (32) and (33).

APPENDIX B

The Simulation Algorithm

As shown in Huge and Rom-Poulsen [2004] it is sufficient to have $N = 1$ in Equation (30) in order to get a good approximation for the price of the underlying security if the number of crude Monte Carlo price estimates (U) is sufficiently high. This can be used to construct a particular simple algorithm for pricing Bermudan options on securities that are priced by simulations. We start by sampling the underlying state space and store payments made by the underlying security (coupons, prepayments, etc.). For each generated path, at an exercise date t_i, the price of the underlying security is given as the time t_i value of future payments along that path. Thus, the crude Monte Carlo price of the underlying security is found using one single path. Before computing option pay-offs we improve the price estimate of the underlying security by regressing the crude Monte Carlo computed prices onto a set of state dependent basis functions. The estimated expression is then used to find the state dependent price of the underlying security, and this price is used to compute option pay-offs. Also, we run a regression in order to find the exercise boundary at time t_i, just as in the Longstaff and Schwartz [2001] algorithm. The Bermudan option value at time t_i is then given as the maximum of the continuation value and the option's approximated intristic value at time t_i. The recursion is carried out for all exercise dates from option expiration until today. In case of a Bermudan option on a bond, the state space is sampled from today until the maturity of the bond, not only until option expiration.

Algorithm 1

Extended Least Squares Monte Carlo

for $(j = 1 \text{ to } U)$ **do**

  for $(I = 1 \text{ to } M)$ **do**

    Simulate and store X_{ij}

    Compute and store $\hat{b}_{ij}^{MC}(X_{ij})$

  end for

end for

for $(i = M - 1 \text{ to } 1)$ **do**

  for $(j = 1 \text{ to } U)$ **do**

    if $(i = M - 1)$ **then**

      $\hat{P}_{M,j}^{MC} = \hat{P}_{Mj}^{MC}$

    end if

    $\hat{P}_{ij}^{MC}(X_{ij}) = P_{ij}^{zcb}(i+1)\, \hat{P}_{i+1,j}^{MC} + \hat{b}_{i,j}^{MC}$

  end for

  if $(t_i \in \mathcal{T})$ **then**

    $\hat{\hat{\upsilon}}_i = (\frac{1}{U}\Sigma_{j=1}^{U}\, \phi(X_{ij})\, \phi(X_{ij})^{\top})^{-1}(\frac{1}{U}\Sigma_{j=1}^{U}\, \phi(X_{ij})\, \hat{P}_{ij}^{MC}(X_{ij}))$

    for $(j = 1 \text{ to } U)$ **do**

      $\hat{P}_{ij}^{LS}(X_{ij}) = \phi(X_{ij})^{\top}\, \hat{\hat{\upsilon}}_i$

    end for

    if $(i = L)$ **then**

      for $(j = 1 \text{ to } U)$ **do**

        $\hat{C}_{ij}^{LS}(X_{ij}) = \max(\hat{P}_{ij}^{LS}(X_{ij}) - K, 0)$

      end for

    else

      $\hat{\zeta}_i = (\frac{1}{U}\Sigma_{j=1}^{U}\psi(X_{ij})\psi(X_{ij})^{\top})^{-1}(\frac{1}{U}\Sigma_{j=1}^{U}\psi(X_{ij})\, P_{ij}^{zcb}(i+1)\, \hat{C}_{i+1,j}(X_{i+1,j}))$

      for $j = 1 \text{ to } U$ **do**

        $\hat{h}_{ij}^{LS}(X_{ij}) = \max(\hat{P}_{ij}^{LS}(X_{ij}) - K, 0)$

        if $(\hat{h}_{ij}^{LS}(X_{ij}) \geq \psi(X_{ij})^{\top}\hat{\zeta}_i)$ **then**

          $\hat{C}_{ij}^{LS}(X_{ij}) = \hat{h}_{ij}^{LS}(X_{ij})$

        else

          $\hat{C}_{ij}^{LS}(X_{ij}) = P_{ij}^{zcb}(i+1)\, \hat{C}_{i+1,j}^{LS}(X_{i+1,j})$

        end if

      end for

    end if

  end if

end for

$\hat{C}_0^{LS}(X_0) = \frac{1}{U}\Sigma_{j=1}^{U} P_{0j}^{zcb}(1)\, \hat{C}_{1j}^{LS}(X_{1j})$

