Rev Deriv Res (2008) 11:83–118
DOI 10.1007/s11147-008-9025-y

Adaptive placement method on pricing arithmetic average options

Tian-Shyr Dai · Jr-Yan Wang · Hui-Shan Wei

Published online: 14 November 2008

Abstract Since there is no analytic solution for arithmetic average options until present, developing an efficient numerical algorithm becomes a promising alternative. One of the most famous numerical algorithms is introduced by Hull and White (J Deriv 1:21–31, 1993). Motivated by the common idea of reducing the nonlinearity error in the adaptive mesh model in Figlewski and Gao (J Financ Econ 53:313–351, 1999) and the adaptive quadrature method, we propose an adaptive placement method to replace the logarithmically equally-spaced placement rule in the Hull and White's model by placing more representative average prices in the highly nonlinear area of the option value as the function of the arithmetic average stock price. The basic idea of this method is to design a recursive algorithm to limit the error of the linear interpolation between each pair of adjacent representative average prices. Numerical experiments verify the superior performance of this method for reducing the interpolation error and hence improving the convergence rate. To show that the adaptive placement method can improve any numerical algorithm with the techniques of augmented state variables and the piece-wise linear interpolation approximation, we also demonstrate how to integrate the adaptive placement method into the GARCH option pricing algorithm in Ritchken and Trevor (J Finance 54:377–402, 1999). Similarly great improvement

T.-S. Dai
Department of Information and Finance Management, National Chiao Tung University, Hsinchu, Taiwan
e-mail: cameldai@mail.nctu.edu.tw

J.-Y. Wang (✉)
Department of International Business, National Taiwan University, No. 1, Sec. 4, Roosevelt Rd, Taipei 106, Taiwan
e-mail: jywang@management.ntu.edu.tw

H.-S. Wei
Department of Finance, National Central University, Taoyuan County, Taiwan
e-mail: selenaway@yahoo.com.tw

of the convergence rate suggests the potential applications of this novel method to a broad class of numerical pricing algorithms for exotic options and complex underlying processes.

Keywords Arithmetic average options · Interpolation error · Equally-spaced placement · Adaptive placement

JEL Classification G13

Asian options are path dependent securities whose payoffs depend on the average of the underlying prices during the option life. They were originally issued in 1987 by Bankers Trust Tokyo on crude oil contracts and hence with the name "Asian" option. The averaging method can be either the arithmetic or geometric average. In addition, a further categorization of these options relies on either the price of the underlying asset at maturity or the strike price being replaced by the average price.

In practice, Asian options are appropriate to meet the hedging needs of users of commodities, energies, or foreign currencies who will be exposed to average prices during a future period. Meanwhile, since the volatility for the average of the underlying prices is inclined to be lower than the volatility for the underling prices, Asian options tend to be less expensive than corresponding vanilla options and are therefore more attractive for some investors. In addition, Asian options are also useful in thinly-traded markets to prevent price manipulation. To this date, more and more financial instruments include the average feature, for example, structure notes issued by many international banks, the contracts of convertible bonds in Taiwan, etc. Therefore, an efficient pricing model is indispensable for financial institutions to manage these sorts of products.

However, the average feature complicates the evaluation of Asian options. If the underlying price process follows the geometric Brownian motion, the analytical pricing formula for geometric average options is feasible since the product of lognormally distributed prices remains to follow the lognormal distribution. Based upon this observation, Kemna and Vorst (1990) propose an analytical solution for European geometric average options. Unfortunately, it is still analytically intractable to price arithmetic average options due to the lack of proper mathematical representation for the sum of lognormal random variables. Thus many researches were devoted to deal with the distribution of the sum of lognormal random variables and derive the approximate pricing formulae for arithmetic average options. Several works along this direction include the fast Fourier transformation in Carverhill and Clewlow (1990), the Edgeworth series expansion in Turnbull and Wakeman (1991), the reciprocal Gamma distribution in Milevsky and Posner (1998), the Laplace transform inversion in Geman and Yor (1993), etc.

The tree-based model is also a possible alternative to value arithmetic average options by introducing the arithmetic average price as a path-dependent state variable. The naive pricing method on the tree-based model, which tracks all possible arithmetic average prices reaching each node, is able to derive exact option values for arithmetic average options without any interpolation approximation. It is worth noting that although exact option values from the naive pricing method are free of interpolation errors, they are still susceptible to the discretization error, which is a trait of the tree-based model. While the time step becomes infinitesimal in length, exact option values will converge to diffusion limit option values.

Unfortunately, the naive pricing method is intractable due to the exponential growth of the number of possible arithmetic average prices with respect to the number of time steps in the tree-based model. To deal with this problem, Dai and Lyuu (2002) develop a trinomial-tree model to generate exact arithmetic average option values by reducing the complexity of recording all possible arithmetic average prices. Although the computational complexity of their model is proven to be sub-exponential with respect to the number of time steps, it is still intractable to price arithmetic average options via this model when the number of time steps is large.

In contrast, instead of keeping track of all possible arithmetic average prices, Hull and White (1993) introduce representative average prices to be logarithmically equally-spaced placed between the maximum and minimum arithmetic average prices for each node.[1] In addition, the piece-wise linear interpolation is adopted to approximate the corresponding option values for nonexistent average prices during the phase of backward induction. As a consequence, the interpolation error emerges and pricing results might not converge to exact option values unless the number of representative average prices for each node is large enough and well collocated with the number of time steps in the tree-based model, see Forsyth et al. (2002). On the other hand, Lin and Ritchken (2006) propose an option pricing algorithm on the tree-based model with an augmented path-dependent state variable. Without the need of a large number of representative values of the path-dependent state variable at each node, they suggest tracking only the conditional expectation of the path-dependent state variable for each node. The success of their algorithm to generate convergent option values is mostly due to employing the conditional expectation of the path-dependent state variable which is strongly related with the option value under some requisite conditions.[2]

Along with the line of Hull and White (1993), Neave and Turnbull (1994) suggest using the conditional frequency distribution to adjust the number of representative average prices at each node. In addition, Cho and Lee (1997) replace the uniform allocation of the number of representative average prices in the Hull and White's model with the distribution of the number of possible geometric average prices. Klassen (2001) and Forsyth et al. (2002) propose revised versions of the algorithm of Hull and White. In their setting, a set of representative average prices at each node is considered, and the grid space for the logarithmically arithmetic average prices is suggested to be a function of the time to maturity, the number of time steps, and the volatility of the stock price process.

