# A SIMPLE AND EFFICIENT NUMERICAL METHOD FOR PRICING DISCRETELY MONITORED EARLY-EXERCISE OPTIONS

MIN HUANG AND GUO LUO

ABSTRACT. We present a simple, fast, and accurate method for pricing a variety of discretely monitored options in the Black-Scholes framework, including autocallable structured products, single and double barrier options, and Bermudan options. The method is based on a quadrature technique, and it employs only elementary calculations and a fixed one-dimensional uniform grid. The convergence rate is  $O(1/N^4)$  and the complexity is  $O(MN \log N)$ , where N is the number of grid points and M is the number of observation dates.

### 1. INTRODUCTION

Exotic options are commonly traded throughout the world. Many popular exotic options are path-dependent and have early-exercise features. These options can often be priced using analytical formulas if they are continuously monitored (e.g. barrier options). In practice, however, most path-dependent exotic options are discretely monitored [4], in which case they need to be priced using numerical techniques. Due to the complicated structures of these options, traditional pricing models based on Monte-Carlo simulations and finite difference methods are often too time-consuming to be useful in practical situations. More recent pricing methods based on advanced mathematical techniques, on the other hand, tend to be more efficient (e.g. [14, 12, 10, 11]), but for many financial institutions, these methods are often too difficult to understand and to properly implement. To strike a balance between model performance and practical utility, we propose a new quadrature-based method that is much faster and more accurate than the traditional Monte-Carlo and PDE methods, yet at the same time is easy to understand and to implement. We will first give a brief review of the types of products considered, as well as the quadrature-based pricing model which is the foundation of our work. Then we will explain our method and provide numerical examples.

1.1. Autocallable Structured Products. Autocallable structured products belong to a class of exotic options with early-exercise features. Many different types of autocallable products have been created and traded in financial markets, and they have become increasingly popular in recent years. We refer to the Appendix of [9] for a description of the main features of various autocallable products.

We will consider a very common autocallable product with discrete observation dates. At each observation date there is a pre-specified barrier level. If the price of

Key words and phrases. Discrete option pricing, quadrature method, autocallable structured product, single and double barrier option, Bermudan option.

Disclaimer: The statements made and opinions expressed here are solely our own, and do not reflect the views of China Mechants Bank or its employees and affiliates.

#### MIN HUANG AND GUO LUO

the underlying asset is greater (less) than or equal to the barrier level (depending on the terms of the product), the option is exercised and a pre-specified fixed-rate return is paid. If the asset price is below (above) the barriers at all observation dates, the option is never exercised and the investor receives a negative return at maturity. In addition, autocallable products may have a knock-in feature. In this case, if the option is never exercised, the negative return the investor receives depends on whether the asset price at maturity reaches a pre-specified knock-in level. While the value of a continuously monitored autocallable product has a simple closed-form solution, the value of a discretely monitored autocallable product cannot be calculated easily. In the discrete case, there exist analytical solutions in terms of multiple integrals, cf. [25, 26, 9]. The numerical calculation of these integrals, however, can become prohibitive if the number of observation dates exceeds five. In practice, discretely monitored autocallable products are commonly priced using Monte-Carlo simulations. This method is straightforward, but convergence is usually slow and acceleration techniques such as variance reduction are often needed (cf. [2, 13]). Another popular method for pricing discretely monitored autocallable products is to solve the governing Black-Scholes partial differential equation (PDE) using finite difference method (cf. [9]). Assuming a second-order central difference approximation in space, the overall convergence rate of a typical finite difference based pricing method is  $O((\delta x)^2 + \delta t)$  if the explicit forward Euler method is used in time, and  $O((\delta x)^2 + (\delta t)^2)$  if the implicit Crank-Nicolson method is used. Since two-dimensional grids are needed for finite difference methods, computational complexities are at least of order  $1/(\delta x \delta t)$ . In addition, since the payoffs of autocallable products are discontinuous (in asset price), additional care (such as smoothing of payoff functions) must be taken to ensure the accuracy of any finite difference approximations.

## **Remark 1.1.** Some autocallable products may have payoffs at maturity that are of the same type as that of European vanilla options. These products can be effectively viewed as combinations of autocallable products and barrier options (see below).

1.2. **Discrete Barrier Options.** Barrier options are among the most popular types of exotic options. A barrier option may be activated (knock-in option) or deactivated (knock-out option) when the price of the underlying asset crosses certain barrier levels. Barrier options may be discretely or continuously monitored. A single barrier option has one barrier at each observation date, while a double barrier option has two barriers at each observation date. The final payoff of a barrier option (if it is active at maturity) may be of the same type as that of a vanilla option or that of a digital option. A special type of barrier option has a constant amount of cash as the final payoff, and such an option is called a touch option. Most barrier options have time-independent barrier levels, but options with time-dependent barrier levels have also been studied [7].

Similar to the case of autocallable structured products, there are closed-form solutions for discrete barrier options in terms of multiple integrals [23], but such solutions are often difficult to evaluate directly. In practice, discrete barrier options can be priced using Monte-Carlo simulations or standard binomial tree methods, but these methods are usually slow [5]. Other methods that have been proposed to price discrete barrier options include continuity correction approximations [4, 27], Wiener-Hopf methods [14], adaptive mesh methods [1], Hilbert transform methods [12], finite element methods [16], Fourier-cosine series expansion methods [10], and quadrature methods [3, 6]. These methods, while useful in certain contexts, have not been as widely used as the traditional Monte-Carlo and finite difference methods, usually due to their complexity.

1.3. Bermudan Options. Bermudan options are discrete versions of American options. A Bermudan option can be exercised at any of the prescribed observation dates, and the payoff is of the same type as that of a vanilla option. Similar to discrete barrier options, Bermudan options can be priced using Monte-Carlo simulations [17], Hilbert transform methods [11], Fourier-cosine series expansion methods [10], and quadrature methods [22, 21, 6].

1.4. Overview of the Quadrature Method. Among the various methods proposed to price discretely monitored options, the quadrature method is particularly appealing because of its high efficiency and accuracy. The method has been applied to discrete barrier options and Bermudan options [3, 22, 21, 6]. The main idea is to solve for option values at each observation date via backward induction in time. The risk-neutral valuation formula is expressed as a single integral, which is then evaluated numerically to produce the option price. Specifically, let V denote the value of the option, S the value of the underlying asset, r the risk-neutral interest rate,  $t_1, \ldots, t_M$  the observation dates, and  $\mathbb{E}$  the risk-neutral expectation. If the underlying asset S does not trigger an early exercise at  $t_m$ , we have

$$V(t_m, S) = e^{-r(t_{m+1} - t_m)} \mathbb{E} [V(t_{m+1}, \cdot) | S]$$
  
=  $e^{-r(t_{m+1} - t_m)} \int_0^\infty V(t_{m+1}, y) f(y|S) \, dy,$ 

where f(y|S) is a probability density function whose form depends on the model of the underlying asset. If S triggers an early exercise at  $t_m$ , on the other hand, the option price  $V(t_m, S)$  would be equal to a prescribed value. The integral above can be calculated using FFT [21, 22] or Fast Gauss Transform [6]. Since, however,  $V(t_{m+1}, y)$  is discontinuous (in y) for autocallable products and barrier options, and non-differentiable (in y) for Bermudan options, care must be taken to ensure the accuracy of the numerical evaluation of the integral. While several shifted or nonuniform grids have been designed in previous studies to address this difficulty [21, 6], the problem becomes particularly challenging when multiple discontinuities are present at each observation point, for instance in the case of double barrier options with time-dependent barrier levels.

1.5. Motivation of Our Work. Although a good number of advanced techniques have been proposed to improve pricing models' accuracy and efficiency, in most practical situations, simple methods that are more cost effective are usually preferred over their more sophisticated counterparts. The reason for this is twofold. First of all, when data quality is not high enough, sophisticated models are not necessarily beneficial. For instance, interest rates and volatilities are crucial components of nearly every pricing model, but they need to be estimated from available market data. If the estimated parameters contain large errors, which is not uncommon in products sold in emerging markets, any advantages gained from the use of sophisticated models may be (more than) offset by these errors, making the simpler models more attractive. Secondly, implementations of pricing models usually involve staff members from multiple business departments, and the resulting products often need active maintenance and updates. As a result, models that are

#### MIN HUANG AND GUO LUO

too complicated in nature may hinder effective business communications, which increases maintenance costs and operational risks. In view of these considerations, it is not difficult to see why traditional Monte-Carlo and PDE methods are still among the most popular methods in the valuation of discretely monitored options, even though their computational costs are already high enough to adversely impact their applicability in business.

In view of the practical concerns mentioned above, we propose a new quadrature method to price the aforementioned discretely monitored options in the Black-Scholes framework. The convergence rate of our method is  $O(1/N^4)$  and the complexity is  $O(MN \log N)$ , where N is the number of grid points and M is the number of observation dates. The performance of our method is on par with previous quadrature-based methods such as the CONV method [21], but it is more straightforward, and is better suited for products with multiple discontinuities. Our method differs from other quadrature methods mainly in three aspects. First, we work with probability density functions directly instead of using characteristic functions or Toeplitz matrices. Secondly, we use only a fixed one-dimensional uniform grid to compute all integrals. Thirdly, we utilize explicit Black-Scholes formulas to improve the accuracy of the calculations. Due to these novel modifications, our method is very easy to implement, and is capable of handling sophisticated products such as double-barrier options with time-dependent barrier levels.

