A General Control Variate Method

Chun-Yuan Chiu* Tian-Shyr Dai† Hua-Yi Lin ‡

April 9, 2013

Abstract

The control variate method is a popular variance reduction technique used in Monte Carlo methods, which are frequently used to price complex derivatives. This technique is used in much financial literature to exploit information about the errors in estimates of known quantities – which are usually values of the derivatives that can be analytically priced – to reduce the error for estimating an unknown quantity, which is usually the price of a complex derivative of interest. This paper generalizes the core idea of the control variate method so that it can be applied to reduce the pricing errors incurred in many numerical pricing methods, such as the tree method, the characteristic-function-based pricing method, and the convolution-based pricing method. For example, a numerical method for pricing a complex derivative, say an Asian option, may need to calculate the density function of the average price of the underlying asset in the convolution-based pricing method. However, this function can not be analytically solved and must be numerically approximated. Thus we find another analytical function that is close to the function of interest and exploit information about the errors in estimates of the analytical function to reduce the error for estimating the function of interest. Numerical results shows that our approach can effectively increase the pricing efficiency of many numerical pricing methods.

JEL classification: C02; G13

Keywords: numerical method, pricing, control variate, tree, fast Fourier transform

*Corresponding author. Department of Mathematics, Florida State University, 208 Love Building, 1017 Academic Way, Tallahassee. E-mail: cchiu@math.fsu.edu.

†Institute of Finance, National Chiao-Tung University, 1001 Ta Hsueh Road, Hsinchu 30010, Taiwan. E-mail: cameldai@mail.nctu.edu.tw.

‡Institute of Finance, National Chiao-Tung University, 1001 Ta Hsueh Road, Hsinchu 30010, Taiwan. E-mail: linhuayi@g2.nctu.edu.tw.

1 Introduction

Due to the booming of financial markets and related academic studies, complex derivative products, like Asian options, are constructed to meet customers' need, and sophisticated pricing models, like the Lévy model, are applied to fit financial markets better. Although the presences of these complex derivatives and pricing models improve the efficiency of financial markets, they also make the derivation of analytical pricing formulae intractable. Thus we must rely on many different numerical methods, such as Monte Carlo simulations, tree methods, characteristic-function-based pricing methods, convolution-based methods, and so on.

In most numerical methods, continuous pricing models or equations are transferred into discrete counterparts that are suitable for numerical evaluations. Take the CRR tree model proposed by Cox et al. (1979) in Fig. 1 as an example. The time interval $[0, T]$ are discretized into several time steps, each with the length Δt. The outgoing binomial structure of each node also discretize the price of the underlying asset at each time step, say Su and Sd at the time step 1. The details of this figure will be discussed in later sections. Usually, the pricing results generated by a numerical method converge to the theoretical derivative price as the discretization get finer; for example, the pricing results generated by the CRR tree converge to the theoretical derivative price as the number of time steps increases (see Duffie (1996)). However, increasing the resolution of the discretization would (dramatically) increases the computational time complexity; in other words, the trade-off between the pricing accuracy and the computational time is a big challenge for using numerical methods.

Improving the performance of the numerical pricing methods is an important issue and is widely studied in academic literature. One possible approach is to improve the efficiency for computing numerical methods. For example, evaluating derivatives under the tree model illustrated in Fig. 1 is traditionally performed by the backward induction method. Roughly speaking, we first evaluate the derivative price at each node located in the last time step, say time step 2 in this example. This information is then used to derive the derivative price at each node in time step 1. The above procedure is repeatedly applied by using the information at time step i to derive the information at time step $i - 1$ until the derivative price at time step 0 is obtained. Lyuu (1998) and Dai et al. (2007) suggest that some calculation process of the backward induction method can be saved by taking advantage of combinatorial properties. Thus the pricing performance is improved since the combinatorial method requires less computational time to achieve the same level of accuracy than the backward induction method does.

Another approach to improve a numerical method is to improve the convergence rate of the pricing results. Take the Monte Carlo pricing method (`MC` hereafter) pioneered by Boyle (1977) as an example. It randomly samples the changes of economic variables, say the underlying asset's price, to evaluate a derivative's price, which is equal to the discounted expected payoff of that derivative under the so-called risk-neutral probability measure (see Harrison and Pliska (1981)). Note that the pricing result is also a random variable and a large amounts of samples (and hence a large amount of computational time) are required to obtain a satisfactory pricing result. The control variate method is one of the important approaches that improve the performance of the `MC` by reducing the variances of the pricing results. Instead of directly applying the `MC` to estimate an unknown quantity which is usually the price of a complex derivative of interest, the control variate method use `MC` to estimate the expected difference between the unknown quantity and the known ones which are usually the values of derivatives that can be analytically solved. Thus the unknown quantity can be estimated as the known quantity (evaluated by the analytical formula) plus the expected difference. The variance of the pricing result can be dramatically reduced if the sample differences are close to zero. This approach is widely adopted in the following financial literature. For example, Kemna and Vorst (1990) take advantage of the analytical pricing formulae for geometric Asian options to evaluate arithmetic Asian options. It is further improved by Han and Lai (2010) by using an optimal control process as a control

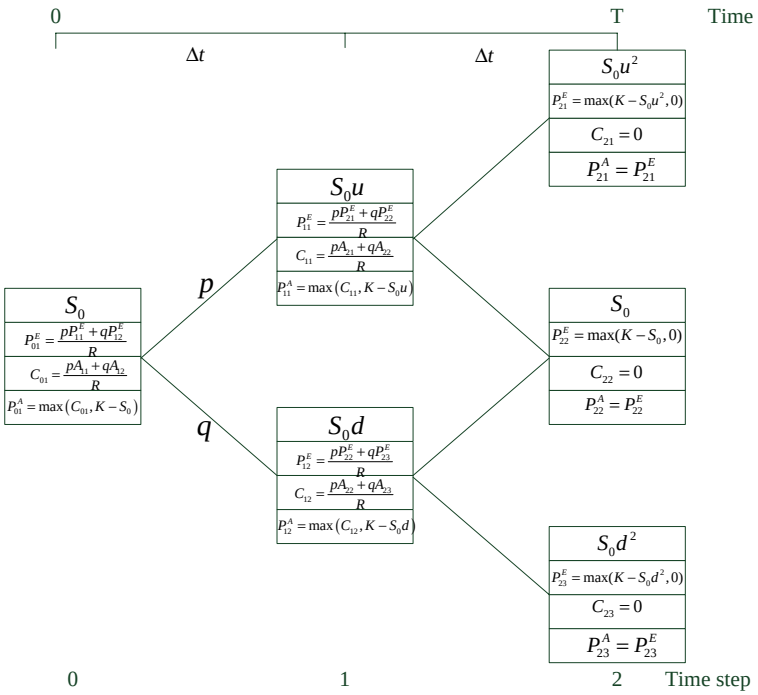


Figure 1: **Pricing American Puts on the Two-time-Step CRR Tree** S_0 denotes the underlying asset's value at time 0, K denotes the strike price, T denotes the time to maturity for the American put, Δt denotes the length of a time step, and R denotes the discount factor. p and q denote the upward and the downward branching probabilities, respectively. u and d denote the upward and the downward multiplication factors, respectively. Each node is represented by a node with four fields, which denote the price of the underlying asset, the price of the European put $\hat{P}_{ij}^E$, the continuation value $\hat{V}_{ij}$, and the price of the American put $\hat{P}_{ij}^A$ of that node. Overhead hat symbols are used to distinguish symbols involved in the traditional backward induction method in this figure from the symbols used in the general control variate method (discussed later). The subscript ij means the j-th node (counted from the uppermost node) at the i-th time step.

variate to gain further variance reduction. Besides, Duan (1995) also applies the control variate technique to efficiently evaluate options under the GARCH model. Dingec and Hormann (2012) price path dependent options under Lévy processes by exploiting information about the error to evaluate the same options under geometric Brownian motion.

Hull and White (1988) generalize the core idea of the control variate method by applying it to price an American put on the CRR tree as illustrated in Fig. 1. Specifically, they take advantage of the error for estimating the price of a European put on a CRR tree to reduce the error for estimating the price of an American put on the same tree. They suggest that the American put can be evaluated as the difference between the estimates of prices of American and European puts (i.e, $\hat{P}_{01}^A - \hat{P}_{01}^E$)[1] plus the European put price which can be analytically solved by Black and Scholes (1973) formula.

This paper will investigate the property of the control variate method further and examine how useful it could be to improve the performance of various numerical pricing methods. Roughly speaking, most numerical pricing methods would evaluate the result for applying an operation O on a certain function X. Usually $O(X)$ can not be analytically evaluated and must be estimated numerically. Our core idea is to suppress the error for estimating $O(X)$ in order to improve the performance of numerical pricing methods. To inherit the spirit of the control variate, we first pick another function Y as a control variate – call it the proxy function for simplicity. That is,

[1]The pricing results generated by the CRR tree for European and American puts are $\hat{P}_{01}^E$ and $\hat{P}_{01}^A$, respectively.

the proxy function Y is close to the function X and $O(Y)$ can be analytically solved. Then a more accurate estimation for $O(X)$ is obtained by numerically estimating $O(X-Y)$ plus the analytical result of $O(Y)$. Take pricing American puts with the CRR tree illustrated in Fig. 1 as an example. The American-put-value function of the underlying asset price at the time step i plays the role of the function X. Finding the continuation-value[2] function (of the underlying asset price) at the time step $i-1$ based on the function X plays the role of the operation O. Operation O can be numerically implemented by a 1-time-step backward induction; in other words, the continuation value of a node is estimated as the expected discounted option values of its two successor nodes as illustrated in the third field of that node. The European-put-value function at time step i plays the role of proxy function Y to suppress the error for numerically estimating $O(X)$. This is because the function Y is close to X and each point in the European-put-value function at time step $i-1$ (or $O(Y)$) can be analytically solved by the Black and Scholes (1973) formula. Note that the American put value for each node at time step $i-1$ is then evaluated as the maximum of the continuation value and the exercise value at that node as illustrated in the fourth field of that node. And the aforementioned evaluations are repeated from the last time step to the beginning of the tree to obtain the final pricing result. Note that our approach can repeatedly apply the control variate method, one for each time step, while Hull and White (1988) approach apply once to overall pricing result. Numerical experiments suggest that our approach outperforms their approach.