[1] In fact, taking the maximum and minimum arithmetic average prices of each node into account is a revised version of the Hull and White's model. In their original paper, the representative average prices are the same for all nodes at each point in time, and they are logarithmically equally-spaced distributed between the maximum and minimum arithmetic average prices for that point in time.

[2] As long as the transition probabilities of the tree model and the evolution process of the path-dependent state variable are linear in the path-dependent state variable, they prove that the results generated by this algorithm can converge to diffusion limit option values when the time step becomes infinitesimal in length.

The aforementioned works are dedicated to reducing the interpolation error by devising more efficient non-uniform allocation rules to replace the uniform allocation rule in Hull and White (1993). The difference between the uniform and non-uniform allocations is shown in Fig. 1, where the numbers of representative average prices are M and $M(i, j)$ for $node(i, j)$ in the uniform and non-uniform allocation rules, respectively. With the uniform allocation rule being replaced, however, the logarithmically equally-spaced placement rule in Hull and White (1993) is still retained in these works.

In this paper, the interpolation error is minimized in a novel way. Aiming at simultaneously guaranteeing the convergence of the interpolation error and improving the efficiency of the Hull and White's model, the adaptive placement method is developed to replace the logarithmically equally-spaced placement rule. Our method argues that more representative average values are needed in the area around which the option value as a function of the arithmetic average prices is with higher degree of curvature, and fewer representative average values are placed where the option value function is with lower degree of curvature. To achieve this goal, the adaptive placement method is actually designed to govern the error of the linear interpolation between each pair of adjacent representative average prices under a limit criterion. In Fig. 1, the logarithmically equally-spaced placement rule and our adaptive placement rule are illustrated as well.

On the other hand, instead of considering the range between the maximum and minimum arithmetic average prices at each node, more compact ranges for the arithmetic average price at each node are derived in Aingworth et al. (2000) and Dai et al. (2002) for European and American arithmetic average options, respectively. Both these methods are proven to reduce the linear interpolation error effectively. Nevertheless, the numerical results in this paper demonstrate that the ideas of the algorithms of Aingworth et al. (2000) and Dai et al. (2002) are implicitly incorporated in our adaptive placement method as natural byproducts.

In addition, there are another two kinds of methods with different points of view to deal with arithmetic average options. First, Chalasani et al. (1998) and Chalasani et al. (1999) introduce the concept of "nodelet" that the paths reaching a node is partitioned such that each nodelet represents the paths with the same geometric average price from time zero to that node. During the phase of backward induction, the Hull and White's algorithm is applied nodelet by nodelet to finding the upper bound of the value of the arithmetic average option. Meanwhile, together with the technique of the iterative expectation and the Jensen's inequality, the lower bound for the arithmetic average option can be derived as well. Second, in a recent work of Costabile et al. (2006), a recursive algorithm based on the binomial tree is proposed to choose representative average prices among all possible arithmetic average prices for each node. Their results show that employing this algorithm together with the piece-wise linear interpolation for option values of nonexistent average prices can produce sufficiently accurate values for arithmetic average options. Our numerical results show that the adaptive placement method can generate more accurate arithmetic average option values than these two kinds of methods without difficulty under a moderate average number of representative average prices per node.

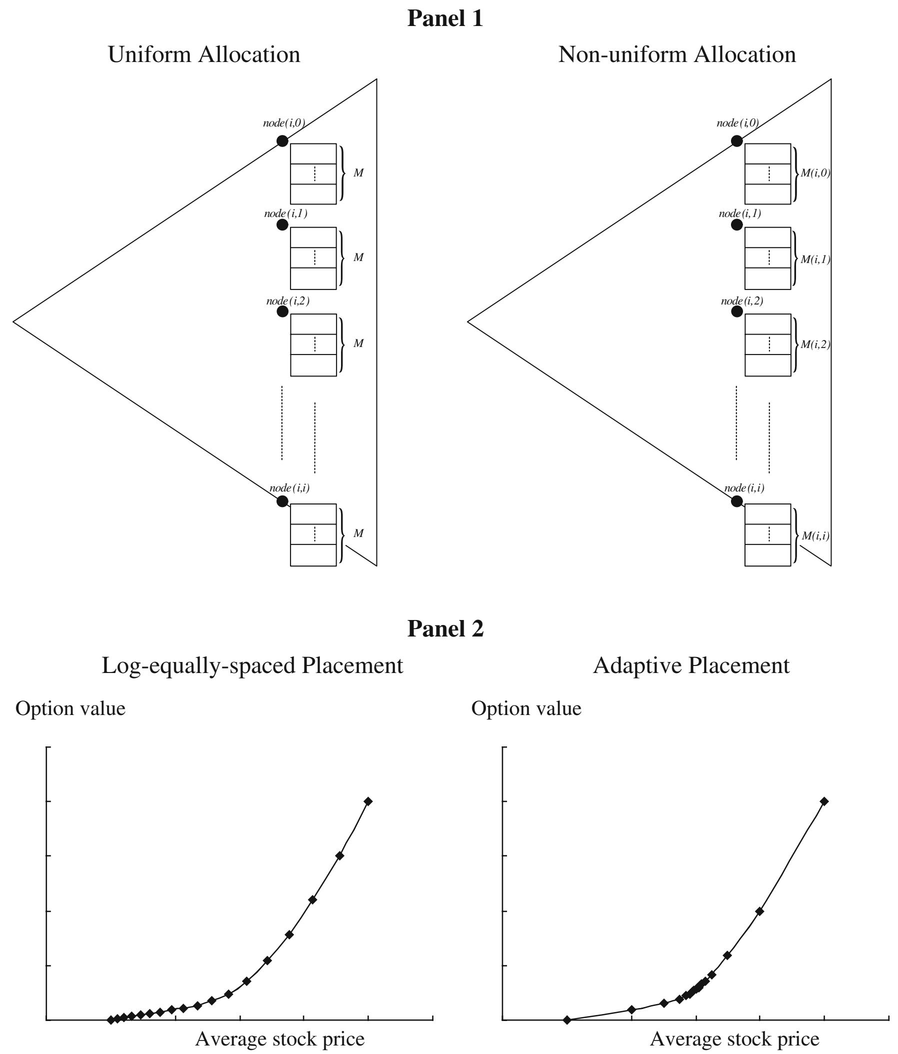

Fig. 1 The illustration of the uniform and non-uniform allocations and logarithmically equally-spaced and our adaptive placement rules. Hull and White (1993) adopt the uniform allocation, i.e. the numbers of representative average prices for all nodes are the same to be a constant integer M. Many modifications of the Hull and White's model focus on devising more efficient non-uniform allocation rules, i.e. they adjust the number of representative average prices for $node(i, j)$, $M(i, j)$, depending on the probability reaching the underlying node, the time to maturity of the underlying node, the number of time steps, and the volatility of the underlying process. Nevertheless, in the Hull and White's model and its modifications, it is still common that M (or $M(i, j)$) representative average prices at each node are distributed following the logarithmically equally-spaced placement rule. In contrast, our adaptive placement model places representative average prices proportional to the degree of curvature so that the linear interpolation error can be minimized effectively and an efficient non-uniform allocation of representative average prices over the tree is achieved automatically