1.6. Organization of the Paper. The rest of the paper is organized as follows. Section 2 specifies the class of (discrete) option pricing problems that our quadrature method is intended to solve, and Section 3 presents the main recursion formula for our method. After detailing the implementation of our method in Section 4–5, we summarize the algorithm in Section 6 and then present numerical examples in Section 7. The Appendix collects a few useful theoretical results, which lay the foundation of a class of (discrete) option pricing algorithms (including the one described here) but which, to our best knowledge, do not seem to have been rigorously proved or even properly formulated in the literature. We supply the proofs here in the hope that they would be useful to interested readers.

## 2. Basic Assumptions

We assume that the price of the underlying asset S(t) satisfies the following stochastic differential equation in the risk-neutral measure:

(2.1) 
$$dS(t) = \left[r(t) - q(t)\right]S(t) dt + \sigma(t)S(t) dW(t),$$

where r(t) is the risk-neutral interest rate, q(t) is the yield rate,  $\sigma(t)$  is the volatility, and W(t) is the Wiener process.

In practice, interest rates are always time dependent. Yield rates for FX products are simply foreign interest rates, and for other types of products they may be implied from futures prices. Thus yield rates are usually time dependent as well. Implied volatilities are time dependent, whereas historical volatilities can often be taken as constant.

The solution to (2.1) is

(2.2) 
$$S(t) = S(t_0) \exp\left\{\int_{t_0}^t \left[r(s) - q(s) - \frac{1}{2}\sigma^2(s)\right] ds + \int_{t_0}^t \sigma(s) \, dW(s)\right\},$$

where  $t_0$  is the present date. We consider a discretely monitored option with observation dates  $t_1, \ldots, t_M$ , where the last observation date  $t_M$  is the maturity date. It follows from (2.2) that each  $S(t_m)$  for  $1 \le m \le M$  has the lognormal distribution

(2.3) 
$$S(t_m) \sim S(t_0) \exp\left\{\int_{t_0}^{t_m} \left[r(s) - q(s) - \frac{1}{2}\sigma^2(s)\right] ds + \left(\int_{t_0}^{t_m} \sigma^2(s) \, ds\right)^{1/2} Z\right\},$$

where Z denotes the standard normal distribution. Now define

$$r_m = \int_{t_{m-1}}^{t_m} \frac{r(s)}{\Delta t_m} \, ds, \quad q_m = \int_{t_{m-1}}^{t_m} \frac{q(s)}{\Delta t_m} \, ds, \quad \sigma_m^2 = \int_{t_{m-1}}^{t_m} \frac{\sigma^2(s)}{\Delta t_m} \, ds,$$

for  $1 \leq m \leq M$  where  $\Delta t_m = t_m - t_{m-1}$ , and define piecewise constant functions

$$\tilde{r}(t) = r_m, \quad \tilde{q}(t) = q_m, \quad \text{and} \quad \tilde{\sigma}(t) = \sigma_m, \quad \text{for} \quad t_{m-1} < t \le t_m.$$

For the process

$$d\tilde{S}(t) = \left[\tilde{r}(t) - \tilde{q}(t)\right]\tilde{S}(t)\,dt + \tilde{\sigma}(t)\tilde{S}(t)\,dW(t)$$

since

$$\int_{t_0}^{t_m} r(s) \, ds = \sum_{n=1}^m \int_{t_{n-1}}^{t_n} r(s) \, ds = \sum_{n=1}^m r_n \Delta t_n = \int_{t_0}^{t_m} \tilde{r}(s) \, ds,$$
$$\int_{t_0}^{t_m} q(s) \, ds = \sum_{n=1}^m \int_{t_{n-1}}^{t_n} q(s) \, ds = \sum_{n=1}^m q_n \Delta t_n = \int_{t_0}^{t_m} \tilde{q}(s) \, ds,$$

and

$$\int_{t_0}^{t_m} \sigma^2(s) \, ds = \sum_{n=1}^m \int_{t_{n-1}}^{t_n} \sigma^2(s) \, ds = \sum_{n=1}^m \sigma_n^2 \Delta t_n = \int_{t_0}^{t_m} \tilde{\sigma}^2(s) \, ds,$$

it follows that  $\hat{S}(t_m)$  has the same distribution as  $S(t_m)$  in (2.3) for each m. Since the value of the option depends only on probability distributions of the asset price at observation dates, the option value remains the same if we replace the process S by the process  $\tilde{S}$ . In other words, we may safely assume that r(t), q(t), and  $\sigma(t)$ are piecewise constant functions. Thus in what follows, we shall assume

$$r(t) = r_m, \quad q(t) = q_m, \quad \text{and} \quad \sigma(t) = \sigma_m, \qquad \text{for} \qquad t_{m-1} < t \le t_m.$$

Consider now a general class of discretely monitored options with barriers. Since the sum of a knock-in barrier option and a knock-out barrier option with the same observation dates and barrier levels is a vanilla option (or a digital option if the barrier options are digital), to study the pricing of these discretely monitored options, it suffices to consider a knock-out barrier option which ceases to exist when barrier levels are crossed. To this end, assume that

- (A) The option has two strike prices  $K_m^-, K_m^+ \in [0, \infty]$ , with  $K_m^- \leq K_m^+$ , at each observation date  $t_m, m = 1, 2, ..., M$ .
- (B) The option is exercised if  $S \leq K_m^-$  or  $S \geq K_m^+$  at some  $t_m$ , and the payoffs are given by  $a_m^-S + b_m^-$  (if  $S \leq K_m^-$ ) and  $a_m^+S + b_m^+$  (if  $S \geq K_m^+$ ), respectively, for some  $a_m^\pm$ ,  $b_m^\pm \in \mathbb{R}$ .
- (C) The final payoff at maturity is

$$V(t_M, S) = a_M S + b_M$$
, for  $K_M^- < S < K_M^+$ .

These assumptions are general enough to cover a wide class of discretely monitored options, such as the ones mentioned in the introduction. For instance, common up-and-out autocallable products would have

$$1 \le m \le M: \quad 0 < K_m^+ < \infty, \ K_m^- = 0, \ a_m^+ = 0, \ b_m^+ > 0;$$
  
$$m = M: \quad a_M = 0, \ b_M < 0.$$

Down-and-out put barrier options would have

$$1 \le m \le M - 1: \quad K_m^+ = \infty, \ 0 < K_m^- < \infty, \ a_m^- = 0, \ b_m^- = 0;$$
  
$$m = M: \quad K_M^- = 0, \ 0 < K_M^+ < \infty, \ a_M = -1, \ b_M = K_M^+,$$
  
$$a_M^+ = b_M^+ = 0.$$

Double barrier knock-out call options would have

$$1 \le m \le M - 1: \quad 0 < K_m^{\pm} < \infty, \ a_m^{\pm} = 0, \ b_m^{\pm} = 0;$$
  
$$m = M: \quad K_M^{+} = \infty, \ 0 < K_M^{-} < \infty, \ a_M = 1, \ b_M = -K_M^{-},$$
  
$$a_M^{-} = b_M^{-} = 0.$$

Bermudan put options with strike K would have

$$1 \le m \le M$$
:  $K_m^-$ : the unique solution of  $K - K_m^- = V(t_m, K_m^-)$ ,  
 $K_m^+ = \infty, \ a_m^- = -1, \ b_m^- = K;$   
 $m = M$ :  $a_M = 0, \ b_M = 0.$ 

We will give a proof of the uniqueness of  $K_m^{\pm}$  for Bermudan options in the Appendix. To summarize, our basic assumptions are

- (1) The underlying asset price S follows a geometric Brownian motion with piecewise constant interest rates, yield rates, and volatilities.
- (2) There are finitely many observation points, and two exercise levels (possibly  $\infty$ ) at each observation point. If S is above the upper exercise level or below the lower exercise level at any observation point, the option is exercised and the payoff is a linear function in S.
- (3) At maturity, if S is between the two exercise levels, a payoff is incurred which is also a linear function in S.

#### 3. Outline of the method

Let V(t, S) denote the value of the option (as a function of asset price S) at any time t, and let

$$V_m(S) = V(t_m, S), \qquad m = 0, 1, \dots, M,$$

denote the value of the option at the observation dates. Our goal is to find  $V_0(S(t_0))$ , and our strategy is to use backward induction in time. Since  $V_M(S)$  is piecewise linear in S,  $V_{M-1}(S)$  has a simple explicit expression. For each  $m = M - 1, \ldots, 1$ , we write  $V_{m-1}(S)$  as the risk-neutral expectation of  $V_m(S)$  for  $K_{m-1}^- < S < K_{m-1}^+$ , and as  $a_{m-1}^{\pm}S + b_{m-1}^{\pm}$  otherwise. The expectation is given by an explicit integral and is calculated numerically. The core of the quadrature method is the calculation of M - 1 expectation integrals step-by-step. Let

$$\tau_m = \frac{1}{2} \sigma_m^2 \Delta t_m = \frac{1}{2} \sigma_m^2 (t_m - t_{m-1}).$$

For each  $1 \le m \le M - 1$ , note that  $S(t_m)$  has a lognormal distribution as in (2.3). The relevant probability density functions are known to be [19]

(3.1) 
$$\rho_m(y,S) = \frac{1}{2\sqrt{\pi\tau_m}y} \exp\left\{-\frac{1}{4\tau_m} \left(\log\frac{y}{S} - \frac{2}{\sigma_m^2} \left[r_m - q_m - \frac{1}{2}\sigma_m^2\right]\tau_m\right)^2\right\}.$$

For simplicity of notations we define  $K_0^+ = \infty$  and  $K_0^- = 0$ . By the fundamental theorems of asset pricing, we have the risk-neutral pricing formula [24]:

$$V_{m-1}(S) = e^{-2r_m\tau_m/\sigma_m^2} \mathbb{E}[V_m(\cdot)|S] = e^{-2r_m\tau_m/\sigma_m^2} \int_0^\infty V_m(y)\rho_m(y,S) \, dy$$
$$= \frac{e^{-2r_m\tau_m/\sigma_m^2}}{2\sqrt{\pi\tau_m}} \int_0^\infty \frac{1}{y} V_m(y) \exp\left\{-\frac{1}{4\tau_m} \left(\log\frac{y}{S} - \frac{2}{\sigma_m^2} \left[r_m - q_m - \frac{1}{2}\sigma_m^2\right]\tau_m\right)^2\right\} \, dy$$

for  $K_{m-1}^- < S < K_{m-1}^+$  and  $1 \le m \le M - 1$ . By Assumption (B) from Section 2, we also have

(3.3) 
$$V_{m-1}(S) = \begin{cases} a_{m-1}^{-}S + b_{m-1}^{-}, & S \le K_{m-1}^{-} \\ a_{m-1}^{+}S + b_{m-1}^{+}, & S \ge K_{m-1}^{+} \end{cases}$$

To further study the formulas (3.2) and (3.3), we first recall some classical results on the pricing of binary options.