Two more examples are given to demonstrate how the general control variate method is applied to improve the performance of numerical pricing methods. First, to capture versatile behaviors of the underlying asset price, various processes, such as the jump diffusion process proposed by Merton (1976), the stochastic volatility process proposed by Heston (1993), and so on, are invented to fit the real world market better. Since the characteristic functions of most of these processes are simpler to derive analytically than the probability density function of these processes, Carr and Madan (1999) and Carr and Wu (2004) take advantage of the relation between the Fourier transforms of option prices and the characteristic functions to price options. Specifically, the option prices can be evaluated as an inverse Fourier transform of the function that can be expressed in terms of the characteristic function, and this transform can be numerically evaluated by the fast Fourier transform (FFT)– which can be viewed as a numerical integration as discussed later. However, the integrand oscillates significantly, which make a high resolution of discretization (and hence a high running time complexity) necessary to obtain a satisfactory pricing result. To improve the pricing performance, we first pick a proxy function; that is, this proxy function is close to the characteristic function and the inverse Fourier transform of the proxy function can be analytically solved. Then we can exploit information of the error for applying FFT on this proxy function to reduce the pricing error under our general control variate framework. Numerical results suggest that our revised method provide more accurate pricing results and can avoid the negative-price problem for pricing deep-out-of money options.

Next, the general control variate method is used to improve the performance of the convolution-based Asian option pricing method. The payoff of an Asian option depends on the average price of the underlying asset. The pricing problem is intractable since the density function of the average asset price f_A can not be analytically solved. Carverhill and Clewlow (1990) and Benhamou (2002) suggest that f_A can be numerically evaluated by alternatively applying convolutions and the Jacobian transformation[3] on density functions. Numerical errors propagate and accumulate when these operations are alternatively applied numerically. To reduce accumulated numerical errors, we select proxy functions that are close to the inputs to these operations. In addition,

[2]The continuation value means the value to hold an American put without exercising it immediately.

[3]It is used to implement the transformation of random variables; that is, to find the probability density of one random variable that is a function another random variable.

applying the operations on the proxy functions must be analytically calculated. Thus, the information of the error for applying operations numerically on proxy functions are applied to suppress accumulation errors and hence the errors of pricing results.

The rest of this paper is organized as follows. In Sec. 2, we will survey the traditional control variate method and introduce the framework of our general version. Then we demonstrate how general control variate method reduce the pricing error of the the tree pricing method, the characteristic-function-based pricing method, and the convolution-based pricing method in Sec. 3, 4, and 5, respectively. A simple survey of each numerical method and the related numerical experiments will also be given in the corresponding section. Sec. 6 concludes the paper.

2 The Control Variate Method

2.1 The Traditional Version

The traditional control variate method is a variance reduction technique used in `MC` (see Glasserman, 2004). It uses the estimation error of known quantities, say $\mathrm{E}(Y)$, to improve the estimation of unknown quantities, say $\mathrm{E}(X)$, where X and Y are random variables. Then the estimation of $\mathrm{E}(X)$ can be improved by estimating $\mathrm{E}(X-Y)$ with `MC` plus the analytical value of $\mathrm{E}(Y)$ instead of directly estimating $\mathrm{E}(X)$ with `MC`. Note that the variances for former and the latter estimation are $\mathrm{Var}(X)+\mathrm{Var}(Y)-2\mathrm{Cov}(X,Y)$ and $\mathrm{Var}(X)$, respectively. This implies that the control variance technique can dramatically reduce the variance if we can find a proper Y that is highly correlated to X. Note that the derivative price is equal to the expected present value of its future payoff under the risk neutral probability measure. Thus much literature take advantage of the traditional control technique to evaluate the value of a complex derivative with a simple derivative that can be analytically priced. For example, Hull and White (1988) price an American put by using a European put's payoff as the control variate. Kemna and Vorst (1990) price an arithmetic Asian option by using a geometric Asian option's payoff as the control variate. Both approaches dramatically improve pricing performance since the payoff of the latter (simple) derivative is highly correlated to the former (complex) derivative.

2.2 The General Version

Recall that evaluating the expected value of a random variable Z can also be expressed as the integral of the random variable Z multiplied by its density function. Thus we can extend the control variate technique from `MC` to the numerical integration described as follows. Note that random variables X and Y can be treated as real-valued functions defined on the sample space Ω.Thus the expected values $\mathrm{E}(X)$ and $\mathrm{E}(Y)$ can be expressed in terms of integrations $\int_\Omega X(\omega)dP(\omega)$ and $\int_\Omega Y(\omega)dP(\omega)$, respectively, where f denotes the density function. If $\mathrm{E}(X)$ can not be analytically solved, we can estimate it by numerical integration instead of `MC`. To suppress the error incurred by numerical integration, we can mimic the strategies used in the traditional control variate method – picking a random variable Y that is highly correlated to X and whose expected value can be analytically solved. Analogously, we pick a function Y that is close to X. The integration $\int_\Omega Y(\omega)dP(\omega)$ can be analytically solved. Thus the estimation error of $\mathrm{E}(X)$ can be reduced by estimating $\mathrm{E}(X-Y)$ with numerical integration plus the analytical value of $\mathrm{E}(Y)$ instead of directly estimating $\mathrm{E}(X)$ with numerical integration.

Many numerical pricing methods are designed to numerically evaluate the result of applying an operation O (instead of simply calculating the expected value) of a function X. The pricing accuracy can be improved by suppressing the error for estimating $O(X)$, and our paper will achieve this goal by generalizing the aforementioned control variate method. Each numerical pricing method would have its own operation and the function that play the roles of O and

X. For example, in a characteristic-function-based pricing method like Carr and Madan (1999) and Carr and Wu (2004), the inverse Fourier transform and the characteristic function of the underlying asset's process play the roles of O and X, respectively. A numerical method may numerically evaluate more than one operations that can be improved by the general control variate method. In a convolution-based pricing method like Carverhill and Clewlow (1990) and Benhamou (2002), the convolution and the Jacobian transformation are alternatively applied repeatedly. The general control variate method is applied on each operation to improve the accuracy of an intermediate output rather than directly on the pricing result. In addition, the intermediate output is not necessary a value; it can be a discretized function, say a discretized density function in the convolution-based pricing method. Our numerical experiments suggest that the accumulations of these "partial" improvements would result in overall performance improvement of a numerical pricing method in terms of accuracy and computational time.

Applying the control variate method to reduce the error for numerically estimating $O(X)$ is similar to the procedure mentioned above. First, we find a proper proxy function Y that makes $O(Y)$ analytically solvable and that is close to X. Then the error for estimating $O(X)$ is reduced by estimating $O(X-Y)$ numerically plus the analytical value of $O(Y)$. Note that the effectiveness of the general control variate method highly depends on how the proxy function Y is close to the function X. For pricing American puts on the tree, we use the European-put-price function of the underlying asset as the proxy function since these two options are almost identical expect the right to exercise the option early. For the characteristic-function-based pricing method, we use the characteristic function of the jump diffusion process proposed in Merton (1976) to approximate the characteristic function of the underlying asset's price process. The parameters of the former process is calibrated to make the first five cumulants of the former process match the cumulants of the latter process. For the convolution-based pricing method, we use the normal density function to approximate the density function inputted to the convolution or the Jacobian transformation. Again, the parameters of the former density function are calibrated to make the first two moments of the former function match the estimated moments of the latter one.

3 Pricing American Puts under Tree Methods

3.1 Background Introduction

A put option grants a holder the right to sell the underlying asset for a predetermined strike price K. While a European put only allows the option holder to exercise the right at the maturity date T, an American put allows the holder to exercise the right at any time τ prior to maturity T. Let S_t denote the underlying asset price process at time t for convenience. The payoff of a European put at maturity is $(K - S_T)^+$, and the payoff to exercise an American put at time τ is $(K - S_\tau)$. By taking advantage of the risk neutral valuation method, the value of a European put P^E and an American put P^A can be expressed as the expected discounted payoff

$$P^E = \mathrm{E}_Q\left[e^{-rT}(K - S_T)^+\right] \quad (1)$$

$$P^A = \max_{\tau\in\Upsilon} \mathrm{E}_Q\left[e^{-r\tau}(K - S_\tau)\right], \quad (2)$$

where r denotes the risk-free rate, Q denotes the risk neutral measure, and Υ denotes the set of all stopping times. Here we follow Hull and White (1988) by assuming that S_t follow the lognormal diffusion process

$$\ln(S_t/S_0) = \left(r - \sigma^2/2\right) t + \sigma W_t, \quad (3)$$

where σ denotes the volatility of the underlying asset's price, and W_t denotes a standard Brownian motion. Note that Eq. (1) can be analytically evaluated by the Black and Scholes (1973)

formula, while Eq. (2) can not be analytically evaluated due to the difficulty to estimate the early exercise premium.