In fact, the adaptive placement method can be extended to all numerical algorithms with the techniques of augmenting state variables and applying the piece-wise linear interpolation in the backward induction. The GARCH option pricing models in Ritchken and Trevor (1999) and Cakici and Topyan (2000) are typical examples for this kind of algorithm. In order to demonstrate the generality of our adaptive placement method, we also apply the adaptive placement method to the Cakici and Topyan's GARCH option pricing model. The numerical results still exhibit the impressive efficiency improvement for the adaptive placement method to deal with the GARCH option pricing problem.

This paper is organized as follows. Section 1 describes the arithmetic average options and the famous Hull and White's model. Our adaptive placement method will be elaborated in Sect. 2. In Sect. 3, extensive numerical experiments for pricing arithmetic average options are conducted to verify the superior performance of our adaptive placement method to existing methods. In addition, we examine the performance of our adaptive placement method for the GARCH option pricing problem in Sect. 4. Section 5 offers the conclusion.

1 Arithmetic average options and the Hull and White's model

In this paper, a non-dividend-paying underlying asset in the risk neutral world is considered and its price is assumed to follow the geometric Brownian motion

$$\frac{dS_t}{S_t} = r dt + \sigma dZ_t,$$

where S_t denotes the stock price at time t, r is the risk free rate, σ is the volatility of the asset price, and Z_t is the Wiener process in the risk neutral world. Suppose that the stock price is sampled at the time points $t_0 = 0 < t_1 < \cdots < t_N = T$ during the life of the arithmetic average options. The arithmetic average price from time t_0 to time t is defined as

$$A(t) = \frac{1}{l+1} \sum_{i=0}^{l} S_{t_i}, \text{ where } t_l \leq t < t_{l+1}.$$

Here we focus on the pricing of the fixed-strike-price arithmetic average call whose payoff at time t is $\max(A(t) - X, 0)$, where X is the strike price. The extension to other types of Asian options is straightforward. Furthermore, the stock price is assumed to be sampled periodically, and therefore $t_i = i\Delta t$ with $\Delta t = T/N$.

In the field of option pricing, the binomial-tree model introduced by Cox et al. (1979) is viewed as a useful tool to deal with European-style as well as American-style options. This model divides the time horizon of an option into N discrete time steps and discretizes the stock prices at each time step. In Fig. 2 , it is shown that the stock price at time step 0 is S_0 (at $node(0, 0)$), and the stock price can either move up to $S_0 u$ (at $node(1, 0)$) with probability $p = (e^{r\Delta t} - d)/(u - d)$ or down to $S_0 d$ (at $node(1, 1)$) with probability $1 - p$ at the first time step, where $u = e^{\sigma\sqrt{\Delta t}}$ and

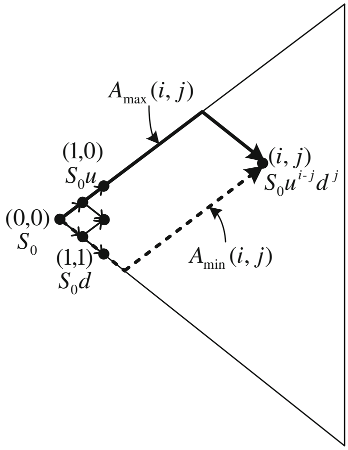

Fig. 2 The paths corresponding to the maximum and minimum arithmetic average prices for $node(i, j)$. The stock price today is denoted by S_0, and the stock price can either move up to S_0u or down to S_0d at the first step. Likewise, each stock price can either move up or move down at subsequent time steps. The $node(i, j)$ stands for the node at time point i with j cumulative down movements and $S_0u^{i-j}d^j$ is the corresponding stock price. $A_{max}(i, j)$ and $A_{min}(i, j)$ correspond to the maximum and minimum arithmetic average prices among all possible paths from $node(0, 0)$ to $node(i, j)$

$d = e^{-\sigma\sqrt{\Delta t}}$ are the magnitudes of the upward and downward movements. Similarly, each stock price can either move up or move down at subsequent time steps.

One of the most famous binomial-tree-based models to price arithmetic average options efficiently is proposed by Hull and White (1993). In their model, to avoid tracking all possible arithmetic average prices of each node, only the maximum and the minimum arithmetic average prices among all traversed paths for each node are calculated, which is illustrated in Fig. 2. For $node(i, j)$ with the stock price $S_0u^{i-j}d^j$ for $0 \leq j \leq i \leq N$, the maximum arithmetic average price is contributed by a price path starting with $i - j$ consecutive up movements followed by j consecutive down movements, whose value can be derived by $A_{max}(i, j) = (S_0\frac{1-u^{i-j+1}}{1-u} + S_0 \cdot u^{i-j} \cdot d \cdot \frac{1-d^j}{1-d})/(i+1)$. Likewise, the value of the corresponding minimum arithmetic average price can be calculated from a price path starting with j consecutive down movements followed by $i - j$ consecutive up movements: $A_{min}(i, j) = (S_0\frac{1-d^{j+1}}{1-d} + S_0 \cdot d^j \cdot u \cdot \frac{1-u^{i-j}}{1-u})/(i+1)$.

Once equipped with the knowledge about the maximum and minimum arithmetic average prices for each node, representative average prices are arrayed logarithmically equally-spaced from the maximum to the minimum arithmetic average prices for each node via the following formula.[3]

[3] In this paper, for the ease of the comparisons between the Hull and White's model and other examined methods, we do not follow the original setting of introducing a fixed parameter h to represent the minimal movement of the average price in Hull and White (1993). Instead we set the number of representative average prices for each node to be M, which corresponds to the value of h being $\ln((A_{max}(i, j)/(A_{min}(i, j))/(M-1)$ for each node.