**Lemma 3.1.** Let K > 0, and let  $\chi_A$  denote the characteristic function of a set A. Consider an option with no early-exercise features.

(1) If the option has payoff  $\hat{V}_m(y) = \chi_{[K,\infty)}y$ , then  $\hat{V}_{m-1}(S) = V_m^a(S,K,1)$ . (2) If  $\hat{V}_m(y) = \chi_{(0,K]}y$ , then  $\hat{V}_{m-1}(S) = V_m^a(S,K,-1)$ . (3) If  $\hat{V}_m(y) = \chi_{[K,\infty)}$ , then  $\hat{V}_{m-1}(S) = V_m^b(S,K,1)$ . (4) If  $\hat{V}_m(y) = \chi_{(0,K]}$ , then  $\hat{V}_{m-1}(S) = V_m^b(S,K,-1)$ .

The functions  $V_m^a$  and  $V_m^b$  are defined as

$$V_m^a(S,K,\epsilon) = e^{-2q_m\tau_m/\sigma_m^2} SN(\epsilon d_1), \qquad V_m^b(S,K,\epsilon) = e^{-2r_m\tau_m/\sigma_m^2} N(\epsilon d_2),$$

where N is the cumulative normal distribution function, and

$$d_1 = \frac{1}{\sqrt{2\tau_m}} \left( \log \frac{S}{K} + \frac{2}{\sigma_m^2} \left[ r_m - q_m + \frac{1}{2} \sigma_m^2 \right] \tau_m \right), \qquad d_2 = d_1 - \sqrt{2\tau_m}.$$

*Proof.* By definition  $V_m^a$  is the value of an asset-or-nothing option, and  $V_m^b$  is the value of a cash-or-nothing option. The valuation formulas are just standard results for binary options [18].

**Remark 3.2.** The standard Black-Scholes formulas in Lemma 3.1 ignore the possible effects of volatility smiles. If such effects need to be taken into account, one may amend the definitions of  $V_m^a$  and  $V_m^b$  (as given in the Lemma) by incorporating suitable vega-induced correction terms [15]. For instance, the value of a cash-or-nothing call option in the presence of volatility smiles would become

$$V_{\text{smile}} = V_{\text{no smile}} - \frac{\partial V_{\text{vanilla}}}{\partial \sigma} \frac{\partial \sigma}{\partial K}.$$

We can use Lemma 3.1 to obtain an explicit formula for the value of  $V_{M-1}(S)$ . By Assumption (B)–(C) from Section 2, we have

(3.4) 
$$V_M(S) = \begin{cases} a_M^- S + b_M^-, & S \le K_M^- \\ a_M S + b_M, & K_M^- < S < K_M^+ \\ a_M^+ S + b_M^+, & S \ge K_M^+ \end{cases}$$

Without loss of generality we may assume  $0 < K_M^{\pm} < \infty$ , since otherwise we may choose some arbitrary  $0 < K_M^{\pm} < \infty$  and set  $a_M^{\pm} = a_M$ ,  $b_M^{\pm} = b_M$ .

**Proposition 3.3.** The value of the option at  $t_{M-1}$  is given by

$$\begin{split} \tilde{V}_{M-1}(S) &= a_M^- V_M^a(S, K_M^-, -1) + b_M^- V_M^b(S, K_M^-, -1) \\ &+ a_M \left[ V_M^a(S, K_M^-, 1) - V_M^a(S, K_M^+, 1) \right] \\ &+ b_M \left[ V_m^b(S, K_M^-, 1) - V_M^b(S, K_M^+, 1) \right] \\ &+ a_M^+ V_M^a(S, K_M^+, 1) + b_M^+ V_M^b(S, K_M^+, 1), \end{split}$$

for  $K_{M-1}^- < S < K_{M-1}^+$ .

*Proof.* Clearly, the option from  $t_{M-1}$  to  $t_M$  is equivalent to a linear combination of binary options consisting of two put options with strike  $K_M^-$ , two call options with strike  $K_M^-$  and four call options with strike  $K_M^+$ . The result then follows from Lemma 3.1.

With the aid of (3.2) and Proposition 3.3, we may write the main recursion of our quadrature method as follows.

**Proposition 3.4.** Let  $\tilde{V}_M = V_M$  be defined in (3.4), and  $\tilde{V}_{m-1}$  be given by the following recursion formula:

$$(3.5) \quad \tilde{V}_{m-1}(S) = e^{-2r_m\tau_m/\sigma_m^2} \int_{K_m^-}^{K_m^+} \tilde{V}_m(y)\rho_m(y,S) \, dy + a_m^+ V_m^a(S,K_m^+,1) + b_m^+ V_m^b(S,K_m^+,1) + a_m^- V_m^a(S,K_m^-,-1) + b_m^- V_m^b(S,K_m^-,-1),$$

for  $1 \leq m \leq M$ . Then we have  $\tilde{V}_m(S) = V_m(S)$  for  $K_m^- < S < K_m^+$  and  $0 \leq m \leq M$ . In particular,  $\tilde{V}_0(S(t_0)) = V_0(S(t_0))$ .

*Proof.* This is merely the classical recursion formula for quadrature methods specialized to the Black-Scholes model. To prove the formula, we only need to show

(3.6) 
$$V_m(S) = \tilde{V}_m(S)\chi_{(K_m^-,K_m^+)} + (a_m^+S + b_m^+)\chi_{[K_m^+,\infty)} + (a_m^-S + b_m^-)\chi_{(0,K_m^-)},$$

for all  $0 \le m \le M$ . By assumption (3.6) is true for m = M. Assume now (3.6) holds for some  $1 \le m \le M$ . Substituting the equation into (3.2), applying Lemma 3.1, comparing the result with (3.5) and using (3.3), we observe that (3.6) holds for m-1. The result then follows from induction.

**Remark 3.5.** Recursion formula (3.5) lies at the heart of our quadrature method and distinguishes our method from other quadrature methods, which are primarily based on (3.2) or one of its many variants. The significance of the formula (3.5) lies in the fact that it makes explicit use of Black-Scholes formulas to separate the expectation integral  $\mathbb{E}[V_m(\cdot)|S]$  into a "quadrature part"

$$F_{m-1}(S) = e^{-2r_m\tau_m/\sigma_m^2} \int_{K_m^-}^{K_m^+} \tilde{V}_m(y)\rho_m(y,S) \, dy,$$

and an "early-exercise part"

$$E_{m-1}(S) = a_m^+ V_m^a(S, K_m^+, 1) + b_m^+ V_m^b(S, K_m^+, 1) + a_m^- V_m^a(S, K_m^-, -1) + b_m^- V_m^b(S, K_m^-, -1).$$

Since the function  $\tilde{V}_m(S)$  is smooth for  $S \in (K_m^-, K_m^+)$  (in fact for all  $S \in (0, \infty)$ , as we will show below), the integral  $F_{m-1}(S)$  can be evaluated accurately and efficiently using a high-order quadrature method such as Simpson's rule. In contrast, the integrand  $V_m(S)$  in the original recursion formula (3.2) is discontinuous on  $(0, \infty)$ (in either  $V_m$  itself or in its first derivative); this makes the accurate evaluation of the expectation integral a difficult and challenging task. Although (3.5) applies specifically to the Black-Scholes model, the same idea can be used for other asset price models, as long as a suitable analytical formula (exact or approximate) can be found for the probability density function  $\rho_m(y, S)$  and early-exercise part  $E_{m-1}(S)$ .

With  $\tilde{V}_{M-1}(S)$  given in Proposition 3.3, Proposition 3.4 implies that we may apply (3.5) successively to obtain  $\tilde{V}_{M-2}(S), \tilde{V}_{M-3}(S), \ldots, \tilde{V}_0(S)$ . The value of the option is equal to  $\tilde{V}_0(S(t_0))$ .

## 4. Details of Implementation

In (3.5),  $\tilde{V}_{m-1}(S)$  is written as a sum of explicit functions and an integral, namely

(4.1) 
$$\tilde{V}_{m-1}(S) = a_m^+ V_m^a(S, K_m^+, 1) + b_m^+ V_m^b(S, K_m^+, 1) + a_m^- V_m^a(S, K_m^-, -1) + b_m^- V_m^b(S, K_m^-, -1) + F_{m-1}(S),$$

where

$$F_{m-1}(S) = e^{-2r_m\tau_m/\sigma_m^2} \int_{K_m^-}^{K_m^+} \tilde{V}_m(y)\rho_m(y,S) \, dy.$$

We may truncate the integral by replacing its upper and lower bounds by

$$L_m^+ = \min\{K_m^+, S(t_0)C\},$$
 and  $L_m^- = \max\{K_m^-, S(t_0)/C\},$ 

respectively, where C > 1 is a suitable constant. In practice, the choice

$$\log C = 10\sigma_0\sqrt{t_M - t_0} + \left(1 + \frac{1}{2}\sigma_0^2\right)(t_M - t_0),$$

where  $\sigma_0 = \max_{1 \le m \le M} \sigma_m$ , is sufficient to reduce the truncation errors to roundoff level. Heuristically this is clear from (2.3), which suggests that the chance that  $S(t_m)$  move outside the range  $(S(t_0)/C, S(t_0)C)$  is negligibly small. The rigorous derivation of the error bounds can be obtained using a recursive argument, as will be explained in the Appendix.