The evolution of a lognormal diffusion process in Eq. (3) can be discretely simulated by a tree, say the CRR tree (see Cox et al., 1979), as illustrated in Fig. 1. It divides the time interval $[0, T]$ into several equal time steps, each with length Δt. For convenience, the number xy in the subscript of $\hat{P}^E$ and $\hat{P}^A$ denote the values of the European put and the American put at node (i,j) – the j-th node (counted from the uppermost node) at the i-th time step. The underlying asset price S at an arbitrary node (i,j) would move to Su (node $(i+1,j)$) with probability p or Sd (node $(i+1,j+1)$) with probability q at the next time step, where multiplication factors u and d are set to $e^{\sigma\sqrt{\Delta t}}$ and $1/u$, respectively, and branching probabilities p and q are set to $\frac{e^{r\Delta t}-d}{u-d}$ and $1-p$, respectively, to match the first two moments of the lognormal diffusion process. Both European and American puts can be numerically priced on a CRR tree by the backward induction method; that is, we evaluate the option values (of the nodes) from the end of the tree to its beginning. Specifically, the value of a European put for an arbitrary node prior to maturity, say $\hat{P}^E_{ij}$, is evaluated as the expected discounted value of its successor node $\hat{P}^E_{ij} = \frac{p\hat{P}^E_{i+1,j}+q\hat{P}^E_{i+1,j+1}}{R}$, where R denotes the 1-time-step discount factor $e^{-r\Delta t}$. An American put grants holder the right to exercise the put at node (i,j) for the profit $K - S(i,j)$, where $S(i,j)$ denotes the asset value at that node. The holder can also choose not to exercise the option at node (i,j) immediately and the value of the option, called the continuation value for simplicity, can be estimated by

$$\hat{V}_{ij} = \frac{p\hat{P}^A_{i+1,j} + q\hat{P}^A_{i+1,j+1}}{R}. \tag{4}$$

The holder will decide whether he/she exercises the option or not to maximize the benefit; that is, the value of American put at node (i,j) is

$$\max\left(\hat{V}_{ij}, K - S(i,j)\right). \tag{5}$$

The pricing results for European and American puts are $\hat{P}^E_{01}$ and $\hat{P}^A_{01}$, respectively. The accuracy of the pricing results can be improved as the discretization of the tree get finer.

3.2 Applying the General Control Variate Method

The operation that calculates the expected discounted payoff as in Eqs. (1) and (2) can be numerically estimated by the tree method. Hull and White (1988) extend the control variate method by applying it on the tree method instead of `MC`. They take advantage of the error for estimating P^E by the tree method to reduce the error for estimating P^A. The estimation result $\hat{P}^A_{01} - \hat{P}^E_{01} + \mathrm{BS}(S_0, T)$ is much more accurate than $\hat{P}^A_{01}$ given the same discretization level (of the tree), where $\mathrm{BS}(a,b)$ denotes the Black and Scholes (1973) put option pricing formula with the underlying asset price a and the time to maturity b.

We generalize the control variate method further by applying it to improve the accuracy of certain operations in the tree method. Here we detail how the continuous pricing model in Eq. (2) is transferred into its discrete counterpart since similar concepts and notations will be used in the following sections. Recall that a n-time-step tree method discretizes the time interval $[0, T]$ into n equal-distance time steps. Thus, pricing the American put on the tree can be viewed to repeat the operation O_B, to evaluate the continuation value function of the underlying asset's value at time step $i-1$ based on the option-value function at time step i, n times from the last step back to the beginning of the tree. Take Fig. 1 as an example. Let P^A_i, V_i, and P^E_i denote the American put value function, continuation value function, and the European put value functions of the underlying asset at time step i, respectively. P^A_2 can be discretely approximated by keeping the option values at certain points, say S_0u^2, S_0, and S_0d^2, in a list $\hat{\mathbb{P}}^{\mathbb{A}}_2$; that is,

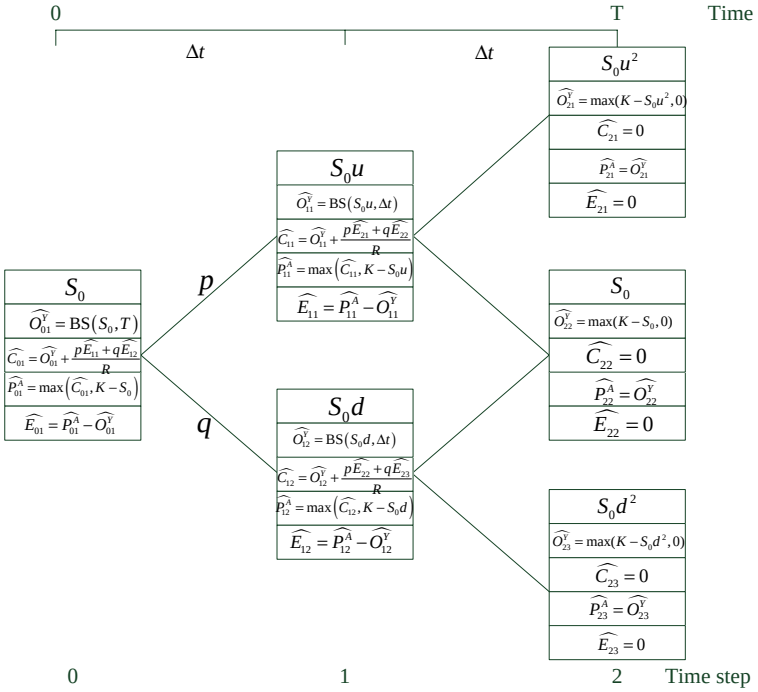


Figure 2: **Pricing American Puts with the General Control Variate Method** The meanings of S_0, u, d, p, q, and K are the same as those in Fig. 1. Each node (i,j) is represented by a rectangle with five fields, which denote the price of the underlying asset, the analytical value of the European put P_{ij}^E, the continuation value V_{ij}, the value of the American put P_{ij}^A, and the early exercise premium E_{ij}.

$\hat{\mathbb{P}}_2^{\mathbb{A}} \equiv \{\hat{P}_{21}^A, \hat{P}_{22}^A, \hat{P}_{23}^A\}$. The operation O_B is numerically evaluated by applying the 1-time-step backward induction formula like Eq. (4) on $\hat{\mathbb{P}}_2^{\mathbb{A}}$ to obtain the discretization of the continuation value function at time step 1, $\hat{\mathbb{V}}_1 \equiv \{\hat{V}_{11}, \hat{V}_{12}\}$. The above numerical estimation can be expressed as $\hat{\mathbb{V}}_1 = O^N(\hat{\mathbb{P}}_2^{\mathbb{A}})$, where O_B^N denotes that the operation O_B is evaluated numerically. Then the list of the American put values at time step 1, $\hat{\mathbb{P}}_1^{\mathbb{A}}$ is obtained by judging whether exercising the put is beneficial or not by Eq. (5) at each node in the time step 1. O_B^N and the early exercise judgement can be alternatively applied to obtain the final pricing result $\hat{P}_{01}^A$.

Now we improve the accuracy for estimating the continuation value function and as a consequence the American put value function by the general control variate method as illustrated in Fig. 2. The American and the European-put-value functions play the roles of the function of interest X and the proxy function Y, respectively. Unlike Eq. (4), the continuation value for a node (i,j) prior to maturity is now estimated as $V_{ij} \equiv \frac{pE_{i+1,j} + qE_{i+1,j+1}}{R} + P_{ij}^E$, where $E_{i,j}$ denotes the early exercise premium, or the difference of the values of the American and the European puts at node (i,j). P_{ij}^E denotes the European put value at node (i,j) that is analytically evaluated by the Black and Scholes (1973) formula[4]. The above formula for evaluating V_{ij} is applied to evaluate the value of each element in the list $\mathbb{V}_i$ that discretely approximate the function V_i. Indeed, the above approximation can be expressed in terms of the general control variate method as

$$\mathbb{V}_i = O_B^N(\mathbb{P}^{\mathbb{A}}{}_{i+1} - \mathbb{P}^{\mathbb{E}}{}_{i+1}) + O_B(P_{i+1}^E), \tag{6}$$

where the difference of two lists $\mathbb{P}^{\mathbb{A}}{}_{i+1} - \mathbb{P}^{\mathbb{E}}{}_{i+1}$ is the list of the early exercise premium or the discretization of the difference between the functions X and Y. Note that applying O_B on the European put value function at time step $i+1$, P_{i+1}^E, will obtain the function P_i^E. This can be

[4]Note that P_{ij}^E and P_i^E are different symbols distinguished by the number of elements in the subscript. The latter symbol denotes the European put value function at the time step i.

proved by applying the tower rule on Eq. (1) as follows:

$$\begin{aligned} P_i^E(S_{i\Delta t}) &= \mathrm{E}_Q\left[e^{-r(T-i\Delta t)}(K-S_T)^+|S_{i\Delta t}\right] \\ &= \mathrm{E}_Q\left[e^{-r\Delta t}\mathrm{E}_Q\left[e^{-r(T-(i+1)\Delta t)}(K-S_T)^+|S_{(i+1)\Delta t}\right]|S_{i\Delta t}\right] \\ &= \mathrm{E}_Q\left[e^{-r\Delta t}P_{i+1}^E S_{(i+1)\Delta t}|S_{i\Delta t}\right]. \end{aligned}$$

By taking advantage of the general control variate method in Eq. (6), we get the list $\mathbb{V}_i$ that is more accurate than the list $\hat{\mathbb{V}}_i$ evaluated by directly applying the backward induction. The list of American put values $\mathbb{P}^{\mathbb{A}}{}_i$ is then obtained by substituting $\mathbb{V}_i$ into Eq. (5). Note that $\mathbb{P}^{\mathbb{A}}{}_i$ tends to be more accurate than $\hat{\mathbb{P}}_i^{\mathbb{A}}$ since $\mathbb{V}_i$ is more accurate than $\hat{\mathbb{V}}_i$ Our accurate pricing method for American put is constructed by repeatedly applying the above procedure as described in Algorithm 1. Note that other numerical methods in the following sections will also be described in this format without giving a specificized example like Fig. 2 for brevity.