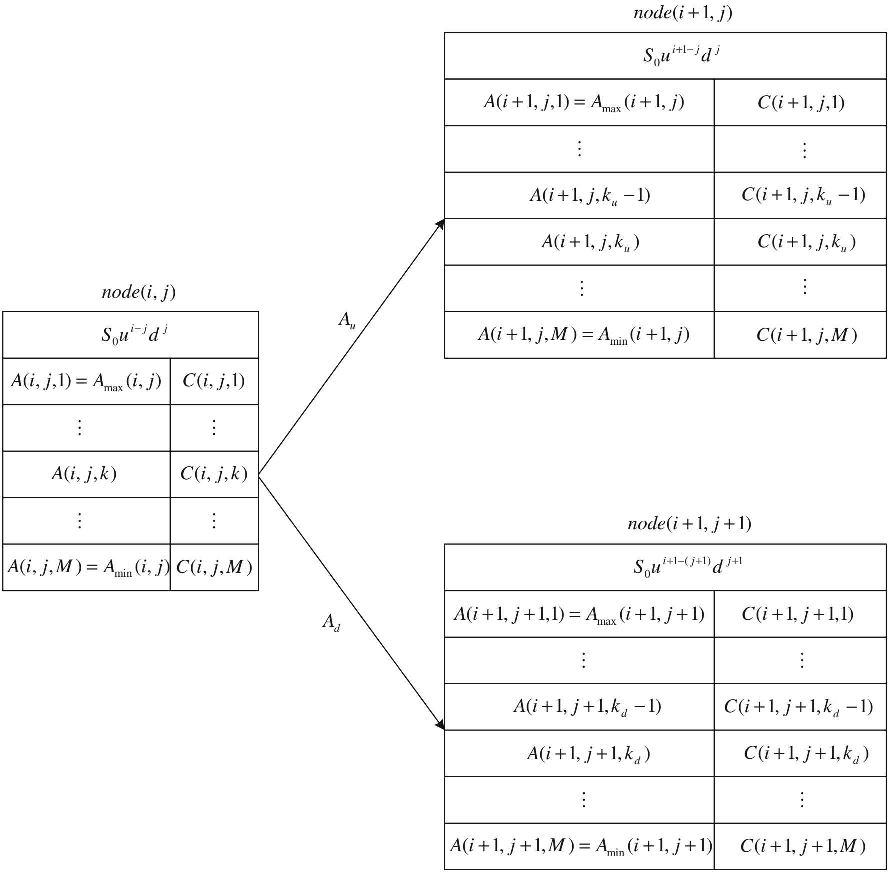

Fig. 3 Calculate the continuation value for each representative average price at $node(i, j)$. Each rectangle represents the data structure for each node on the binomial tree, in which the first row is the stock price for that node, and the following rows contain the pairs of representative average prices and their corresponding option values. For each representative average price $A(i, j, k)$ at $node(i, j)$, it is necessary to calculate the evolutions of the arithmetic average price A_u and A_d first and then to derive the corresponding option values C_u and C_d by the piece-wise linear interpolation approximation in Eq. 1. Finally, the continuation value for $A(i, j, k)$ is $C(i, j, k) = e^{-r\Delta t}(p \cdot C_u + (1-p) \cdot C_d)$

$$A(i, j, k) = \exp\left(\frac{M-k}{M-1}\ln(A_{max}(i, j)) + \frac{k-1}{M-1}\ln(A_{min}(i, j))\right), \text{ for } k = 1, \ldots, M.$$

After building the tree and the table of representative average prices for each node, we decide the payoff of each representative average price of the nodes at maturity first. Next, the option value is derived via the backward induction procedure from $node(i+1, j)$ and $node(i+1, j+1)$ to $node(i, j)$, which is illustrated in Fig. 3. For $A(i, j, k)$, the evolutions of the arithmetic average price at the next point in time are $A_u = ((i+1)A(i, j, k) + S_0u^{i+1-j}d^j)/(i+2)$ and $A_d = ((i+1)A(i, j, k) + S_0u^{i+1-(j+1)}d^{(j+1)})/(i+2)$.

For any average price $A \in [A_1, A_2]$, where A_1 and A_2 stand for any pair of adjacent representative average prices in the table of representative average prices at some node, the corresponding option value for A can be approximated from the linear interpolation method,

$$C = C_1 \cdot \frac{A - A_2}{A_1 - A_2} + C_2 \cdot \frac{A - A_1}{A_2 - A_1}, \tag{1}$$

where C_1 and C_2 are corresponding option values for A_1 and A_2. Suppose A_u is inside the range $[A(i+1, j, k_u), A(i+1, j, k_u - 1)]$. The corresponding option value C_u for A_u can be approximated by setting $C_1 = C(i+1, j, k_u)$, $C_2 = C(i+1, j, k_u - 1)$, $A_1 = A(i+1, j, k_u)$, $A_2 = C(i+1, j, k_u - 1)$, and $A = A_u$ in Eq. 1. Similarly, the corresponding option value C_d for $A_d \in [A(i+1, j+1, k_d), A(i+1, j+1, k_d - 1)]$ can be approximated following the same rule.

Finally, the continuation value for $A(i, j, k)$ is

$$C(i, j, k) = e^{-r\Delta t}(p \cdot C_u + (1-p) \cdot C_d). \tag{2}$$

If the feature of early exercise is taken into account, the option value corresponding to $A(i, j, k)$ becomes

$$\max(C(i, j, k), A(i, j, k) - X). \tag{3}$$

Following the above procedure for all $A(i, j, k)$ backward over the binomial tree, the value of $C(0, 0, 1)$ will be the arithmetic average option price derived by the Hull and White's model.

2 Our model

The goal of our adaptive placement method is to intelligently reduce the interpolation error for pricing arithmetic average options in a tree-based model. Motivated by the common idea of dealing with the nonlinearity error in the Figlewski and Gao's (1999) adaptive mesh model and the adaptive quadrature method,[4] our method differs from the Hull and White's method in the sense that more representative average prices are placed in the range where the option value function is with higher degree of curvature and fewer are placed in the range where the option value function is with lower degree of curvature.