Now we consider the truncated integral

$$\tilde{F}_{m-1}(S) = e^{-2r_m\tau_m/\sigma_m^2} \int_{L_m^-}^{L_m^+} \tilde{V}_m(y)\rho_m(y,S) \, dy.$$

If  $K_m^- \ge S(t_0)C$  or  $K_m^+ \le S(t_0)/C$ , then by convention the integral is zero. Thus in what follows we shall assume  $K_m^- < S(t_0)C$  and  $K_m^+ > S(t_0)/C$ . Let

$$B_m^{\pm} = \log \frac{L_m^{\pm}}{S(t_0)}$$

and denote

$$S = S(t_0)e^x, \qquad y = S(t_0)e^z,$$
  

$$\alpha_m = \frac{1}{\sigma_m^2} [r_m - q_m - \frac{1}{2}\sigma_m^2], \qquad \beta_m = \frac{1}{\sigma_m^4} [r_m - q_m - \frac{1}{2}\sigma_m^2]^2 + \frac{2r_m}{\sigma_m^2}$$
  

$$u_m(x) = \tilde{V}_m(S(t_0)e^x), \qquad w_m(x) = \exp\left\{-\frac{x^2}{4\tau_m} - \alpha_m x\right\}.$$

The truncated integral can be rewritten as

(4.2) 
$$\tilde{F}_{m-1}(S(t_0)e^x) = \frac{e^{-\beta_m \tau_m}}{2\sqrt{\pi\tau_m}} \int_{B_m^-}^{B_m^+} w_m(x-z)u_m(z) \, dz.$$

One can show by differentiating (4.2) that  $\tilde{F}_m$ , and thus  $\tilde{V}_m$  and  $u_m$ , are smooth functions in x. This means we can compute the integrals efficiently using a high-order quadrature such as Simpson's rule.

In general,  $B_m^{\pm}$  are different for different values of m, so they cannot all be placed on one grid. Now we choose a uniform grid  $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}$ , where  $x_1 = -\log C$  and  $x_N = \log C$ . Let

$$h = \frac{x_N - x_1}{N - 1} = \frac{2 \log C}{N - 1}.$$

For each m, let

$$p_m^- = \min\{i \colon x_i \ge B_m^-\}, \qquad p_m^+ = \max\{i \colon x_i < B_m^+\},$$

where by definition  $p_m^- \ge 1$  and  $p_m^+ < N$ . Since we will use Simpson's rule which requires an odd number of grid points, we define

$$p_0 = (p_m^+ - p_m^-) \mod 2,$$

and rewrite (4.2) as

(4.3)

$$\tilde{F}_{m-1}(S(t_0)e^x) = \frac{e^{-\beta_m \tau_m}}{2\sqrt{\pi\tau_m}} \left( \int_{x_{p_m^-}}^{x_{p_m^++p_0}} w_m(x-z)u_m(z) \, dz + \int_{B_m^-}^{x_{p_m^-}} w_m(x-z)u_m(z) \, dz + \int_{x_{p_m^++p_0}}^{B_m^+} w_m(x-z)u_m(z) \, dz \right).$$

For each  $2 \le m \le M - 1$ , we will compute  $\tilde{F}_{m-1}(S(t_0)e^x)$  for all

$$x \in \{x_1, x_2, \dots, x_N, B_{m-1}^-, B_{m-1}^+, \xi_{m-1}^-, \xi_{m-1}^+\}$$

where

$$\xi_{m-1}^{-} = \frac{1}{2} \left( x_{p_{m-1}^{-}} + B_{m-1}^{-} \right), \qquad \xi_{m-1}^{+} = \frac{1}{2} \left( x_{p_{m-1}^{+} + p_{0}} + B_{m-1}^{+} \right).$$

For m = 1 we only need to compute  $\tilde{F}_{m-1}(S(t_0)e^x)$  for x = 0, since the value of the option is given by  $\tilde{V}_0(S(t_0))$ .

4.1. Computation of the first integral in (4.3). To compute the first integral in (4.3) using Simpson's rule, we let

$$U_m(i) = \begin{cases} u_m(x_i), & i = p_m^-, p_m^+ + p_0 \\ 4u_m(x_i), & i = p_m^- + 1, p_m^- + 3, \dots, p_m^+ + p_0 - 1 \\ 2u_m(x_i), & i = p_m^- + 2, p_m^- + 4, \dots, p_m^+ + p_0 - 2 \end{cases}$$

The integral is discretized as

(4.4) 
$$\int_{x_{p_m^-}}^{x_{p_m^++p_0}} w_m(x-z)u_m(z)\,dz = \frac{h}{3}\sum_{i=p_m^-}^{p_m^++p_0} w_m(x-x_i)U_m(i) + O(h^4),$$

since Simpson's rule is of order 4 [8]. Note that  $U_m(i)$  is known from the previous step (or by Proposition 3.3 for m = M - 1) for all i = 1, 2, ..., N. For  $x \in \{B_{m-1}^{\pm}, \xi_{m-1}^{\pm}, 0\}$ , the sum (4.4) can be computed directly with complexity O(N). For all grid points  $x \in \{x_1, x_2, ..., x_N\}$ , on the other hand, the sum (4.4) can be computed altogether using FFT with complexity  $O(N \log N)$ . This latter fact is crucial to the efficient implementation of our quadrature method and is a consequence of the following simple observation.

**Proposition 4.1.** Define (2N-1)-periodic grid functions  $\hat{z}$ ,  $\hat{U}_m$ , and  $\hat{F}_m$  by

$$\hat{z}(i) = z_i = -2\log C + (i-1)h, \qquad 1 \le i \le 2N - 1,$$
$$\hat{U}_m(i) = \begin{cases} 0, & 1 \le i < p_m^- \\ U_m(i), & p_m^- \le i \le p_m^+ + p_0 \\ 0, & p_m^+ + p_0 < i \le 2N - 1 \end{cases},$$
$$\hat{F}_m = \mathcal{F}^{-1} \Big\{ \mathcal{F}\big(w_m(\hat{z})\big) \mathcal{F}(\hat{U}_m) \Big\},$$

where  $\mathcal{F}$  and  $\mathcal{F}^{-1}$  denote the discrete Fourier transform and the inverse discrete Fourier transform of size 2N - 1, respectively. Then

$$\int_{x_{p_m^-}}^{x_{p_m^++p_0}} w_m(x_j-z)u_m(z) \, dz = \frac{h}{3} \, \hat{F}_m(j+N) + O(h^4),$$

for all  $1 \leq j \leq N$ . The above discrete Fourier transforms and inverse discrete Fourier transform can be calculated using FFT, and the total computational complexity is  $O(N \log N)$ .

*Proof.* We consider the discrete convolution

$$G_m(j) = \sum_{i=1}^{2N-1} w_m(z_{j-i})\hat{U}_m(i),$$

for  $j \in \mathbb{Z}$ . Note that by definition,

$$z_{j+N-i} = -2\log C + (j-i+N-1)h = (j-i)h = x_j - x_i,$$

for all  $1 - N \leq j - i \leq N - 1$ . Thus for  $1 \leq j \leq N$ , we have

$$G_m(j+N) = \sum_{i=1}^{2N-1} w_m(z_{j+N-i})\hat{U}_m(i)$$
  
= 
$$\sum_{i=p_m^-}^{p_m^++p_0} w_m(z_{j+N-i})U_m(i) = \sum_{i=p_m^-}^{p_m^++p_0} w_m(x_j-x_i)U_m(i).$$

Therefore (4.4) with  $x = x_j$  can be written as

(4.5) 
$$\int_{x_{p_m^-}}^{x_{p_m^++p_0}} w_m(x_j-z)u_m(z)\,dz = \frac{h}{3}\,G_m(j+N) + O(h^4).$$

The discrete convolution  $G_m$  can be calculated using FFT as

$$G_m = \mathcal{F}^{-1} \Big\{ \mathcal{F} \big( w_m(\hat{z}) \big) \mathcal{F}(\hat{U}_m) \Big\} = \hat{F}_m,$$

with a complexity of  $O(N \log N)$  [8].

**Remark 4.2.** Our method differs from other well-known FFT-based methods (such as [21, 22]) in that we express the discrete quadrature rule (4.4) directly in terms of discrete Fourier transforms, instead of applying continuous Fourier transform to the integral and then discretizing the Fourier integrals (in other words, we have exchanged the order of Fourier transform and discretization). The direct application of the discrete Fourier transform (to the discrete quadrature rule) not only eliminates the need for artificially-introduced damping factors, which are required for the existence of the continuous Fourier transforms, but also eliminates the need for additional specially-designed computational grids which are required to satisfy Nyquist relations. This enables us to carry out the main recursion (3.5) on a fixed uniform grid, without any additional artificial parameters.