In Algorithm 1, $\hat{b}_{ij}^{MC}$ denotes the crude Monte Carlo simulated payment from the underlying security at time i along path j. Recall that the set S is constructed so that both the payment dates and the exercise dates belong to S, and the algorithm thus iterates over the dates in S. In general the subscript $_{ij}$ denotes the value of some security at time i along path j whereas the subscript i alone denotes a $U \times 1$ vector with j'th row equal to the value along the j'th path (ij). Values of the underlying security are denoted by $\hat{P}_{ij}^{MC}$ and along the j'th path, prices of zero coupon bonds for the period (t_i, t_k) are denoted by $\hat{P}_{ij}^{zcb}(k)$.

ENDNOTES

We are gratefull for comments from Stephen Figlewski, Carsten Sørensen and an anonymous referee. All errors are of course our own.

[1]Using a parameterized exercise strategy creates a downward bias in the underlying Bermudan swaption price due to the non-optimality of the used exercise strategy. However, given the true exercise strategy the Monte Carlo estimated price is an unbiased estimate.

[2]Such dates could for example be payment dates of a callable mortgage backed bond, which will infuence future values of the bond, because future payments from the bond might be dependent on historical prepayments.

[3]We will not go into detail on how to price callable MBS but instead refer to Huge and Rom-Poulsen [2004], which contains a detailed account on the MBS pricing model.

[4]The option price computed by solving the fundamental Partial Differential Equation. We have used a Crank-Nicholson scheme to solve the partial differential equation.

REFERENCES

Barraquand, J., and D. Martineau. "Numerical Valuation of High Dimensional Multivariate American Securities." *Journal of Financial and Quantitative Analysis*, 30 (1995), pp. 383–405.

Björk, T. *Arbitrage Theory in Continuous Time,* 1st ed. Oxford University Press, 1998.

Bossaerts, P. "Simulation Estimators of Optimal Early Exercise." Working Paper, Graduate School of Industrial Administration, Carnegie Mellon University, 1989.

Broadie, M., and P. Glasserman. "Pricing American-Style Securities using Simulation." *Journal of Economic Dynamics and Control,* 21 (1997), pp. 1323–1352.

——. "A Stochastic Mesh Method for Pricing High Dimensional American Options." *Journal of Computational Finance,* 7 (2004), pp. 35–72.

Carriere, J. "Valuation of the Early-Exercise Price for Options Using Simulations and Non-parametric Regression." *Insurance: Mathematics and Economics,* 19 (1996), pp. 19–30.

Clemente, E., D. Lamberton, and P. Protter. "An Analysis of a Least Squares Regression Method for American Option Pricing." *Finance and Stochastics,* 6 (2002), pp. 449–471.

Dufe, D. *Dynamic Asset Pricing Theory.* 2nd ed., Princeton University Press, 1996.

Fu, M., S. Laprise, D. Madan, Y. Su, and R. Wu. "Pricing American Options: A Comparison of Monte Carlo Simulation Approaches." *Journal of Computational Finance,* 4 (2001), pp. 39–88.

Glasserman, P. Monte Carlo Methods in Financial Engineering. 1st ed., Springer, 2004.

Glasserman, P., and B. Yu. "Number of Paths Versus Number of Basis Functions in American Option Pricing." *The Annals of Applied Probability,* 4 (2004), pp. 2090–2119.

Huge, B., and N. Rom-Poulsen. "Bias Reduction in European Option Pricing." Working Paper, Copenhagen Business School, 2004.

Longstaff, F., and E. Schwartz. "Valuing American Options by Simulation: A Simple Least-Squares Approach." *The Review of Financial Studies,* 14 (2001), pp. 113–147.

Omberg, E. "The Valuation of American Put Options With Exponential Exercise Polices." *Advances in Futures and Options Research,* 2 (1987), pp. 117–142.

Rogers, L. "Monte Carlo Valuation of American Options." *Mathematical Finance,* 12 (2002), pp. 271–286.

Tilley, J. "Valuing American Options in a Path Simulation Model." *Transactions of the Society of Actuaries,* 45 (1993), pp. 83–104.

Tsitsiklis, J., and B. Van Roy. "Regression Methods for Pricing Complex American Style Options." *IEEE Transactions on Neural Networks,* 12 (2001), pp. 694–703.