The details of our adaptive placement method are explained as follows. For the linear interpolation approximation in Eq. 1, by the mean-value theorem, the linear interpolation error caused from the nonlinearity of the option value function in $[A_1, A_2]$ can be expressed as

$$\frac{C''(\xi)}{2!} \cdot (A - A_1) \cdot (A - A_2), \text{ for some number } \xi \text{ between } A_1 \text{ and } A_2. \tag{4}$$

[4] See Sect. 4.6 in Faires and Burden (2003) for reference.

Our adaptive placement method is designed to examine whether the error of the linear interpolation between each pair of adjacent representative average prices in Eq. 4 is below some pre-specified limit. If the error of the linear interpolation inside the range of $[A_1, A_2]$ is not negligible, i.e. $C''(\xi)$ is too large or the distance between A_1 and A_2 is too far, we divide $[A_1, A_2]$ into finer subsets by inserting an extra representative average price in between and then repeat the same procedure of examining the error of the linear interpolation for each subset. Once the value of the error term between any pair of adjacent representative average prices is smaller than the pre-specified limit, termed the *second order error criterion* in our method, this examining-and-dividing process is stopped.

In practice, another constant termed the *precision criterion* is also defined to represent the threshold of negligible refinement for both average prices and option values in our method. The above examining-and-dividing process is also terminated when the difference between adjacent representative average prices or their corresponding option values is smaller than this minimum criterion. The purpose of introducing the *precision criterion* is to prevent possibly infinite dividing caused from the non-differentiable point during the examining-and-dividing process.

Within each examination of the linear interpolation error, we approximate $C''(\xi)$ in Eq. 4 by the second order numerical differentiation. For any pair of adjacent representative average prices A_1 and A_2, the midpoint $A = (A_1 + A_2)/2$ is employed together to approximate the error term of the linear interpolation for this range. The approximation for Eq. 4 in our adaptive placement method is described in the following pseudo code.

Function ***Error Term of the Linear Interpolation***

/* Given the pairs of the arithmetic average price and its corresponding option value (A_1, C_1), $(\frac{A_1+A_2}{2} = A, C_A)$, and (A_2, C_2), approximate the error term $(C''(\xi)/2!)(A - A_1)(A - A_2)$ */

input: $A_1, C_1, A, C_A, A_2, C_2$
real: C_double_prime

if (Abs($C_1 - C_2$) or Abs($A_1 - A_2$) $\leq$ *precision criterion*)
    Error Term of the Linear Interpolation := 0;
else {
    //$C''(\xi)$ is approximated by the second order numerical differentiation
    C_double_prime := $((C_1 - C_A)/(A_1 - A) - (C_A - C_2)/(A - A_2))/(0.5 \times (A_1 - A_2))$;
    Error Term of the Linear Interpolation := (C_double_prime/2!) $\times (A - A_1) \times (A - A_2)$;
}

End

The procedures for the tree-building and the backward induction of our adaptive placement method are described as follows. During the phase of building the stock price tree, all possible arithmetic average prices for the nodes at the first four points in time are recorded since there are only a few possible arithmetic average prices for those nodes.[5] For other nodes, only the maximum and minimum arithmetic average prices for each node are recorded as representative average prices initially. In addition, if the strike price is within the range between the maximum and minimum arithmetic average prices, it is inserted into the table of representative average prices. This is because the

[5] In our settings, the first four points in time are t_0, t_1, t_2, and t_3. For the binomial tree model, the number of possible arithmetic average prices reaching each node at the first four points in time is not larger than three.

linear interpolation approximation performs poorly near the highly curved area of the option value function, and the area with the highest degree of curvature is usually near where the arithmetic average price equals the strike price. The results in our preliminary study show that this small change does help reducing the interpolation error for pricing arithmetic average options.

In our model, the tables of representative average prices for all nodes are mainly constructed during the phase of backward induction. For each node, the goal of adaptively placing representative average prices is to let the range between each pair of adjacent representative average prices be small enough such that the linear interpolation error throughout the range is smaller than the *second order error criterion*. To be more explicit, during the backward induction phase, for any $node(i, j)$, the linear interpolation error for each pair of adjacent representative average prices $[A(i, j, k), A(i, j, k+1)]$[6] is examined. If the approximate error term of the linear interpolation for $(A(i, j, k), C(i, j, k))$, $(A = (A(i, j, k) + A(i, j, k+1))/2, C_A)$, and $(A(i, j, k+1), C(i, j, k+1))$[7] is smaller than the *second order error criterion*, the linear interpolation error for the range $[A(i, j, k), A(i, j, k+1)]$ is considered to be small enough and no further processing will be conducted. Otherwise, the pair (A, C_A) is inserted into the table of representative average prices of $node(i, j)$, and the subsets $[A(i, j, k), A]$ and $[A, A(i, j, k+1)]$ are examined separately to check whether the linear interpolation errors inside them are small enough. The above examining-and-dividing process is repeated until the approximate error of the linear interpolation for every pair of adjacent representative average prices is less than the *second order error criterion*.

We take an example to illustrate the examining-and-dividing process step by step. Suppose $S_0 = X = 50$, $N = 40$, $T = 1$ year, $r = 10\%$, $\sigma = 80\%$, and both the *second order error criterion* and the *precision criterion* are 0.5. For pricing a European arithmetic average call, the examining-and-dividing process of $node(37, 25)$ is sketched in Fig. 4. Inside the black frame for each step, there are three pairs of representative average prices and the corresponding call values, and we also report the linear interpolation error when taking these three pairs of representative average prices and option values as the inputs to the **Function *Error Term of the Linear Interpolation***.

For the initial table of representative average prices of $node(37, 25)$, the maximum and minimum arithmetic average prices are 83.4062 and 12.3309, and the strike price 50 is inserted as a representative average price since 50 is within the range [12.3309, 83.4062]. The call values for these three average prices computed by the Hull and White's model in Sect. 1 are 28.1577, 0, and 0, respectively. In addition, the

[6] Here we abuse the notations $A(i, j, k)$ and $C(i, j, k)$ slightly, which are borrowed from the aforementioned Hull and White's model. Different from the previous setting in which k is an index number from 1 to a constant number M, k is from 1 to $M(i, j)$ in our adaptive placement method. In fact, after building the stock prices tree, if the strike price is inserted into the table of representative average prices, the number of representative average prices $M(i, j)$ equals 3. Otherwise, the number of representative average prices $M(i, j)$ is 2.

[7] We derive the corresponding option values $C(i, j, k)$, C_A, and $C(i, j, k+1)$ following the Hull and White's algorithm, i.e. via Eqs. 2 and 3.