4.2. Computation of the last two integrals in (4.3). The last two integrals in (4.3) are calculated in similar ways using Simpson's rule. First, note that we may use Proposition 3.3 to calculate  $\tilde{V}_{M-1}(S(t_0)e^x)$  for  $x \in \{B_{M-1}^{\pm}, \xi_{M-1}^{\pm}\}$ . Generally all four points are needed if the option has two barriers, and only two are needed if the option has one barrier. For each  $2 \leq m \leq M-1$ , assume  $u_m(x) = \tilde{V}_m(S(t_0)e^x)$  has been calculated for  $x \in \{B_m^{\pm}, \xi_m^{\pm}\}$ . The last two integrals in (4.3) are calculated using Simpson's rule as follows:

$$(4.6) \\ \int_{B_m^-}^{x_{p_m^-}} w_m(x-z)u_m(z) dz = \frac{1}{6} (x_{p_m^-} - B_m^-) [w_m(x-B_m^-)u_m(B_m^-) + 4w_m(x-\xi_m^-)u_m(\xi_m^-) + w_m(x-x_{p_m^-})u_m(x_{p_m^-})] + O(h^4),$$

$$(4.7) \\ \int_{x_{p_m^++p_0}}^{B_m^+} w_m(x-z)u_m(z) dz = \frac{1}{6} (B_m^+ - x_{p_m^++p_0}) [w_m(x-B_m^+)u_m(B_m^+) + 4w_m(x-\xi_m^+)u_m(\xi_m^+) + w_m(x-x_{p_m^++p_0})u_m(x_{p_m^++p_0})] + O(h^4).$$

12

Unlike autocallable products and barrier options, Bermudan options do not have pre-specified exercise levels. Instead, one needs to solve for  $K_m^{\pm}$  from the equations

$$\tilde{V}_m(K_m^+) = K_m^+ - K,$$

for call options and

$$\tilde{V}_m(K_m^-) = K - K_m^-,$$

for put options, where  $\tilde{V}_m$  is determined by (3.5). For simplicity we assume the yield rates  $q_m \geq 0$ , which is almost always the case in practice. We will demonstrate how to find  $K_m^-$ , as the same procedure applies to  $K_m^+$ . Let

$$p = \min\{i : \tilde{V}_m(S(t_0)e^{x_i}) > K - S(t_0)e^{x_i}\}$$

If p = 1 there is no early exercise, so  $K_m^- = 0$ . Otherwise we have  $S(t_0)e^{x_{p-1}} \leq K_m^- < S(t_0)e^{x_p}$  by Corollary 8.3. The value of  $K_m^-$  can be found using classical root-finding methods such as the bisecting method or the secant method. Note that the bisecting method is guaranteed to converge by Corollary 8.3, and it takes  $O(\log N)$  steps to reduce the error of the approximate root to an order of  $O(h^4)$ . Since the cost for calculating  $\tilde{V}_m$  at one point using (4.1) is O(N), the total cost for finding the optimal exercise price is  $O(N \log N)$ . The secant method is superlinear and converges faster than the bisecting method, though its error estimates are not as straightforward.

### 6. Summary of the Algorithm

We summarize our algorithm as follows:

- 1: Define the functions  $V^a_m, V^b_m$  as in Lemma 3.1
- 2: Define the function  $\tilde{V}_{M-1}$  as in Proposition 3.3
- 3: if option style is Bermudan then
- 4: Calculate  $K_{M-1}^{\pm}$  as in Section 5
- 5: **end if**
- 6: Calculate  $p_{M-1}^{\pm}$  and  $p_0$  to find the bounds of integration in (4.3)
- 7: Use Proposition 3.3 to compute  $\tilde{V}_{M-1}(S(t_0)e^x)$  for  $x \in \{B_{M-1}^{\pm}, \xi_{M-1}^{\pm}\}$ , and assign their values to  $v_1^{\pm}, v_2^{\pm}$  respectively
- 8: Define a vector **S** as  $S(i) \leftarrow S(t_0)e^{x_i}$  for i = 1, 2, ..., N
- 9: Define a vector  $\mathbf{y}$  as  $y(i) \leftarrow \tilde{V}_{M-1}(S(i))$  for i = 1, 2, ..., N
- 10: for m = M 1 downto 2 do
- 11: Let  $\hat{z}$  and  $\hat{U}_m$  be as defined in Proposition 4.1
- 12:  $\hat{F}_m \leftarrow \mathcal{F}^{-1} \Big\{ \mathcal{F} \big( w_m(\hat{z}) \big) \mathcal{F}(\hat{U}_m) \Big\}$
- 13: Define (or redefine) the vector  $\mathbf{Y}_1$  as  $Y_1(j) \leftarrow \frac{h}{3}\hat{F}_m(j+N)$  for  $j = 1, 2, \dots, N$
- 14: Use (4.6), (4.7), and the values of  $v_1^{\pm}, v_2^{\pm}, y(p_m^-), y(p_m^+ + p_0)$  to compute the last two integrals in (4.3) at  $x_i$  for i = 1, 2, ..., N, and assign their values to  $\mathbf{Y}_2$  and  $\mathbf{Y}_3$
- 15: if option style is Bermudan then
- 16: Calculate  $K_{m-1}^{\pm}$  as in Section 5
- 17: end if
- 18: Calculate  $p_{m-1}^{\pm}$  and  $p_0$  to find the bounds of integration in (4.3)

19: Compute  $\tilde{V}_{m-1}(S(t_0)e^x)$  for  $x \in \{B_{m-1}^{\pm}, \xi_{m-1}^{\pm}\}$  using (4.1) (where the first integral in (4.3) is computed using (4.4), and the last two integrals using (4.6), (4.7), and the existing values of  $v_1^{\pm}, v_2^{\pm}, y(p_m^-), y(p_m^+ + p_0)$ ), and assign their values to  $v_1^{\pm}, v_2^{\pm}$  respectively

$$\begin{split} \mathbf{y} \leftarrow \mathbf{Y_1} + \mathbf{Y_2} + \mathbf{Y_3} &+ a_m^+ V_m^a(\mathbf{S}, K_m^+, 1) + b_m^+ V_m^b(\mathbf{S}, K_m^+, 1) \\ &+ a_m^- V_m^a(\mathbf{S}, K_m^-, -1) + b_m^- V_m^b(\mathbf{S}, K_m^-, -1), \end{split}$$

as in (4.1) (note that **y** now stores  $\tilde{V}_{m-1}(S(i))$  for i = 1, 2, ..., N) 21: end for

22: Compute  $\tilde{V}_0(S(t_0))$  using (4.1), where the first integral in (4.3) is computed using (4.4), and the last two integrals using (4.6), (4.7), and the existing values of  $v_1^{\pm}, v_2^{\pm}, y(p_1^-), y(p_1^+ + p_0)$ 

Since the computational complexity of each step of the loop is  $O(N \log N)$ , the total complexity is  $O(MN \log N)$ .

**Remark 6.1.** While other quadrature methods typically employ multiple uniform grids or specially-designed (nonuniform or shifted) grids, our method utilizes only a fixed one-dimensional uniform grid, which not only eliminates the need for complicated inter-grid data transfer procedures, but also eliminates the need for special subroutines that are often required to interpolate data across discontinuities. This makes our method particularly easy to implement.

### 7. Numerical Examples

We will demonstrate the accuracy and efficiency of the proposed method using two examples, in which the value of an autocallable structured product and that of a double barrier option with time-dependent barriers are found.

7.1. Example 1: Autocallable Structured Product. We consider a knock-out autocallable structured product maturing in one year. The price of the underlying asset is 3000, the nominal amount is 1, and the volatility is 20%. The observation dates (in years from now), barrier levels, and risk-free rates (in %) are given below in Table 1.

Observation date	Barrier level	Risk-free rate
0.2	3050	2
0.4	3100	2.1
0.6	3150	2.2
0.8	3200	2.3
1	3250	2.4

TABLE 1. An autocallable structured product.

If the asset price reaches or goes above the barrier level at some observation date t, the investor receives a payment of  $4\% \times t$ . If the asset price is below the barrier at every observation date, the investor will have to pay a premium of 1%. The relative errors of the computed option values with varying grid sizes are shown

14



FIGURE 1. Errors for the autocallable structured product, computed using (a) the proposed method and (b) Monte-Carlo simulations.

below in Figure 1(a), where the exact option value is taken to be the one computed on the grid of size 70,001.

As a comparison, the relative errors of the option values computed using Monte-Carlo simulations with antithetic variates technique are shown in Figure 1(b). As is clear from the figures, the error of the option value computed using the proposed method is well within  $10^{-5}$  with just 501 points in the grid, and drops very quickly as the grid size increases. In contrast, it takes more than ten million paths for Monte-Carlo simulations to reduce the error of the computed option value to within  $10^{-3}$ , and the error decays very slowly as the number of paths increases.

Figure 2 below shows the CPU time used by the proposed method to price the autocallable structured product, where the code is developed in Matlab and is run on a personal computer. As is clear from the figure, the CPU time required by



FIGURE 2. CPU time for autocallable structured product valuation, using the proposed method.

the proposed method is well within 0.01 seconds, and it increases approximately linearly as grid size increases. It is difficult to compare the speed of the proposed

method with that of Monte-Carlo simulations directly, since the CPU time required by the latter depends largely on specific implementations. Nevertheless, the typical CPU time consumed by a Monte-Carlo simulation with tens of millions of paths ranges from tens of seconds to a few minutes.

7.2. Example 2: Double Barrier Option. As another example, consider a knock-out double barrier put option with time-dependent barrier levels. The price of the underlying asset is 2500, the strike price is 2600, the nominal amount is 1, and the volatility is 25%. The option matures in two years. The observation dates (in years from now), barrier levels, and risk-free rates (in %) are given below in Table 2.