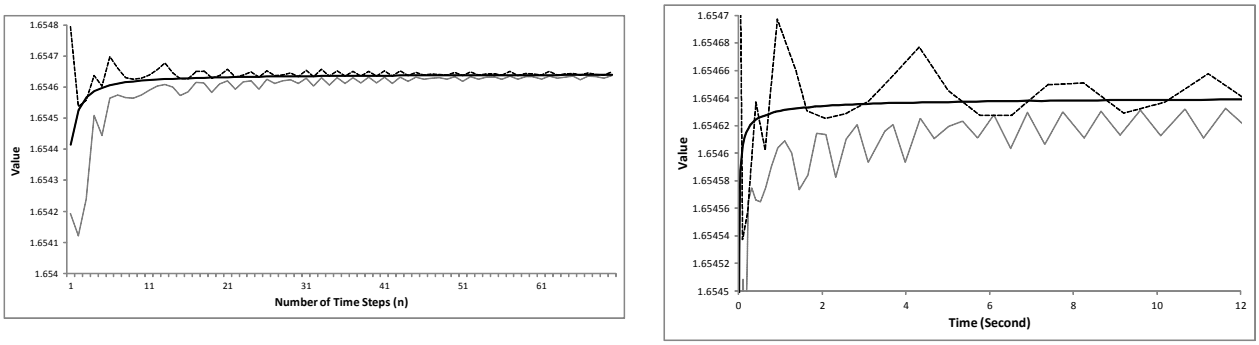
Algorithm 1 The General Control Variate Method for Pricing American Puts on a n-Time-Step Tree.

1: Evaluate the list of the early exercise premium $\mathbb{E}_n(\equiv \mathbb{P}^{\mathbb{A}}{}_n - P_n^E)$.
2: **for** $i = n-1$ down to 0 **do**
3: Evaluate the list of the continuation value $\mathbb{V}_i$ by substituting $\mathbb{E}_{i+1}$ into Eq. (6).
4: Evaluate the list of the American put value $\mathbb{P}^{\mathbb{A}}{}_i$ by substituting $\mathbb{V}_i$ into Eq. (5).
5: Evaluate the list of the early exercise premium $\mathbb{E}_i(\equiv \mathbb{P}^{\mathbb{A}}{}_i - P_i^E)$.
6: **end for**
7: The pricing result is the American put value at the beginning node of the tree; i.e, P_{01}^A.

3.3 Numerical Experiments

Recall that the pricing results of a tree method converge to the theoretical value (about 1.64563 in this example) as the number of time steps n increases. Compared to the results generated by directly applying the backward induction method to the tree model (denoted by the gray dotted line), the Hull and White (1988) method applies the control variate method once on the overall pricing results (denoted by black dashed line) and slightly improve the accuracy for pricing American puts as illustrated in Fig. 3 (a). That is, given the same number of time steps (of the tree method), the pricing result generated by the Hull and White (1988) method tends to be closer to 1.65463 than the result generated by the direct backward induction. In addition, our approach repeatedly applies the general control variate method and the pricing results (denoted by the black solid line) significantly improve the accuracy and reduce the oscillation problem.

Note that a sophisticated pricing method like our approach would require extra computations (like evaluating the Black-Scholes pricing formula many times) and hence long computational time. Thus evaluating the performances among different pricing methods should compare the convergence rates of these methods in terms of computational time as illustrated in Fig. 3 (b). Indeed, most of extra computations in Algorithm 1 can be skipped by taking advantage of the property of the early exercise boundary (see Curran, 1994). For convenience, we call a node (i,j) an "early-exercise node" if the option holder decides to exercise the American put immediately at that node (i.e, $V_{ij} < K - S(i,j)$); otherwise, we call it a "non-early-exercise node". There exists a default boundary, like the gray curve in Fig. 2, that divide the tree nodes into two groups: the group of non-early-exercise nodes in the upper part of the tree and the group of early-exercise nodes in the lower part. We mimic the Curran (1994) method to estimate the default boundary and use the general control variate method to improve the estimation. For each non-early-exercise node (i,j), we directly evaluate the early exercise premium E_{ij} without calculating P_{ij}^A; that is, we evaluate E_{ij} as $\frac{pE_{i+1,j}+qE_{i+1,j+1}}{R}$ without involving the Black-Scholes



(a) (b)

Figure 3: Convergence Comparisons for Pricing American Puts

The x axes in panel (a) and (b) denote the number of time steps (in thousands) and the computational time, respectively. The y axes in both panels denote the American put prices generated by different numerical methods. The gray dotted lines denote the results generated by directly applying the backward induction method on the CRR tree. The black dashed lines denote the results generated by Hull and White (1988) method. The black solid lines denote the results generated by our method. The initial underlying asset's price S_0 is 40, the time to maturity T is 3 year, the strike price K is 35, the risk-free rate r is 0.05, and the volatility σ is 0.2.

pricing formula. For early-exercise nodes, we only calculate the values for the node right below the default boundary. The evaluation for other early-exercise nodes can be skipped without influencing the pricing results as mentioned in Curran (1994). Indeed, the above concept can be used to improve all tree-based pricing algorithms analyzed in this numerical experiment. The experiment suggests that our method still converges much faster than the other two methods in terms of the computational time as in Fig. 3 (b).

4 Characteristic-Function-Based Pricing Methods

4.1 Background Introduction

Many price processes, say the variance gamma process studied by **?** and the stochastic volatility model studied by Heston (1993), are proposed to fit the market phenomena better. However, this makes derivative valuation become intractable since under those underlying assumptions, the analytical formulae for even the European options no longer exist. Fortunately, the return characteristic function for a vast class of underlying processes are still available. Therefore, many scholars have proposed plenty of literature to take advantage of the return characteristic functions to price derivatives (see e.g. Bakshi and Chen (1997), Bates (1996), Chen and Scott (1992)). Among all characteristic-function-based pricing methods, the method studied by Carr and Madan (1999) is the most computationally favorable one since it further incorporates the computational power of the FFT into the contingent valuation region.

Consider a target underlying dynamics G by which we are interested in the European call price implied. Let C_G^{EU} and ψ_G respectively denote the European call price and the Fourier transform of the damped call price implied by the dynamics G. Denote by S_t the underlying asset's price at time t. Carr and Madan (1999) propose that the value of an European call option with strike price e^k and time to maturity T is:

$$C_G^{\mathrm{E}}(e^k) = E_Q\left[e^{-rT}(S_T - e^k)^+\right] = \int_k^\infty (e^s - e^k)q(s)\,ds = \frac{e^{-dk}}{\pi}\int_0^\infty e^{-ivk}\psi_G(v)\,dv, \tag{7}$$

where q is the risk-neutral density of $\ln S_T$ implied by the dynamics G; ϕ_G denotes the characteristic function of q:

$$\phi_G(v) \equiv \int_{-\infty}^\infty e^{ivs}q(s)\,ds;$$

ψ is the Fourier transform of the damped call price $c^{\mathrm{E}}(e^k) \equiv e^{dk}C^{\mathrm{E}}(e^k)$ for some damping constant $d > 0$ [5], which can be further expressed in terms of ϕ:

$$\psi_G(v) \;=\; \frac{e^{-rT}\,\phi_G(v-(d+1)i)}{d^2+d-v^2+i(2d+1)v}. \tag{8}$$

As a result, the call value in Eq. (7) can be approximated by the Trapezoid rule numerically:

$$C_G^{\mathrm{E}}(e^k) \approx \frac{e^{-dk}}{\pi}\sum_{j=0}^{N} e^{-i\eta jk}\psi_G(j\eta)\frac{\eta}{2}\left[2-\delta_j-\delta_{N-j}\right], \tag{9}$$

where η is the distance between quadrature points; N is the number of quadrature points; δ_j is the Kronecker delta which equals 1 when j is zero and 0 otherwise.

Note that this method can also be used to simultaneously evaluate N otherwise identical European options with different strike prices via the FFT.

4.2 The Use of Our Generalized Control Variate Technique on the Characteristic-Function-Based Method

In this section, we will demonstrate the application of our general control variate technique on the characteristic-function-based method.

[5] The Fourier transform of $C^{\mathrm{E}}(e^k)$ doesn't exist since $C^{\mathrm{E}}(e^k)$ tends to S_0 as k tends to $-\infty$, and therefore is not a square integrable function. Carr and Madan (1999) invoke the damping constant d so that the Fourier transform of $c^{\mathrm{E}}(e^k)$ exists.