Initial table for *node(37,25)* (*M(37,25)=3*)

Average	Call
83.4062	28.1577
50.0000	0.0000
12.3309	0.0000

Step 1. (*M(37,25)=4*)

Average	Call	Interpolation error	
83.4062	28.1577		
66.7031	**13.2048**	0.8740	(inserted)
50.0000	0.0000		
12.3309	0.0000		

Step 2. (*M(37,25)=4*)

Average	Call	Interpolation error	
83.4062	28.1577		
75.0546	**20.6813**	3.2452E-15	(not inserted)
66.7031	13.2048		
50.0000	0.0000		
12.3309	0.0000		

Step 3. (*M(37,25)=5*)

Average	Call	Interpolation error	
83.4062	28.1577		
66.7031	13.2048		
58.3515	**5.7284**	0.8740	(inserted)
50.0000	0.0000		
12.3309	0.0000		

Step 4. (*M(37,25)=5*)

Average	Call	Interpolation error	
83.4062	28.1577		
66.7031	13.2048		
62.5273	**9.4666**	3.7088E-15	(not inserted)
58.3515	5.7284		
50.0000	0.0000		
12.3309	0.0000		

Step 5. (*M(37,25)=6*)

Average	Call	Interpolation error	
83.4062	28.1577		
66.7031	13.2048		
58.3515	5.7284		
54.1758	**2.1788**	0.6853	(inserted)
50.0000	0.0000		
12.3309	0.0000		

Step 6. (*M(37,25)=6*)

Average	Call	Interpolation error	
83.4062	28.1577		
66.7031	13.2048		
58.3515	5.7284		
56.2637	**3.8593**	9.4341E-02	(not inserted)
54.1758	2.1788		
50.0000	0.0000		
12.3309	0.0000		

Step 7. (*M(37,25)=6*)

Average	Call	Interpolation error	
83.4062	28.1577		
66.7031	13.2048		
58.3515	5.7284		
54.1758	2.1788		
52.0879	**0.7221**	0.3673	(not inserted)
50.0000	0.0000		
12.3309	0.0000		

Step 8. (*M(37,25)=6*)

Average	Call	Interpolation error	
83.4062	28.1577		
66.7031	13.2048		
58.3515	5.7284		
54.1758	2.1788		
50.0000	0.0000		
31.1655	**0.0000**	0.0000	(not inserted)
12.3309	0.0000		

Final table for *node(37,25)* (*M(37,25)=6*)

Average	Call
83.4062	28.1577
66.7031	13.2048
58.3515	5.7284
54.1758	2.1788
50.0000	0.0000
12.3309	0.0000

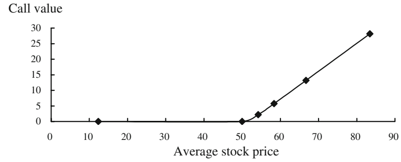

Fig. 4 The numerical example for the examining-and-dividing process. The values of parameters in this example are $S_0 = X = 50$, $N = 40$, $T = 1$, $r = 10\%$, $\sigma = 80\%$, and the *second order error criterion* and the *precision criterion* are both 0.5. The examining-and-dividing process for $node(37, 25)$ is illustrated step by step in this figure. Each table stands for a one-step change of the table of representative average prices and their corresponding call values for $node(37, 25)$. Inside the black frame for each step, there are three pairs of representative average prices and the corresponding call values, and we also report the linear interpolation error when taking these three pairs of representative average prices and option values as the inputs to the **Function** ***Error Term of the Linear Interpolation***. Once the approximate linear interpolation error is larger than the *second order error criterion* 0.5, the pair of the representative average price and the call value in boldface will be inserted into the table of representative average prices of $node(37, 25)$, and the number of representative average price for that node, $M(37, 25)$, is increased by one. In the final table, the approximate linear interpolation error for any pair of adjacent representative prices is bounded above by the *second order error criterion*. In addition, the option value as the function of the average price is plotted as well. It is apparent that the above examining-and-dividing process will place representative average prices proportional to the degree of the curvature of the option value as a function of the arithmetic average price

number of representative average prices for $node(37, 25)$ is $M(37, 25)$, which is set to be three after the tree-building procedure.

When the backward induction procedure progresses to $node(37, 25)$, the examining-and-dividing process of our adaptive placement method is described as follows. In step 1, the pairs of (83.4062, 28.1577), ((83.4062 + 50)/2 = 66.7031, 13.2048), and $(50, 0)$ are as the inputs to the **Function** ***Error Term of the Linear Interpolation*** to approximate the linear interpolation error for the range between 83.4062 and 50. Because the approximate linear interpolation error 0.8740 is larger than the *second order error criterion* 0.5, the pair of the average price 66.7031 and the corresponding call value 13.2048 should be inserted into the table of representative average prices. In the meanwhile, the number of representative average prices for this node, $M(37, 25)$, increases by one to become four.

In step 2, the approximate linear interpolation error in the range between 83.4062 and 66.7031 is 3.2452E−15, which is smaller than the *second order error criterion* 0.5. Therefore, we do not insert the pair of the average price 75.0546(=(83.4062 + 66.7031) /2) and the corresponding option value 20.6813 since the linear interpolation works pretty well in the range between 83.4062 and 66.7031. Thus, the value of $M(37, 25)$ remains to be four. For steps 3 to 8, once the approximate linear interpolation error for the three pairs of representative average prices and call values in the black frame is larger than the *second order error criterion* 0.5, the pair of the representative average price and the call value in boldface is inserted into the table of representative average prices of $node(37, 25)$, and the number of representative average price for that node, $M(37, 25)$, is increased by one. As a consequence, the approximate linear interpolation error for any pair of adjacent representative prices in the final table is bounded above by the *second order error criterion*.

In addition, the option value as the function of the average price is also plotted in Fig. 4. It is apparent that the above examining-and-dividing process will balance the distribution of placed representative average prices so that it is proportional to the degree of the curvature of the option value as a function of the arithmetic average price. Since the performance of the piece-wise linear interpolation is poor around where the option value function is with high degree of curvature, our algorithm places more representative average prices in these areas to reduce the error of the piece-wise linear interpolation. On the other hand, due to the satisfactory performance of the piece-wise linear interpolation for dealing with the option value function with low degree of curvature, our algorithm argues that less representative average prices placed in those areas will be sufficient.

As to the nodes at the first four points in time, this examining-and-dividing process is not necessary, and only the corresponding option values for all possible arithmetic average prices of each node should be calculated. Finally, $C(0, 0, 1)$ stands for the value of the arithmetic average option.