Observation date	Barrier level 1	Barrier level $2$	Risk-free rate
0.25	2200	2800	1
0.50	2100	2900	1.1
0.75	2000	3000	1.2
1	1900	3100	1.3
1.25	1800	3200	1.2
1.50	1700	3300	1.3
1.75	1600	3400	1.4
2	_	_	1.5

TABLE 2. A double barrier option.

If the asset price falls below barrier level 1 or rises above barrier level 2 at any observation date, the option ceases to exist. If the option is still valid at maturity, the payoff is the same as that of a vanilla put option. The relative errors of the computed option values with varying grid sizes are shown below in Figure 3(a), where the exact option value is taken to be the one computed on the grid of size 50,001.



FIGURE 3. Errors for the double barrier option, computed using (a) the proposed method and (b) Monte-Carlo simulations.

As a comparison, the relative errors of the option values computed using Monte-Carlo simulations with antithetic variates technique are shown in Figure 3(b). As is clear from the figures, the error of the option value computed using the proposed method is within  $10^{-5}$  with just 701 points in the grid, and drops very quickly as the grid size increases. In contrast, it takes more than ten million paths for Monte-Carlo simulations to reduce the error of the computed option value to within  $10^{-3}$ , and the error decays very slowly as the number of paths increases.

Figure 4 below shows the CPU time used by the proposed method to price the double barrier option, where the code is developed in Matlab and is run on a personal computer. As is clear from the figure, the CPU time required by the



FIGURE 4. CPU time for double barrier option valuation, using the proposed method.

proposed method is within 0.01 seconds, and it increases approximately linearly as grid size increases. In contrast, the typical CPU time consumed by a Monte-Carlo simulation with tens of millions of paths ranges from tens of seconds to a few minutes.

**Remark 7.1.** Although the designed order of the proposed method is 4, in the above numerical examples, a lower order of convergence (close to 3) is actually observed for the grid sizes considered. It is interesting to note that this apparent "loss" of order of accuracy is not a defect of our method; rather, it is a manifestation of the subtle influences that barrier levels can have on option pricing algorithms. These influences can be understood from two perspectives. First, in the above examples, the barrier levels  $K_m^{\pm}$  are close to the spot price  $S(t_0)$ . This gives rise to a relatively small integration domain  $[B_m^-, B_m^+]$  compared with the entire computational domain  $[-\log C, \log C]$  (recall that

$$B_m^+ - B_m^- = \log \frac{L_m^+}{S(t_0)} - \log \frac{L_m^-}{S(t_0)} = \log \frac{L_m^+}{L_m^-} \le \min \left\{ \log \frac{K_m^+}{K_m^-}, \log \frac{K_m^+}{C^{-1}} \right\}$$

), which means that the set of grid points that are available for the discrete quadrature rule (4.4) represents only a relatively small fraction of the set of grid points introduced on the entire computational domain. Secondly, in the above examples the option prices  $V_{m+1}(S)$  contain discontinuities at each observation date  $t_{m+1}$ . These discontinuities necessarily show up in the form of large gradients in the (smooth) functions  $\tilde{V}_m(S)$  (via the expectation integrals  $\mathbb{E}[V_{m+1}(\cdot)|S]$ ), which means that the discrete quadrature rule (4.4) is being applied to fast-varying functions with only a relatively small number of grid points, leaving the integrands only marginally resolved and hence explaining the degeneracy observed in the convergence rate. If the barrier levels  $K_m^{\pm}$  are pushed farther away from the spot price  $S(t_0)$ , so that the option becomes increasingly like a vanilla option, then the discrete quadrature rule (4.4) effectively applies to slow-varying functions with a relatively large number of grid points, which improves the resolution of the integrands and hence the convergence rate (to close to 4). Despite these caveats on convergence rate, we emphasize that our method is capable of pricing a sophisticated discretely monitored option and obtaining five to six significant digits within a fraction of a second, while at the same time being very easy to understand and to implement. Thus the (relatively technical) issue of convergence rate should pose no real concerns in practice.

**Remark 7.2.** Although a relative error of the order  $10^{-5}$  or  $10^{-6}$  may not always seem necessary for option pricing problems considered in the real financial world, this extra accuracy is actually needed in the calculation of the Greeks, which are typically approximated by finite difference formulas and which are much more sensitive to numerical errors incurred in the calculation of option prices.

## 8. Appendix

8.1. Estimate of Truncation Errors. To estimate the truncation error for the integral in (4.1), we first introduce

Lemma 8.1. Let

$$A = \max\{|a_1^{\pm}|, \dots, |a_M^{\pm}|, |a_M|\}, \qquad B = \max\{|b_1^{\pm}|, \dots, |b_M^{\pm}|, |b_M|\},\$$
$$R = \min\{r_1, \dots, r_M, 0\}, \qquad Q = \min\{q_1, \dots, q_M, 0\}.$$

Then

$$|\tilde{V}_m(S)| \le e^{Q(t_m - t_M)} AS + e^{R(t_m - t_M)} B, \qquad \forall 0 \le m \le M, \ \forall S \in (K_m^-, K_m^+),$$

and

$$|V_m(S)| \le e^{Q(t_m - t_M)} AS + e^{R(t_m - t_M)} B, \qquad \forall 0 \le m \le M, \ \forall S \in (0, \infty).$$

*Proof.* Clearly, by assumption (cf. (3.4)),

$$|\tilde{V}_M(S)| = |V_M(S)| \le AS + B, \quad \forall S \in (0, \infty).$$

Now assume

$$|\tilde{V}_m(S)| \le e^{Q(t_m - t_M)} AS + e^{R(t_m - t_M)} B, \qquad \forall S \in (K_m^-, K_m^+),$$

and

$$|V_m(S)| \le e^{Q(t_m - t_M)} AS + e^{R(t_m - t_M)} B, \qquad \forall S \in (0, \infty),$$

for some  $1 \le m \le M$ . By (3.2) and Proposition 3.4, we have

$$\tilde{V}_{m-1}(S) = e^{-2r_m\tau_m/\sigma_m^2} \int_0^\infty V_m(y)\rho_m(y,S)\,dy,$$

for  $K_{m-1}^- < S < K_{m-1}^+$ . By definition (3.1), it is clear that  $\rho_m(y, S) = \rho_m(y/S, 1)/S$ . Thus a simple change of variable z = y/S yields

$$\begin{split} |\tilde{V}_{m-1}(S)| &= e^{-2r_m\tau_m/\sigma_m^2} \left| \int_0^\infty V_m(Sz)\rho_m(z,1) \, dz \right| \\ &\leq e^{-2r_m\tau_m/\sigma_m^2} \int_0^\infty \left[ e^{Q(t_m-t_M)} ASz + e^{R(t_m-t_M)}B \right] \rho_m(z,1) \, dz \\ &= e^{-2r_m\tau_m/\sigma_m^2} \left( e^{Q(t_m-t_M)} AS \int_0^\infty z\rho_m(z,1) \, dz + e^{R(t_m-t_M)}B \right) \end{split}$$

The integral  $\int_0^\infty z \rho_m(z, 1) dz$  is the expectation of the lognormal distribution, which is simply  $\exp\{2(r_m - q_m)\tau_m/\sigma_m^2\}$  [19]. Thus we have (observe  $R \le 0$  and  $Q \le 0$ )

$$|\tilde{V}_{m-1}(S)| \le e^{-2q_m\tau_m/\sigma_m^2 + Q(t_m - t_M)}AS + e^{-2r_m\tau_m/\sigma_m^2 + R(t_m - t_M)}B$$
$$\le e^{Q(t_{m-1} - t_M)}AS + e^{R(t_{m-1} - t_M)}B,$$

for all  $S \in (K_{m-1}^{-}, K_{m-1}^{+})$ . By (3.3) and Proposition 3.4, we then deduce

$$|V_{m-1}(S)| \le e^{Q(t_{m-1}-t_M)}AS + e^{R(t_{m-1}-t_M)}B,$$

for all  $S \in (0, \infty)$ . The result then follows from induction.

The possibly infinite integral in (3.5) is approximated by a finite integral. To be specific, let C > 1,  $S_0 = S(t_0)$ , and  $\tilde{G}_{M-1}(S) = \tilde{V}_{M-1}(S)$ . We consider  $\tilde{G}_{m-1}$   $(1 \le m \le M-1)$  defined recursively by

$$\begin{split} \hat{G}_{m-1}(S) &= e^{-2r_m\tau_m/\sigma_m^2} \mathbb{E}[\hat{G}_m(\cdot)|S] \\ &= e^{-2r_m\tau_m/\sigma_m^2} \int_{L_m^-}^{L_m^+} \tilde{G}_m(y)\rho_m(y,S) \, dy + a_m^+ V_m^a(S,K_m^+,1) \\ &+ b_m^+ V_m^b(S,K_m^+,1) + a_m^- V_m^a(S,K_m^-,-1) + b_m^- V_m^b(S,K_m^-,-1) \end{split}$$

A direct calculation shows that the errors  $\tilde{R}_m = \tilde{V}_m - \tilde{G}_m$  satisfy the recursion

$$\tilde{R}_{m-1}(S) = e^{-2r_m \tau_m / \sigma_m^2} \left( \int_{L_m^-}^{L_m^+} \tilde{R}_m(y) \rho_m(y, S) \, dy \right. \\ \left. + \int_{K_m^-}^{L_m^-} \tilde{V}_m(y) \rho_m(y, S) \, dy + \int_{L_m^+}^{K_m^+} \tilde{V}_m(y) \rho_m(y, S) \, dy \right) .$$