Assume that there is a suitable proxy function ψ_{proxy} such that a closed-form solution for the European call option on the underlying dynamics implied by ψ_{proxy}, which is denoted by $C^{\text{EU}}_{\text{proxy}}$, is available. Let $\psi_{\text{residual}}(v) \equiv \psi_G(v) - \psi_{\text{proxy}}(v)$ be the difference between ψ_G and ψ_{proxy}. Based on the decomposition above, C^{EU}_G can be decomposed into a proxy part and a residual part:

$$\begin{aligned} C^{\text{EU}}_G(e^k) &= \frac{e^{-dk}}{\pi}\int_0^\infty e^{-ivk}\psi_G(v)\,dv \\ &= \frac{e^{-dk}}{\pi}\int_0^\infty e^{-ivk}[\psi_{\text{proxy}}(v)+\psi_{\text{residual}}(v)]\,dv \\ &\equiv C^{\text{EU}}_{\text{proxy}}(e^k) + C^{\text{EU}}_{\text{residual}}(e^k). \end{aligned} \tag{10}$$

Since we choose a proxy function ψ_{proxy} which closely approximates ψ so that the amplitude of ψ_{residual} becomes smaller and smoother than ψ, the amplitude of the second order derivative of ψ_{residual} will have a better chance to be smaller than ψ. Because the Trapezoid rule is used for numerical approximation, the quadrature errors incurred are bounded above by the supremum of the second order derivative of the integrand. Consequently, our method will have a better chance to generate less quadrature errors because the upper bound of quadrature error has been significantly improved.

In this paper we choose ψ_{proxy} based on the Merton's jump-diffusion model. Merton (1976) assumes that the underlying asset's dynamics follows the stochastic differential equation:

$$dS_t = \left(\mu + \frac{\sigma^2}{2}\right) S_t\,dt + \sigma S_t\,dW_t + (e^{J_t}-1)S_t\,dM_t, \tag{11}$$

where μ and σ^2 are respectively the instantaneous mean and variance of the return conditional on the absence of the Poisson events; W_t is a standard Brownian motion; M_t is a Poisson process with intensity λ, which is independent of W_t; J_t is a sequence of independent normal random variables with mean μ_J and standard derivation σ_J. In Merton's jump-diffusion model, the characteristic function of the logarithm of underlying asset's price can be expressed as an infinite series:

$$\phi_{\text{MJD}}(v) = \exp\left(-\lambda T + iv(\mu T + \ln S_0) - \frac{1}{2}\sigma^2 T v^2\right)\sum_{h=0}^{\infty}\frac{(\lambda T\exp(iv\mu_J - \frac{\sigma_J^2 v^2}{2}))^h}{h!}.$$

Let H be a positive integer. We choose the first H terms of ϕ_{MJD} to play the role of ψ_{proxy}:

$$\psi_{\text{proxy}}(v) \equiv \frac{e^{-(\mu+\frac{\sigma^2}{2}+\lambda(e^{\mu_J+\frac{\sigma_J^2}{2}}-1))T}\,\phi_{\text{MJD}_H}(v-(d+1)i)}{d^2+d-v^2+i(2d+1)v},$$

where

$$\phi_{\text{MJD}_H}(v) \equiv \exp\left(-\lambda T + iv(\mu T + \ln S_0) - \frac{1}{2}\sigma^2 T v^2\right)\sum_{h=0}^{H-1}\frac{(\lambda T\exp(iv\mu_J - \frac{\sigma_J^2 v^2}{2}))^h}{h!}$$

Consequently, the proxy part call price will be:

$$C^{\text{E}}_{\text{proxy}}(e^k) = \sum_{h=0}^{H-1}\frac{e^{-\lambda' T}(\lambda' T)^h}{h!}\text{BS}_C(S_0, e^k, \sigma_h, r_h) \quad \text{for} \quad u = 0, 1, \ldots, m-1,$$

where $\lambda' \equiv \lambda e^{\mu_J+\frac{\sigma_J^2}{2}}$; $\text{BS}_C(S_0, e^k, \sigma_h, r_h)$ denotes a list of Black-Scholes call option prices with underlying asset's price S_0, strike price e^k, time to maturity T, risk-free rate $r_h \equiv \mu + \frac{\sigma^2}{2} + \frac{h(\mu_J+\frac{\sigma_J^2}{2})}{T}$, and volatility rate $\sigma_h \equiv \sqrt{\sigma^2 + \frac{h\sigma_J^2}{T}}$.

Define by $\psi_{\mathbb{G}} \equiv \{\psi_G(0), \psi_G(\eta), \cdots, \psi_G(N\eta)\}$ the list of ψ's which is used to estimate $C_G^{\mathrm{E}}(e^k)$ in Eq. (9)and $\psi_{\mathbb{PROXY}} \equiv \{\psi_{\mathrm{proxy}}(0), \psi_{\mathrm{proxy}}(\eta), \cdots, \psi_{\mathrm{proxy}}(N\eta)\}$. Let O_{IF} symbolize the Fourier inversion operation. The original Carr and Madan's method which applies the Fourier inversion numerically to $\psi_{\mathbb{G}}$ to price options:

$$C_G^{\mathrm{E}}(e^k) = O_{\mathrm{IF}}^N(\psi_{\mathbb{G}}).$$

In contrast, our generalized control variate method exploits the information of $\psi_{\mathbb{G}} - \psi_{\mathbb{PROXY}}$ to reduce the numerical errors generated by estimating $C_G^{\mathrm{E}}(e^k)$:

$$C_G^{\mathrm{E}}(e^k) = O_{\mathrm{IF}}^N(\psi_{\mathbb{G}} - \psi_{\mathbb{PROXY}}) + O_{\mathrm{IF}}(\psi_{\mathrm{proxy}}).$$

Next we calibrate the parameters for the proxy characteristic function to approximate the characteristic function implied by the dynamics G. We solve for the corresponding parameter values through cumulant matching method. Since the characteristic functions (for a vast class of process) admit infinite differentiability, we assume the first five cumulants of $\ln S_T$ dynamics G exists and equal to $m_1 T + \ln S_0$, $m_2 T$, $m_3 T$, $m_4 T$, and $m_5 T$, respectively. The key idea is to match the first five cumulants of $\ln S_T$ for dynamics G with those for Merton's jump-diffusion model, which can be obtained through the differentiation of ϕ_{MJD}. This yields

$$\begin{aligned}
&(\mu + \lambda\mu_J)\, T + \ln S_0 = m_1 T + \ln S_0, && (12)\\
&\left(\sigma^2 + \lambda\left(\mu_J^2 + \sigma_J^2\right)\right) T = m_2 T, && (13)\\
&\lambda(\mu_J^3 + 3\mu_J\sigma_J^2)T = m_3 T, && (14)\\
&\lambda(\mu_J^4 + 6\mu_J^2\sigma_J^2 + 3\sigma_J^4)T = m_4 T, && (15)\\
&\lambda(\mu_J^5 + 10\mu_J^3\sigma_J^2 + 15\mu_J\sigma_J^4)T = m_5 T. && (16)
\end{aligned}$$

Note that the left hand sides of the above equations are the first five cumulants of $\ln S_T$ for Merton's jump-diffusion model.

We first rewrite Eq. (14) as

$$\sigma_J^2 = \frac{m_3 - \lambda\mu_J^3}{3\lambda\mu_J} \tag{17}$$

and then substitute all the σ_J^2 in Eq. (15) and Eq. (16) into Eq. (17). This yields

$$\begin{aligned}
&-2\mu_J^6\lambda^2 + (4m_3\mu_J^3 - 3\mu_J^2 m_4)\lambda + m_3^2 = 0,\\
&-2\mu_J^6\lambda^2 - 3\mu_J m_5\lambda + 5m_3^2 = 0.
\end{aligned}$$

By equating the above two equations, we get

$$\lambda = \frac{4m_3^2}{3m_5\mu_J - 3m_4\mu_J^2 + 4m_3\mu_J^3}. \tag{18}$$

Now we can replace the λ and σ_J^2 in Eq. (14) by Eq. (17) and Eq. (18) respectively and get a polynomial equation of μ_J as follows:

$$48\mu_J^4 m_3^4 - 120\mu_J^3 m_3^3 m_4 + 9\mu_J^2 m_3^2\left(8m_3m_5 + 5m_4^2\right) - 54\mu_J m_3^2 m_4 m_5 + 9m_3^2 m_5^2 = 0. \tag{19}$$

Clearly, the variable μ_J in Eq. (19) can be easily solved since it is a polynomial equation of μ_J with degree 4. If there are two or more roots among the real numbers, we pick the one implies smallest λ because in this case, $C_{\mathrm{proxy}}^{\mathrm{EU}}$ will play a major role in C_G^{EU}[6]. However, it is possible

[6]Note that we invoke the first H terms of the call price implied by Merton's model as the the proxy part call price. Therefore, we choose the answer with smaller λ since in this case, the probability mass of the Poisson distribution will be more concentrated on the first H terms.

that Eq. (19) does not have any roots among the real numbers. If so, we simply pick a proper μ_J which minimizes the absolute value of the left hand side of Eq. (19). Once μ_J is determined, the other 4 parameters, λ, σ_J, μ, and σ, can also be determined sequentially by Eqs. (18), (17), (12), and (13).

Algorithm 2 sums up the use of our generalized control variate method on Carr and Madan's characteristic-function-based pricing method.

Algorithm 2 The General Control Variate Method for the Characteristic-Function-Based Method.

1: Find a suitable proxy function ψ_{proxy} to approximate ψ_G ;
2: Decompose the call price list $\mathbb{C}_G^{\text{EU}}(e^{k_u})$ into $\mathbb{C}_{\text{proxy}}^{\text{EU}}(e^{k_u})$ and $\mathbb{C}_{\text{residual}}^{\text{EU}}(e^{k_u})$;
3: Evaluate the price list $\mathbb{C}_{\text{proxy}}^{\text{EU}}(e^{k_u})$ by analytical formulae and $\mathbb{C}_{\text{residual}}^{\text{EU}}(e^{k_u})$ via the FFT;
4: Add them up and get $\mathbb{C}_G^{\text{EU}}(e^{k_u})$.