3 Numerical results

This section provides extensive numerical results to study the characteristic and the performance of our adaptive placement model. First, we take European as well as American arithmetic average calls as examples to show the different characteristics between our method and the classic Hull and White's model. After that, the performance comparison is conducted among our method, the Hull and White's model, and some methods proved to be efficient on pricing arithmetic average options, including Aingworth et al. (2000), Dai et al. (2002), Chalasani et al. (1998), Chalasani et al. (1999), and Costabile et al. (2006). The numerical results suggest that our adaptive placement method is significantly superior to these methods in terms of the accuracy, convergence rate, and computational time.

3.1 Comparisons with the Hull and White's model

In this subsection, we present the comparison between the adaptive placement rule and the logarithmically equally-spaced placement rule in the Hull and White's model

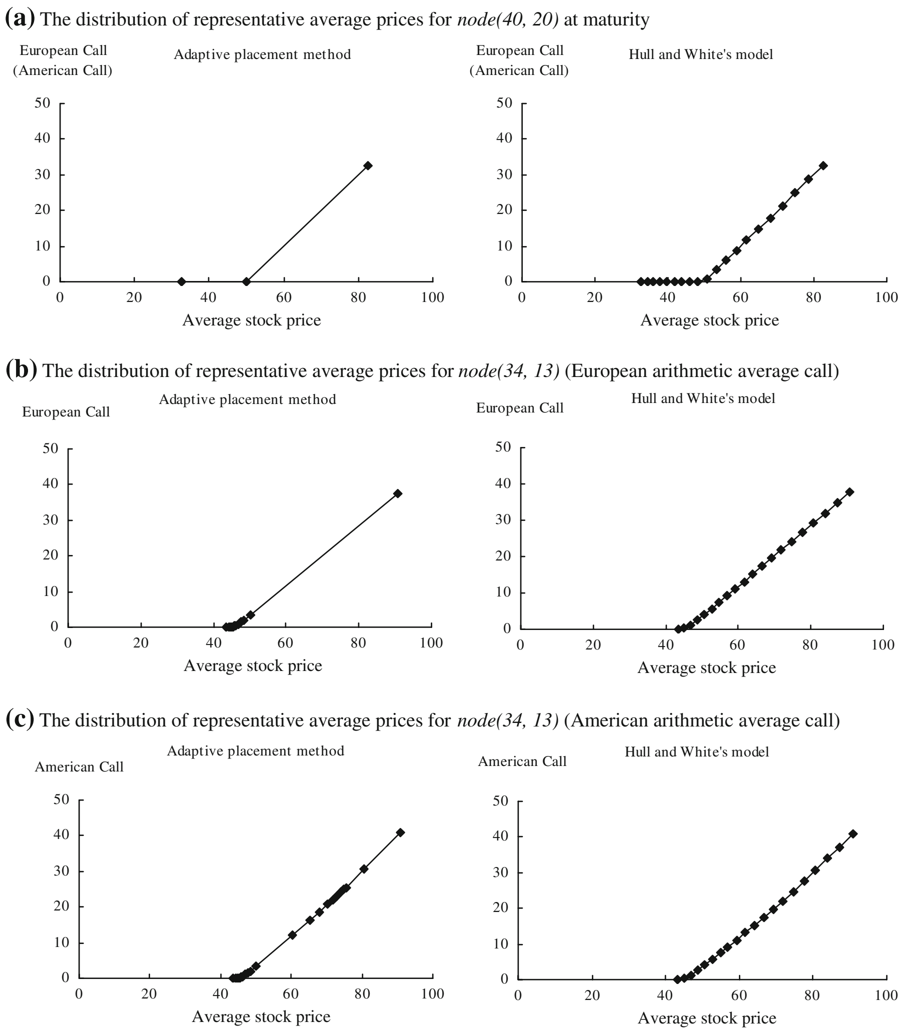

Fig. 5 Comparisons of the distributions of representative average prices in the adaptive placement method and the Hull and White's model. For the readability of this figure, the values of parameters are specified as: $S_0 = X = 50$, $N = 40$, $T = 1$, $r = 10\%$, $\sigma = 30\%$, the *second order error criterion* is 0.01, and the *precision criterion* is 0.001. In addition, the number of representative average prices in the Hull and White's model is 20. According to the results in parts (**a**) and (**b**), it is apparent that our adaptive placement method places more representative average prices in the area with higher degree of curvature, but the Hull and White's model always places representative average prices logarithmically equally-spaced between the maximum and minimum arithmetic average prices. Furthermore, according to the results in parts (**b**) and (**c**), different distributions of representative average prices are employed for American and European arithmetic average options in the adaptive placement method, whereas in the Hull and White's model, the same distribution of representative average prices is used for both American and European arithmetic average options

in Fig. 5. The numerical settings for Fig. 5 are specified as: $S_0 = X = 50$, $N = 40$, $T = 1$, $r = 10\%$, $\sigma = 30\%$, the *second order error criterion* is 0.01, and the *precision criterion* is 0.001. In addition, the number of representative average prices in the Hull and White's model is 20.

In Fig. 5a, for some node at maturity with the strike price between the maximum and minimum arithmetic average prices, it is easily found that there is no linear interpolation error for both linear segments of the payoff function,[8] and therefore it is not necessary to insert any representative average price in our adaptive placement method. However, the Hull and White's model still employs 20 representative average prices for each node at maturity. For each linear segment, our method provides the interpolated results as accurate as those in the Hull and White's model, but around the kink, our method strictly outperforms the Hull and White's model unless the strike price happens to be one of the representative average prices in the Hull and White's model.

In Fig. 5b, it is clear that the logarithmically equally-spaced placement rule in the Hull and White's model places too many representative average prices in the area with low degree of curvature, but only a few representative average prices are needed in our adaptive placement method to derive interpolated results with sufficient accuracy in this area. On the contrary, to deal with the area with high degree of curvature, the Hull and White's model is inclined to generate unexpected large pricing errors, since it disregards the significant magnitude of the nonlinearity error in this area.

In addition, it is our finding that in Fig. 5b, c, under the same value of the *second order error criterion*, the American-style arithmetic average options will need more representative average prices than the European-style ones, while there is no difference for the Hull and White's model to deal with the American and European arithmetic average options. This is because more representative average prices are needed to tackle the area around the early exercise boundary for American arithmetic average option in our model. The more detailed analysis is illustrated in Fig. 6.