We use the operator notation

$$\mathcal{T}_m(f)(S) = e^{-2r_m\tau_m/\sigma_m^2} \int_0^\infty f(y)\rho_m(y,S)\,dy,$$

to write the recursion of  $\tilde{R}_m$  as

$$\tilde{R}_{m-1} = \mathcal{T}_m(\tilde{R}_m \chi_{(L_m^-, L_m^+)}) + \mathcal{T}_m(\tilde{V}_m \chi_{(K_m^-, L_m^-] \cup [L_m^+, K_m^+)}),$$

for  $1 \leq m \leq M - 1$ . Note also that  $\tilde{R}_{M-1}(S) = 0$ . Now we consider  $\tilde{Q}_m$  defined by the recursion

(8.1) 
$$\tilde{Q}_{m-1} = \mathcal{T}_m(\tilde{Q}_m) + \mathcal{T}_m((e^{Q(t_0 - t_M)}Ay + e^{R(t_0 - t_M)}B)\chi_{(0,S_0/C] \cup [S_0C,\infty)}),$$

and  $\tilde{Q}_{M-1} = 0$ . It is easy to show using Lemma 8.1 and induction that  $|\tilde{R}_m| \leq \tilde{Q}_m$  for all  $0 \leq m \leq M - 1$ . We can also apply the recursion formula (8.1) to get the expansion

$$\tilde{Q}_0 = \sum_{m=1}^{M-1} \mathcal{T}_1 \circ \dots \circ \mathcal{T}_m \big( (e^{Q(t_0 - t_M)} Ay + e^{R(t_0 - t_M)} B) \chi_{(0, S_0/C] \cup [S_0 C, \infty)} \big).$$

Since  $\mathcal{T}_m$  is the risk-neutral expectation operator discounted from  $t_m$  to  $t_{m-1}$ ,  $\mathcal{T}_1 \circ \cdots \circ \mathcal{T}_m(f)$  is simply the value of a European option whose payoff at  $t_m$  is given by f. Thus  $\tilde{Q}_0$  is equal to the value of a sum of 2(M-1) binary call options with strike  $S_0C$  and 2(M-1) binary put options with strike  $S_0/C$ . As a result,

(8.2) 
$$\tilde{Q}_{0}(S_{0}) = e^{Q(t_{0}-t_{M})}A \sum_{m=1}^{M-1} \left[C_{a}(S_{0}, S_{0}C, t_{m}) + P_{a}(S_{0}, S_{0}/C, t_{m})\right] \\ + e^{R(t_{0}-t_{M})}B \sum_{m=1}^{M-1} \left[C_{b}(S_{0}, S_{0}C, t_{m}) + P_{b}(S_{0}, S_{0}/C, t_{m})\right],$$

where  $C_a, P_a, C_b, P_b$  denote values of asset-or-nothing call, asset-or-nothing put, cash-or-nothing call, and cash-or-nothing put respectively. According to Lemma 3.1, the values of these binary options are given by

$$C_a(S_0, S_0C, t_m) = S_0 e^{-\bar{q}_m(t_m - t_0)} N(d_3), \quad P_a(S_0, S_0/C, t_m) = S_0 e^{-\bar{q}_m(t_m - t_0)} N(d_4),$$
  

$$C_b(S_0, S_0C, t_m) = e^{-\bar{r}_m(t_m - t_0)} N(d_5), \quad P_b(S_0, S_0/C, t_m) = e^{-\bar{r}_m(t_m - t_0)} N(d_6),$$

where

$$d_{3} = \frac{1}{\bar{\sigma}_{m}\sqrt{t_{m}-t_{0}}} \left[ -\log C + (\bar{r}_{m} - \bar{q}_{m} + \frac{1}{2}\bar{\sigma}_{m}^{2})(t_{m} - t_{0}) \right],$$
  

$$d_{4} = \frac{1}{\bar{\sigma}_{m}\sqrt{t_{m}-t_{0}}} \left[ -\log C - (\bar{r}_{m} - \bar{q}_{m} + \frac{1}{2}\bar{\sigma}_{m}^{2})(t_{m} - t_{0}) \right],$$
  

$$d_{5} = \frac{1}{\bar{\sigma}_{m}\sqrt{t_{m}-t_{0}}} \left[ -\log C + (\bar{r}_{m} - \bar{q}_{m} - \frac{1}{2}\bar{\sigma}_{m}^{2})(t_{m} - t_{0}) \right],$$
  

$$d_{6} = \frac{1}{\bar{\sigma}_{m}\sqrt{t_{m}-t_{0}}} \left[ -\log C - (\bar{r}_{m} - \bar{q}_{m} - \frac{1}{2}\bar{\sigma}_{m}^{2})(t_{m} - t_{0}) \right],$$

and  $\bar{r}_m, \bar{q}_m, \bar{\sigma}_m$  represent the time-weighted averages of  $\{r_n, q_n, \sigma_n\}_{n=1}^m$  respectively. Generally, in practice, the absolute values of annual interest and yield rates will not exceed 50%,  $t_M - t_0$  will not exceed 10 years, and A will not exceed 1. For general autocallable structured products, B will not exceed  $t_M - t_0$ , and for Bermudan options B will not exceed K, which is not much larger than  $S_0$ . We may also assume  $M \leq 120$ , which corresponds to products that are not too frequently monitored, say monthly (for more frequently monitored products, such as daily monitored products, continuity correction methods [4, 5] are usually more appropriate). Let  $\sigma_0 = \max_{1 \leq m \leq M} \sigma_m$ . If we choose

$$\log C = 10\sigma_0 \sqrt{t_M - t_0} + \left(1 + \frac{1}{2}\sigma_0^2\right)(t_M - t_0),$$

we can make sure that

$$d_{3,4,5,6} \le -10$$
, and thus  $N(d_{3,4,5,6}) < 10^{-23}$ .

20

A crude estimate using (8.2) then shows that the error bound  $\tilde{Q}_0(S_0)$  does not exceed  $10^{-15}(S_0 + 1)$ . This means the relative truncation error is negligible for all practical purposes.

8.2. Analysis of  $K_m^{\pm}$  for Bermudan Options. The proper application of Proposition 3.4 requires the uniqueness of the exercise prices  $K_m^{\pm}$ , which we now establish for Bermudan options.

To begin with, observe that the risk-neutral pricing formulas (3.2)–(3.3) applied to Bermudan options can be written in an alternative form as

(8.3) 
$$V_{m-1}(S) = \max\{V_{m-1}(S), \epsilon(S-K)\}, \quad \forall 1 \le m \le M, \ \forall S \in (0,\infty),$$
  
where 
$$\begin{cases} 1 & \text{if the option is a Bermudan call} \end{cases}$$

$$\epsilon = \begin{cases} 1, & \text{if the option is a Bermudan can} \\ -1, & \text{if the option is a Bermudan put} \end{cases},$$

and

(8.4) 
$$\tilde{V}_{m-1}(S) = e^{-2r_m\tau_m/\sigma_m^2} \int_0^\infty V_m(y)\rho_m(y,S) \, dy$$
$$= e^{-2r_m\tau_m/\sigma_m^2} \int_0^\infty V_m(Sz)\rho_m(z,1) \, dz.$$

Since, by definition,

$$V_M(S) = \max\{0, \epsilon(S-K)\},\$$

(8.3) and (8.4) define  $V_m$  and  $\tilde{V}_m$  recursively for all  $0 \le m \le M - 1$ . It is easy to see that  $V_m(S) \ge \tilde{V}_m(S) > 0$  for all  $0 \le m \le M - 1$  and  $S \in (0, \infty)$ .

**Proposition 8.2.** Assume  $q_m \ge 0$  for all  $1 \le m \le M$ . For a Bermudan call option with strike K, the equation  $\tilde{V}_m(K_m^+) = K_m^+ - K$  has at most one finite solution, and

$$0 < \tilde{V}_m(S_2) - \tilde{V}_m(S_1) < S_2 - S_1, \qquad \forall 1 \le m \le M - 1, \ \forall S_2 > S_1 > 0.$$

For a Bermudan put option with strike K, the equation  $\tilde{V}_m(K_m^-) = K - K_m^-$  has at most one finite solution, and

$$0 < \tilde{V}_m(S_1) - \tilde{V}_m(S_2) < S_2 - S_1, \qquad \forall 1 \le m \le M - 1, \ \forall S_2 > S_1 > 0.$$

*Proof.* Clearly Q = 0 since  $q_m \ge 0$ . The proofs for call and put options are very similar, and we will present the argument for put options only which proceeds by induction. Since  $\tilde{V}_{M-1}$  is the value of a European vanilla put option and  $q_m \ge 0$ , its delta is between -1 and 0 [18]. Thus

$$0 < \tilde{V}_{M-1}(S_1) - \tilde{V}_{M-1}(S_2) = -\int_{S_1}^{S_2} \tilde{V}'_{M-1}(S) \, dS < S_2 - S_1,$$

for all  $S_2 > S_1 > 0$ . Now suppose  $\tilde{V}_{M-1}(S_3) = K - S_3$  and  $\tilde{V}_{M-1}(S_4) = K - S_4$ for some  $S_4 > S_3 > 0$ . This means

$$S_4 - S_3 = V_{M-1}(S_3) - V_{M-1}(S_4) < S_4 - S_3,$$

which is a contradiction. So the proposition is true for m = M - 1.