4.3 Numerical Results

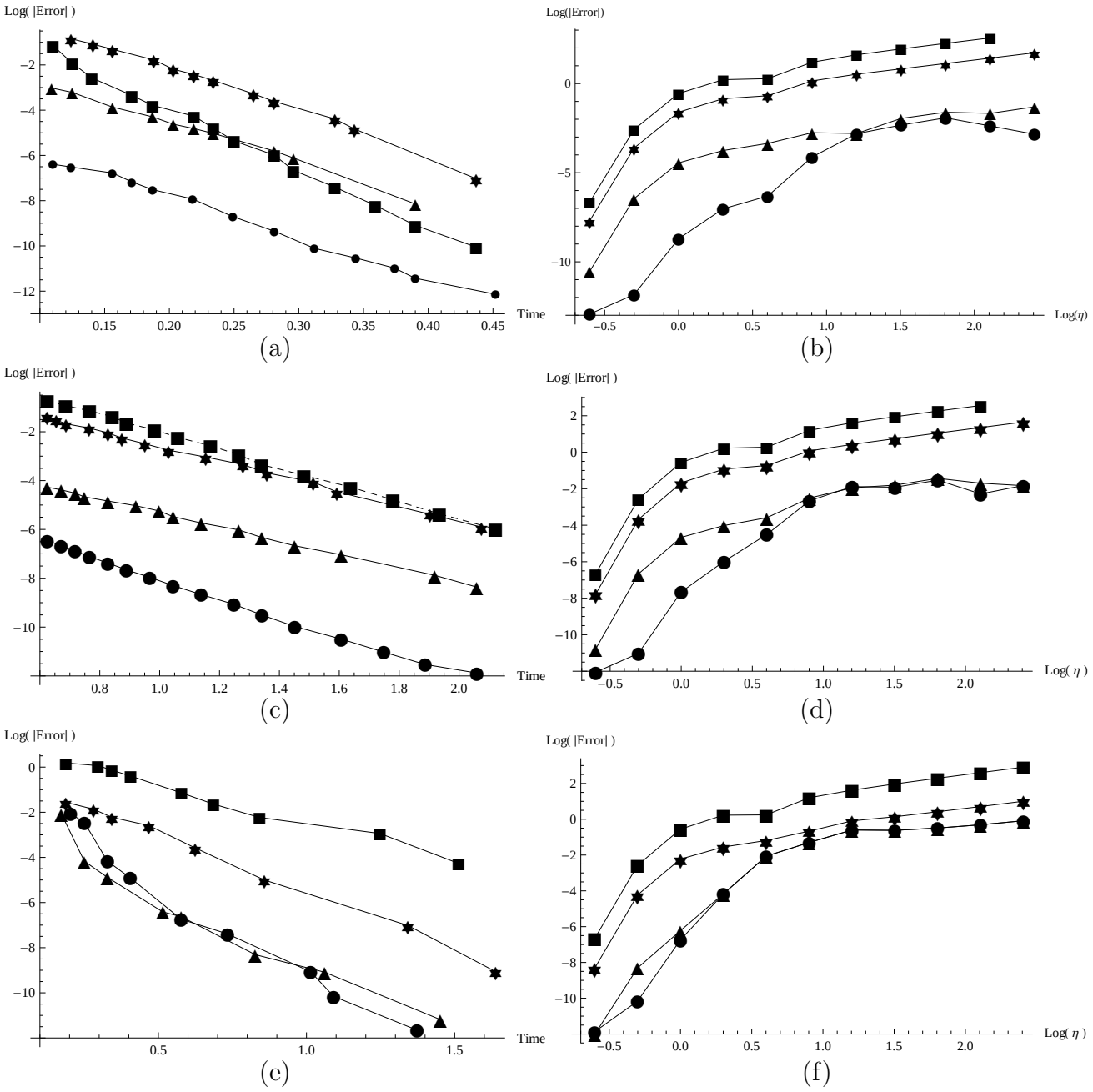


Figure 4: Convergence of pricing results.

The pricing errors are computed from, in panel (a) and (b), the VG model with $S_0 = 100$, $K = 100$, $r = 0$, $T =$ 4 months, $\varsigma = 0.1213$, $\nu = 0.1686$, $\vartheta = -0.1436$, in panel (c) and (d), the Heston model with $S_0 = 100$, $K = 100$, $r = 0$, $T =$ 4 months, $\kappa = 1.49$, $\theta = 0.0671$, $\epsilon = 0.742$, $\rho = -0.571$, $V_0 = 0.0262$ and, in panel (e) and (f), Kou's jump-diffusion model with $S_0 = 100$, $K = 98$, $r = 0.05$, $T =$ 6 months, $\omega_1 = 10$, $\omega_2 = 5$, $\tilde{\lambda} = 1$, $p = 0.4$, $q = 0.6$, $\tilde{\sigma} = 0.16$. In panel (a), (c), and (e), lines are computational time used plotted against the logarithm of the absolute pricing errors. In panel (b), (d), and (f), lines are the logarithm of the grid spacing plotted against the logarithm of the absolute pricing errors. The squares denote the pricing results for the original Carr and Madan's method; the stars, triangles, and circles denote the pricing results for the Carr and Madan's method in combination with our generalized control variate technique with $H = 1, 3$, and 7, respectively. The damping coefficient, the number of quadrature points, and the grid spacing are set to be 1.5, 4096, and 0.25, respectively, as in Carr and Madan (1999). The pricing results implied by the Carr and Madan's method with $N = 2^{20}$ and $\eta = 2^{-10}$ are chosen as the benchmarks.

5 Convolution-Based Asian Option Pricing Method

5.1 Background Introduction

Consider an arithmetic Asian call option with strike price K initiated at time 0 which matures at time T. Denote by S_t its underlying asset's price at time t. The payoff of an Asian call option is dependent on the average underlying asset's price sampled at times t_0, t_1, t_2, $\ldots, t_n$, where $0 = t_0 < t_1 < t_2 < \cdots < t_n = T$. Define the average underlying asset's price as

$$A \equiv \frac{1}{n+1} \sum_{i=0}^{n} S_{t_i}.$$

According to Harrison and Pliska (1981), the value of an arithmetic Asian call option can be expressed as the expectation of the discounted payoff :

$$C_{\text{AS}} = E_Q \left[e^{-rT} (A - K)^+ \right]. \tag{20}$$

Since the expectation in (20) is intractable, Carverhill and Clewlow (1990) and Benhamou (2002) discuss an useful convolution-based method for pricing arithmetic Asian options.

The underlying asset's return from time t_{i-1} to time t_i, denoted as $R_i \equiv \ln(S_{t_i}/S_{t_{i-1}})$, is assumed to follow the same Brownian motion in Eq. (3). Define $U_i \equiv e^{R_i} = S_{t_i}/S_{t_{i-1}}$. Since $S_{t_i} = S_{t_0} e^{R_1 + R_2 + \cdots R_i}$, Eq. (20) can be rewritten as:

$$\begin{aligned}
A &= \frac{1}{n+1} \sum_{i=0}^{n} S_{t_i} = \frac{S_{t_0}}{n+1} \left(1 + U_1 + U_1 U_2 + \cdots + U_1 U_2 \cdots U_n\right) \\
&= \frac{S_{t_0}}{n+1} \left(1 + U_1 \left(1 + U_2 \left(\cdots U_{n-1} \left(1 + U_n\right)\right)\right)\right) \\
&= \frac{S_{t_0}}{n+1} \left(1 + e^{R_1} \left(1 + e^{R_2} \left(\cdots e^{R_{n-1}} \left(1 + e^{R_n}\right)\right)\right)\right) \\
&= \frac{S_{t_0}}{n+1} \left(1 + e^{R_1 + \ln(1 + \exp(\cdots + \ln(1 + \exp(R_n))))}\right) \\
&= \frac{S_{t_0}}{n+1} \left(1 + e^{D_0 + \mu_0}\right), \qquad (21)
\end{aligned}$$

where μ_i is defined by the following recurrence relation:

$$\begin{aligned}
\mu_{n-1} &= E[R_n] \\
\mu_{i-1} &= E[R_i] + \ln(1 + \exp(\mu_i)), \qquad i = n-1, n-2, \ldots, 1;
\end{aligned}$$

the sequence $\{D_i\}$ is defined recursively as follows:

$$D_{n-1} = R_n - \mu_{n-1}, \tag{22}$$

$$D_{i-1} = R_i + Z_i, \qquad i = n-1, n-2, \ldots, 1, \tag{23}$$

where

$$Z_i \equiv \ln(1 + \exp(D_i + \mu_i)) - \mu_{i-1}. \tag{24}$$

Let f_U represent the density function of a random variable U for convenience. Since Z_i is a increasing function of D_i, f_{Z_i} can be expressed by a function of f_{D_i} according to the *Jacobian transformation method*:

$$f_{Z_i}(x) = \begin{cases} f_{D_i}(\ln(e^{x+\mu_{i-1}} - 1) - \mu_i) \frac{e^{x+\mu_{i-1}}}{e^{x+\mu_{i-1}} - 1} & \text{if } x > -\mu_{i-1}, \\ 0 & \text{otherwise.} \end{cases} \tag{25}$$

Besides, according to Eq. (23), the density function $f_{D_{i-1}}$ equals the convolution of f_{R_i} and f_{Z_i}. Thus $f_{D_{i-1}}$ can be efficiently calculated by applying the inverse Fourier transform on the product of the characteristic functions of f_{R_i} and f_{Z_i}. Since the analytical solution for the inverse transform is not available, Benhamou (2002) harnesses the fast Fourier transform to evaluate $f_{D_{i-1}}$ on a grid list $-b = x_{-m} < \cdots < x_0 = 0 < \cdots < x_m = b$ between $[-b, b]$, a window which contains the bulk of the probability mass of all densities involved in his algorithm, and stores it in $\mathbb{D}_{i-1}$. Moreover, when estimating f_{Z_i} with Eq.(25), the Jacobian transformation method has to be implemented in combination with numerical interpolation. It is because only the density list $\mathbb{D}_i$ but the analytical formula of f_{D_i} is available. Therefore Benhamou (2002) numerically estimates a list of density points on f_{Z_i} by the Jacobian transformation method and stores it in $\mathbb{Z}_i$.

Algorithm 3 sums up the Carverhill-Clewlow and Benhamou algorithm, where the list of density estimate for f_{R_i} is stored in $\mathbb{R}_i$.

Algorithm 3 Carverhill-Clewlow and Benhamou algorithm.

1: Evaluate the densities values in the list $\mathbb{D}_{n-1}$.
2: **for** $i = n - 1$ down to 1 **do**
3: Calculate $\mathbb{Z}_i$ by applying the Jacobian transformation method on $\mathbb{D}_i$ in combination with numerical interpolation (see Eq. (24)).
4: Calculate $\mathbb{D}_{i-1}$, the discrete convolution of $\mathbb{R}_i$ and $\mathbb{Z}_i$, via the FFT.
5: **end for**
6: Evaluate the Asian option value C_{AS} by the density function $\mathbb{D}_0$ and Eqs. (20) and (21).

5.2 The Use of Control Variate Technique on the Convolution-Based Pricing Method

Two operations, convolution and the Jacobian transformation, are alternatively applied repeatedly to help construct the forward probability density of an Asian option for pricing. In this section we are going to demonstrate the application of our generalized control variate method on the convolution-based pricing method studied by Carverhill and Clewlow (1990) and Benhamou (2002).

First of all, in the third step of Algorithm 3, we implement the the Jacobian transformation operation in combination with the linear interpolation with a view to estimating $\mathbb{Z}_i$. It is because only a density list $\mathbb{D}_i$ but the analytical formula of f_{D_i} is available; therefore the numerical interpolation is used to estimate $\{f_{D_i}(\ln(e^{x+\mu_{i-1}} - 1)) - \mu_i\}_{x=x_{-m}}^{x_m}$ (see Eq. (25)). Clearly, the accuracy of $\mathbb{Z}_i$ depends significantly on the estimation quality of $\{f_{D_i}(\ln(e^{x+\mu_{i-1}} - 1) - \mu_i)\}_{x=x_{-m}}^{x_m}$. When more accurate pricing results are required, usually plenty of computational grids have to be used to improve the interpolation quality. Nevertheless, doing this will deteriorate the pricing speed dramatically. It provides us a motivation to apply our generalized control variate technique to speed up the Carverhill-Clewlow and Benhamou algorithm.

Assume that there is a proxy function $f_{D_i}^{\text{proxy}}$ which serves as a control variate for estimating the density list $\{f_{D_i}(\ln(e^{x+\mu_{i-1}} - 1) - \mu_i)\}_{x=x_{-m}}^{x_m}$. We decompose f_{D_i} into two parts: $f_{D_i}^{\text{proxy}}$ and $f_{D_i}^{\text{residual}}$, where $f_{D_i}^{\text{residual}} \equiv f_{D_i} - f_{D_i}^{\text{proxy}}$ be the difference between f_{D_i} and $f_{D_i}^{\text{proxy}}$. According to the density decomposition above, $\mathbb{D}_i$ can also be decomposed into $\mathbb{D}_i^{\text{proxy}}$ and $\mathbb{D}_i^{\text{residual}}$, where $\mathbb{D}_i^{\text{proxy}}$ denotes the list of density values calculated from $f_{D_i}^{\text{proxy}}$ on the same grids as $\mathbb{D}_i$; $\mathbb{D}_i^{\text{residual}} \equiv \mathbb{D}_i - \mathbb{D}_i^{\text{residual}}$ denotes the difference between $\mathbb{D}_i$ and $\mathbb{D}_i^{\text{residual}}$ [7]. To estimate $\mathbb{Z}_i$ more accurately, instead of directly applying the Jacobian transformation numerically to $\mathbb{D}_i$, we first calculate

[7] If we view $\mathbb{D}_i$ and $\mathbb{D}_i^{\text{residual}}$ as m-dimensional Euclidean vectors, then the minus operator can be treated as vector subtraction.

the densities list $\{f_{D_i}^{\text{proxy}}(\ln(e^{x+\mu_{i-1}}-1)-\mu_i)\frac{e^{x+\mu_{i-1}}}{e^{x+\mu_{i-1}}-1}\}_{x=x_{-m}}^{x_m}$ by the control variate $f_{D_i}^{\text{proxy}}$, and then apply the Jacobian transformation numerically to $\mathbb{D}_i^{\text{residual}}$ in order to estimate the densities $\{f_{D_i}^{\text{residual}}(\ln(e^{x+\mu_{i-1}}-1)-\mu_i)\frac{e^{x+\mu_{i-1}}}{e^{x+\mu_{i-1}}-1}\}_{x=x_{-m}}^{x_m}$. Finally, the two estimations above will add up to $\mathbb{Z}_i$. In this way the evaluation of $\mathbb{Z}_i$ will become much more efficient since the use of our generalized control variate technique significantly suppress the numerical errors generated by estimating $\{f_{D_i}(\ln(e^{x+\mu_{i-1}}-1)-\mu_i)\}_{x=x_{-m}}^{x_m}$; therefore fewer grid points (hence less computational time) are needed to attain a given accuracy level.

Eq.(26) illustrates the first application of our generalized control variate technique on the convolution-based pricing method: The operation J symbolizes the Jacobian transformation, which can be used to estimate a list of probability densities of a transformed random variable; $\{f_{D_i}^{\text{proxy}}(\ln(e^{x+\mu_{i-1}}-1)-\mu_i)\frac{e^{x+\mu_{i-1}}}{e^{x+\mu_{i-1}}-1}\}_{x=x_{-m}}^{x_m}$ is the density list which can be calculated from the control variate term.

$$\mathbb{Z}_i = J^{\text{CV}}(\mathbb{D}_i) = J^{\text{N}}(\mathbb{D}_i^{\text{residual}}) + \{f_{D_i}^{\text{proxy}}\left(\ln(e^{x+\mu_{i-1}}-1)-\mu_i\right)\frac{e^{x+\mu_{i-1}}}{e^{x+\mu_{i-1}}-1}\}_{x=x_{-m}}^{x_m}. \tag{26}$$

Moreover, in the fourth step of Algorithm 3, discrete convolutions are employed to estimate $\mathbb{D}_{i-1}$. Our generalized control variate method can be harnessed to speed up this step as well. Assume that there is a proxy function $f_{Z_i}^{\text{proxy}}$ which possesses the following two critical properties to play the role of a control variate when estimating $\mathbb{D}_{i-1}$: One is that $f_{Z_i}^{\text{proxy}}$ should closely approximates f_{Z_i}. The other is that there should be an analytical solution for the convolution of $f_{Z_i}^{\text{proxy}}$ and f_{R_i}. Let $f_{Z_i}^{\text{residual}} \equiv f_{Z_i} - f_{Z_i}^{\text{proxy}}$ be the difference between f_{Z_i} and $f_{Z_i}^{\text{proxy}}$. According to the density decomposition above, $\mathbb{Z}_i$ can also be divided into $\mathbb{Z}_i^{\text{proxy}}$ and $\mathbb{Z}_i^{\text{residual}}$, where $\mathbb{Z}_i^{\text{proxy}}$ denotes the list of density values calculated from $f_{Z_i}^{\text{proxy}}$ on the same grid points as $\mathbb{Z}_i$; $\mathbb{Z}_i^{\text{residual}} \equiv \mathbb{Z}_i - \mathbb{Z}_i^{\text{proxy}}$ denotes the differences between $\mathbb{Z}_i$ and $\mathbb{Z}_i^{\text{proxy}}$. Based on the set-ups above, the discrete convolution step in Algorithm 3 can be decomposed into two parts: The discrete convolution of $\mathbb{Z}_i^{\text{proxy}}$ and $\mathbb{R}_i$ plus the discrete convolution of $\mathbb{Z}_i^{\text{residual}}$ and $\mathbb{R}_i$. The first part can be evaluated analytically by the closed-form solution for the convolution of $f_{Z_i}^{\text{proxy}}$ and f_{R_i} without invoking any numerical techniques. Only the second part needs to be evaluated via the FFT. Comparing to the original the Carverhill-Clewlow and Benhamou algorithm which only takes advantage of the information on the density list $\mathbb{R}_i$, our algorithm exploits more information on the analytical formula f_{R_i}. This could be one of the explanations why the use of our generalized control variate technique can suppress numerical errors generated by discrete convolution.

Eq.(27) illustrates the second application of our generalized control variate technique on the convolution-based pricing method: Beside invoking the FFT to evaluate the discrete convolution of $\mathbb{Z}_i$ and $\mathbb{R}_i$, we additionally take advantage of the closed-form solution for the convolution of $f_{Z_i}^{\text{proxy}}$ and f_{R_i} to make better estimates for the convolution of $\mathbb{Z}_i$ and $\mathbb{R}_i$. The notation $\otimes$ denotes the convolution.

$$\mathbb{D}_{i-1} = \otimes^{\text{CV}}(\mathbb{Z}_i, \mathbb{R}_i) = \otimes^{\text{N}}(\mathbb{Z}_i^{\text{residual}}, \mathbb{R}_i) + \otimes(f_{Z_i}^{\text{proxy}}, f_{R_i}). \tag{27}$$

Algorithm 4 sums up the use of control variate method on Benhamou algorithm

One of the most critical problems is that what functions could be chosen as suitable control variates, namely, $f_{D_i}^{\text{proxy}}$ and $f_{Z_i}^{\text{proxy}}$. Since we assume the underlying assets dynamics follows the geometric Brownian motion, one natural candidate for the control variates could be the probability density function of the normal distribution function.