Suppose now the results hold for some  $2 \le m \le M - 1$ , which implies that the function  $S - K + \tilde{V}_m(S)$  is strictly increasing. Since  $\tilde{V}_m(S) > 0$ , for sufficiently large S we have  $S - K + \tilde{V}_m(S) > 0$ , or  $\tilde{V}_m(S) > K - S$ . This means that

$$K_m^- = \inf\{S > 0 \colon V_m(S) > K - S\}$$

is well-defined and satisfies  $K_m^- < \infty.$  Consider now (cf. (8.3))

(8.5) 
$$V_m(S) = \begin{cases} \tilde{V}_m(S), & S \ge K_m^-, \\ K - S, & S < K_m^-, \end{cases}$$

and any  $S_2 > S_1 > 0$ . If  $S_2 < K_m^-$  (and hence  $S_1 < K_m^-$ ), we have

$$V_m(S_1) - V_m(S_2) = (K - S_1) - (K - S_2) = S_2 - S_1,$$

by (8.5). If  $S_1 \ge K_m^-$  (and hence  $S_2 \ge K_m^-$ ), we have

V

$$\tilde{V}_m(S_1) - V_m(S_2) = \tilde{V}_m(S_1) - \tilde{V}_m(S_2) \in (0, S_2 - S_1),$$

by (8.5) and inductive hypothesis. If  $S_1 < K_m^- \leq S_2$ , we have

$$V_m(S_1) - V_m(S_2) = (K - S_1) - V_m(S_2)$$
  
<  $(K - S_1) - (K - S_2) = S_2 - S_1,$   
 $V_m(S_1) - V_m(S_2) > (K - S_1) - \tilde{V}_m(S_1) \ge 0,$ 

by (8.5), inductive hypothesis, and the definition of  $K_m^-$ . In conclusion, we have shown that

$$\begin{array}{ll} (8.6a) & 0 < V_m(S_1) - V_m(S_2) = S_2 - S_1, & \forall K_m^- > S_2 > S_1 > 0, \\ (8.6b) & 0 < V_m(S_1) - V_m(S_2) < S_2 - S_1, & \text{otherwise.} \end{array}$$

8.6b) 
$$0 < V_m(S_1) - V_m(S_2) < S_2 - S_1$$
, otherwise.

With the aid of (8.4) and (8.6), we write

$$\tilde{V}_{m-1}(S_1) - \tilde{V}_{m-1}(S_2) = e^{-2r_m \tau_m / \sigma_m^2} (I_1 + I_2),$$

where

$$0 < I_{1} = \int_{0}^{K_{m}^{-}/S_{2}} \left[ V_{m}(S_{1}z) - V_{m}(S_{2}z) \right] \rho_{m}(z,1) dz$$
  
$$= (S_{2} - S_{1}) \int_{0}^{K_{m}^{-}/S_{2}} z \rho_{m}(z,1) dz,$$
  
$$0 < I_{2} = \int_{K_{m}^{-}/S_{2}}^{\infty} \left[ V_{m}(S_{1}z) - V_{m}(S_{2}z) \right] \rho_{m}(z,1) dz$$
  
$$< (S_{2} - S_{1}) \int_{K_{m}^{-}/S_{2}}^{\infty} z \rho_{m}(z,1) dz.$$

As a result,

(8.7) 
$$0 < \tilde{V}_{m-1}(S_1) - \tilde{V}_{m-1}(S_2)$$
$$< e^{-2r_m \tau_m / \sigma_m^2} (S_2 - S_1) \int_0^\infty z \rho_m(z, 1) \, dz \le S_2 - S_1,$$

since by elementary properties of lognormal distributions [19],

$$e^{-2r_m\tau_m/\sigma_m^2} \int_0^\infty z\rho_m(z,1) \, dz = e^{-2q_m\tau_m/\sigma_m^2} \le 1.$$

Now suppose  $\tilde{V}_{m-1}(S_3) = K - S_3$  and  $\tilde{V}_{m-1}(S_4) = K - S_4$ . This means  $\tilde{V}_{m-1}(S_3) - \tilde{V}_{m-1}(S_4) = S_4 - S_3,$ 

so by (8.7) we must have  $S_3 = S_4$ . The proposition then follows from induction.  $\Box$ 

**Corollary 8.3.** Assume a Bermudan put option with strike K has an optimal earlyexercise level  $K_m^- > 0$  for some  $1 \le m \le M - 1$ . Then we have  $\tilde{V}_m(S) > K - S$  for  $S > K_m^-$  and  $\tilde{V}_m(S) < K - S$  for  $S < K_m^-$ . Similarly, for a Bermudan call option we have  $\tilde{V}_m(S) < S - K$  for  $S > K_m^+$  and  $\tilde{V}_m(S) > S - K$  for  $S < K_m^+$ .

*Proof.* We will present the proof for put options only as the argument for call options is similar. It follows from Proposition 8.2 that the function  $\tilde{V}_m(S) + S - K$  is increasing in S. Since  $\tilde{V}_m(K_m^-) + K_m^- - K = 0$ , we have  $\tilde{V}_m(S) > K - S$  for  $S > K_m^-$  and  $\tilde{V}_m(S) < K - S$  for  $S < K_m^-$ .

#### Acknowledgements

We are very grateful to Qingshuo Song for his valuable insights and helpful suggestions, as well as to Zhenan Sui for her careful reading and editing of the manuscript.

#### References

- D.-H. Ahn, S. Figlewski, and B. Gao, Pricing discrete barrier options with an adaptive mesh model, J. Deriv. 6(4), 33–43 (1999).
- [2] T. Alm, B. Harrach, D. Harrach, and M. Keller, A Monte Carlo pricing algorithm for autocallables that allows for stable differentiation, J. Comput. Financ. 17(1), 43–70 (2013).
- [3] A. D. Andricopoulos, M. Widdicks, P. W. Duck, and D. P. Newton, Universal option valuation using quadrature methods, J. Financ. Econ. 67(3), 447–471 (2003).
- M. Broadie, P. Glasserman, and S. Kou, A continuity correction for discrete barrier options, Math. Financ. 7(4), 325–348 (1997).
- [5] M. Broadie, P. Glasserman, and S. G. Kou, Connecting discrete and continuous pathdependent options, Financ. Stoch. 3(1), 55–82 (1999).
- [6] M. Broadie and Y. Yamamoto, A double-exponential fast Gauss transform algorithm for pricing discrete path-dependent options, Oper. Res. 53(5), 764–779 (2005).
- [7] P. Buchen and O. Konstandatos, A new approach to pricing double-barrier options with arbitrary payoffs and exponential boundaries, Appl. Math. Financ. 16(6), 497–515 (2009).
- [8] R. L. Burden, J. D. Faires, and A. M. Burden, Numerical Analysis (10th Edition), Brooks Cole (2015).
- [9] G. Deng, J. Mallett, and C. McCann, Modeling autocallable structured products, J. Deriv. Hedge Funds 17(4), 326–340 (2011).
- [10] F. Fang and C. W. Oosterlee, Pricing early-exercise and discrete barrier options by Fouriercosine series expansions, Numer. Math. 114, 27–62 (2009).
- [11] L. Feng and X. Lin, Pricing Bermudan options in Lévy process models, SIAM J. Finan. Math. 4(1), 474–493 (2013).
- [12] L. Feng and V. Linetsky, Pricing discretely monitored barrier options and defaultable bonds in Lévy process models: a fast Hilbert transform approach, Math. Financ. 18(3), 337–384 (2008).
- [13] C. P. Fries and M. S. Joshi, Perturbation stable conditional analytic Monte-Carlo pricing scheme for auto-callable products, Int. J. Theor. Appl. Finance 14(2), 197–219 (2011).
- [14] G. Fusai, I. D. Abrahams, and C. Sgarra, An exact analytical solution for discrete barrier options, Finance Stochast. 10(1), 1–26 (2006).
- [15] J. Gatheral, The Volatility Surface: A Practitioner's Guide, John Wiley & Sons (2006).
- [16] A. Golbabai, L. V. Ballestra, and D. Ahmadian, A highly accurate finite element method to price discrete double barrier options, Comput. Econ. 44(2), 153–173 (2014).
- [17] A. Ibáñez and C. Velasco, The optimal method for pricing Bermudan options by simulation, Math. Financ. 28(4), 1143–1180 (2018).
- [18] J. C. Hull, Options, Futures, and Other Derivatives (9th Edition), Pearson (2014).
- [19] N. L. Johnson, S. Kotz, and N. Balakrishnan, "14: Lognormal Distributions", Continuous Univariate Distributions (Volume 1, 2nd Edition), Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons (1994).

- [20] C. F. Lo, H. C. Lee, and C. H. Hui, A simple approach for pricing barrier options with time-dependent parameters, Quant. Financ. 3(2), 98–107 (2003).
- [21] R. Lord, F. Fang, F. Bervoets, and C. W. Oosterlee, A fast and accurate FFT-based method for pricing early-exercise options under Lévy processes, SIAM J. Sci. Comput. 30(4), 1678–1705 (2008).
- [22] C. O'Sullivan, Path dependant option pricing under Lévy processes, EFA 2005 Moscow Meetings Paper (2005).
- [23] E. Reiner, Convolution methods for path-dependent options, Financial Mathematics: Risk Management, Modeling and Numerical Methods, IPAM UCLA (Jan. 3–12, 2001).
- [24] S. E. Shreve, Stochastic Calculus for Finance II: Continuous-Time Models, Springer-Verlag (2004).
- [25] T. Guillaume, Autocallable structured products, J. Deriv. 22(3), 73–94 (2015).
- [26] T. Guillaume, Analytical valuation of autocallable notes, Int. J. Financ. Eng. 2(2), 1–23 (2015).
- [27] J. Z. Wei, Valuation of discrete barrier options by interpolations, J. Deriv. 6(1), 51-73 (1998).

#### MIN HUANG

China Merchants Bank 7088 Shennan Boulevard, Shenzhen, Guangdong, China Email: huang.479@osu.edu

#### Guo Luo

Department of Mathematics, City University of Hong Kong Tat Chee Avenue, Kowloon, Hong Kong Email: guoluo@cityu.edu.hk