Here comes another problem: what parameters, say mean and variance, should we choose for the control variates? Now we will introduce an idea to tackle this problem. Suitable parameters for the control variates can be determined by moment matching method. The first step is to find the first few moments of D_i and Z_i.

Comparing to determining the first n^{th} order moments of D_i and Z_i, it is much easier to

Algorithm 4 The use of control variate technique on Carverhill-Clewlow and Benhamou algorithm

1: Evaluate the densities values in the list $\mathbb{D}_{n-1}$;
2: **for** $i = n-1$ **downto** 1 **do**
3: Decompose $\mathbb{D}_i$ into $\mathbb{D}_i^{\text{proxy}}$ and $\mathbb{D}_i^{\text{residual}}$;
4: Calculate $\mathbb{Z}_i$ by applying the Jacobian transformation method on $\mathbb{D}_i^{\text{residual}}$ in combination with numerical interpolation and the analytical formula of $f_{D_i}^{\text{proxy}}$ (see Eq. (26));
5: Decompose $\mathbb{Z}_i$ into $\mathbb{Z}_i^{\text{proxy}}$ and $\mathbb{Z}_i^{\text{residual}}$;
6: Calculate $\mathbb{D}_{i-1}$ via the discrete convolution of $\mathbb{Z}_i^{\text{residual}}$ and $\mathbb{R}_i$ and the analytical convolution of $f_{Z_i}^{\text{proxy}}$ and f_{R_i} (see Eq. (27)).
7: **end for**
8: Evaluate the Asian option value C_{AS} by the density function $\mathbb{D}_0$ and Eq. (21)

solve for the moments of e^{D_i} and e^{Z_i}:

$$e^{D_i} = U_n(1+U_{n-1}(1+U_{n-2}(\ldots U_{i+2}(1+U_{i+1}(1+U_i))))e^{-\mu_i}, \quad (28)$$
$$e^{Z_i} = 1+U_n(1+U_{n-1}(1+U_{n-2}(\ldots U_{i+2}(1+U_{i+1}(1+U_i))))e^{-\mu_i}. \quad (29)$$

The moments of e^{D_i} and e^{Z_i} can be determined iteratively – for example, to determine the n^{th} moment of e^{D_i}, the first step is to determine the n^{th} moment of U_i by by utilizing the characteristic function of the normal random variable since U_i is log-normally distributed. Afterwards, the n^{th} moment of $1+U_i$ can be easily determined as well because it is a linear combination of the first $n-1$ moments of U_i. The third step is to calculate the n^{th} moment of $U_{i+1}(1+U_i)$. It equals the product of the n^{th} moments of U_{i+1} and $(1+U_i)$ because U_{i+1} is independent of $(1+U_i)$. After several similar operations we can obtain the n^{th} moment of $U_n(1+U_{n-1}(1+U_{n-2}(\ldots U_{i+2}(1+U_{i+1}(1+U_i)))))$. In the end, the n^{th} moment can be computed by multiplying the n^{th} moment of $U_n(1+U_{n-1}(1+U_{n-2}(\ldots U_{i+2}(1+U_{i+1}(1+U_i)))))$ by $e^{-n\mu_i}$. Although D_i and Z_i are not *exactly* normally distributed, they are composed of normal random variables. Accordingly, if we *view* them as normal random variables, then the following system of equations hold and can be used to solve for the mean and variance of D_i (Z_i):

$$E(e^{D_i}) = e^{E(D_i)+Var(D_i)/2},$$
$$Var(e^{D_i}) = \left(e^{Var(D_i)}-1\right)e^{2E(D_i)+Var(D_i)},$$

where $E(e^{D_i})$ and $Var(e^{D_i})$ denote the expected value and variance of e^{D_i} respectively. It turns out that

$$E(D_i) = \ln\frac{\left(E(e^{D_i})\right)^2}{\sqrt{Var(e^{D_i})+\left(E(e^{D_i})\right)^2}} \equiv \mu_{D_i},$$
$$Var(D_i) = \ln\left(1+\frac{Var(e^{D_i})}{\left(E(e^{D_i})\right)^2}\right) \equiv \sigma_{D_i}^2.$$

Note that the mean and variance for $f_{Z_i}^{\text{proxy}}$ can also be determined by the same method above.

Furthermore, the value of an Asian call option can be decomposed into two parts based on

the density decomposition of f_{D_0}:

$$\begin{aligned} C_{\text{AS}} &= e^{-rT} E\left[\left(\frac{S_{t_0}}{n+1}\left(1+e^{D_0+\mu_0}\right)-K\right)^+\right] \\ &= e^{-rT}\int_{-\infty}^{\infty}\left(\frac{S_{t_0}}{n+1}\left(1+e^{x+\mu_0}\right)-K\right)^+ f_{D_0}(x)dx \\ &= e^{-rT}\int_{-\infty}^{\infty}\left(\frac{S_{t_0}}{n+1}\left(1+e^{x+\mu_0}\right)-K\right)^+ \left(f_{D_0}^{\text{proxy}}+f_{D_0}^{\text{residual}}(x)\right)dx \\ &\equiv C_{\text{AS}}^{\text{proxy}}+C_{\text{AS}}^{\text{residual}} \end{aligned} \tag{30}$$

Since $f_{D_0}^{\text{proxy}}(x)$ is a normal density, obviously there is a closed-form solution for $C_{\text{AS}}^{\text{proxy}}$:

$$C_{\text{AS}}^{\text{proxy}} = \frac{S_0 e^{-rT+\mu_0+\mu_{D_0}+\frac{\sigma_{D_0}^2}{2}}}{(n+1)}\Phi\left(\frac{\mu_{D_0}+\sigma_{D_0}^2-\xi}{\sigma_{D_0}}\right)+\left(\frac{S_{t_0}}{n+1}-K\right)e^{-rT}\Phi\left(\frac{\mu_{D_0}-\xi}{\sigma_{D_0}}\right), \tag{31}$$

where $\Phi(\cdot)$ denotes the standard normal distribution function; ξ denotes $\ln\left(K(n+1)/(S_{t_0})\right)-\mu_0$, the solution of $S_{t_0}\left(1+e^{x+\mu_0}\right)/n+1=0$. Only the term $C_{\text{AS}}^{\text{residual}}$ has to be evaluated numerically; therefore numerical errors can be further suppressed. Eq. (31) can be viewed as an application of our generalized control variate method as well: We perform analytical integration to calculate $C_{\text{AS}}^{\text{proxy}}$ and numerical integration to estimate $C_{\text{AS}}^{\text{residual}}$.

6 Conclusions

References

Bakshi, G. and Z. Chen (1997). An alternative valuation model for contingent claims. *Journal of Financial Economics 44*(1), 123–165.

Bates, D. S. (1996). Jumps and stochastic volatility: Exchange rate processes implicit in deutsche mark options. *Review of Financial Studies 9*(1), 69–107.

Benhamou, E. (2002). Fast fourier transform for discrete asian options. *Journal of Computational Finance 6*(1), 49–68.

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

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

Carr, P. and D. Madan (1999). Option valuation using the fast Fourier transform. *Journal of Computational Finance 2*(4), 61–73.

Carr, P. and L. Wu (2004). Time-changed Lévy processes and option pricing. *Journal of Financial Economics 71*(1), 113–141.

Carverhill, A. and L. Clewlow (1990). Flexible convolution. *Risk 3*(4), 25–29.

Chen, R.-R. and L. Scott (1992). Pricing interest rate options in a two-factor cox-ingersoll-ross model of the term structure. *Review of Financial Studies 5*(4), 613–636.

Cox, J., S. Ross, and M. Rubinstein (1979). Option pricing: A simplified approach. *Journal of Financial Economics 7*(3), 229–263.

Curran, M. (1994). Accelerating american option pricing in lattices. *Journal of Derivatives 3*, 101–123.

Dai, T.-S., L.-M. Liu, and Y.-D. Lyuu (2007). Linear-time Option Pricing Algorithms by Combinatorics. *Computers and Mathematics with Applications 55*, 2142–2157.

Dingec, K. D. and W. Hormann (2012). A general control variate method for option pricing under levy processes. *European Journal of Operational Research 221*, 368–377.

Duan, J. (1995). The GARCH option pricing model. *Mathematical Finance 5*(1), 13–32.

Duffie, D. (1996). *Dynamic Asset Pricing Theory*. Princeton, NJ: Princeton University Press.

Glasserman, P. (2004). *Monte Carlo Methods in Financial Engineering*. New York: Springer.

Han, C. and Y. Lai (2010). Generalized control variate methods for pricing asian options. *Journal of Computational Finance 14*(2), 87–118.

Harrison, J. M. and S. R. Pliska (1981). Martingales and stochastic integrals in the theory of continuous trading. *tochastic Processes and Their Applications 11*, 215–260.

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

Hull, J. and A. White (1988). The use of the control variate technique in option pricing. *Journal of Financial and Quantitative Analysis 23*(3), 237–251.

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

Lyuu, Y. (1998). Very fast algorithms for barrier option pricing and the ballot problem. *Journal of Derivatives 5*(3), 68–79.

Merton, R. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics 3*(1-2), 125–144.