Figure 6 shows the relation among the continuation value, the exercise value, the American arithmetic average call value, and the distribution of representative average prices in our adaptive placement method for pricing American arithmetic average calls. If the feature of early exercise is not taken into consideration, representative average prices are mainly placed around the kink, near which the degree of curvature is high. However, for American arithmetic average calls, the degree of curvature is also significant near the early exercise threshold, which is the intersection of the exercise value and the continuation value. Therefore, it needs extra representative average prices to limit the linear interpolation error for adjacent representative average prices near this area. To our knowledge, the adaptive placement method is the first model that automatically employs different placement distributions for pricing American and European arithmetic average options.

[8] If the strike price is not within the range between the maximum and minimum arithmetic average prices for some node at maturity, there are only two representative average prices for this node, and the option payoff is a perfectly linear function. Thus it is not necessary to insert any representative average price between the maximum and minimum arithmetic average prices.

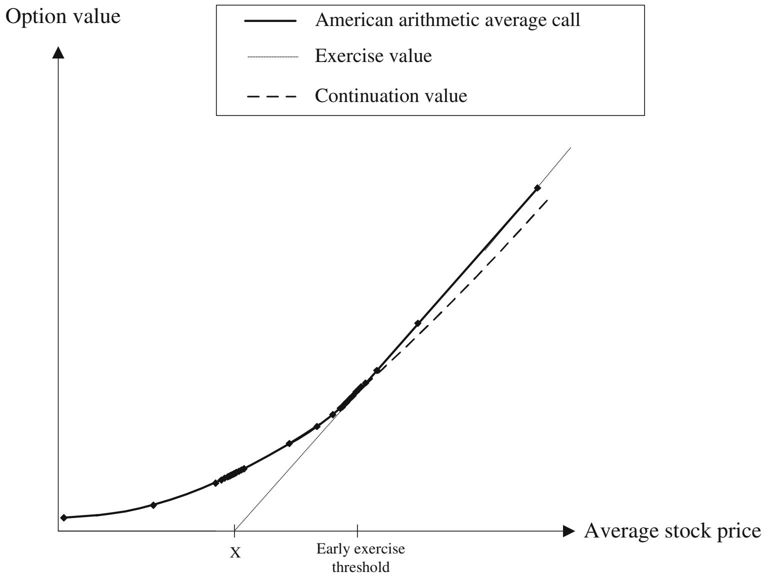

Fig. 6 The distribution of representative average prices for American arithmetic average calls in the adaptive placement method. The dash and dotted lines represent the continuation value and the exercise value, the solid line is the option value of the American arithmetic average call, and the diamonds stand for the representative average prices in our adaptive placement method. Due to the characteristic of early exercise, the option value of the American arithmetic average call is replaced by the exercise value when the average stock price exceeds the early exercise threshold. This replacement causes a kink near this threshold. Therefore our adaptive placement method needs more representative average prices for American arithmetic average calls than for European arithmetic average calls

3.2 Performance comparisons in terms of the interpolation error and the convergence rate

This section is dedicated to compare the performance of examined methods in terms of the magnitude of the interpolation error and the rate of convergence with respect to the number of representative average prices per node. In addition, the effect of the linear extrapolation based on the results of our adaptive placement method is also analyzed. The European and American arithmetic average calls are used as the illustrative examples and the results are listed in Table 1. The values of parameters in our numerical example are as follows: S_0 is 50, X is 50, T is 1 year, r is 10%, σ is 80%,[9] and N is 40.

3.2.1 Analysis of the interpolation error

Panels 1 and 2 in Table 1 report the values of European and American arithmetic average calls, respectively. In addition to the aforementioned parameters for the

[9] It is well known that the performance of some numerical methods will deteriorate with the increasing of the value of σ. Therefore, a large value of σ is examined here to ensure that our adaptive placement method still works well in this situation.

Table 1 The values of European and American arithmetic average calls

Panel 1: European arithmetic average calls (The benchmark is 9.684010)												
Average number of representative average prices	10.1	20.0	30.5	40.1	50.4	60.1	70.5	80.0	90.1	100.4	MAE	RMSE
Second order error criterion	0.0024000	0.0004200	0.0001500	0.0000800	0.0000470	0.0000315	0.0000220	0.0000165	0.0000125	0.0000098		
Adaptive placement method	9.70007	9.68696	9.68502	9.68454	9.68432	9.68422	9.68416	9.68412	9.68409	9.68407		
	(0.1659%)	**(0.0304%)**	**(0.0104%)**	**(0.0055%)**	**(0.0032%)**	**(0.0022%)**	**(0.0015%)**	**(0.0011%)**	**(0.0008%)**	**(0.0006%)**	**0.1659%**	**0.0535%**
Average number of representative average prices	9.5	19.2	30.0	39.5	49.8	59.5	69.7	79.4	89.7	99.9		
Second order error criterion	0.0024000	0.0004200	0.0001500	0.0000800	0.0000470	0.0000315	0.0000220	0.0000165	0.0000125	0.0000098		
Adaptive placement method (*AMO*)	9.69977 **(0.1627%)**	9.68696 **(0.0305%)**	9.68504 **(0.0106%)**	9.68455 **(0.0055%)**	9.68432 **(0.0032%)**	9.68422 **(0.0022%)**	9.68416 **(0.0015%)**	9.68412 **(0.0011%)**	9.68409 **(0.0008%)**	9.68407 **(0.0007%)**	**0.1627%**	**0.0525%**
Number of representative average prices	10	20	30	40	50	60	70	80	90	100		
Linear interpolation—Hull	11.61797	10.26071	9.87494	9.78855	9.75759	9.73001	9.71837	9.71014	9.70539	9.70136		
and White's model	(19.9707%)	(5.9551%)	(1.9716%)	(1.0795%)	(0.7598%)	(0.4750%)	(0.3548%)	(0.2698%)	(0.2207%)	(0.1792%)	**19.9707%**	**6.6365%**
Linear interpolation (*AMO*)	10.22038	9.83222	9.74901	9.71916	9.70546	9.69835	9.69432	9.69176	9.69016	9.68900		
	(5.5387%)	(1.5305%)	(0.6712%)	(0.3630%)	(0.2215%)	(0.1480%)	(0.1065%)	(0.0800%)	(0.0635%)	(0.0515%)	**5.5387%**	**1.8357%**
Panel 2: American arithmetic average calls (The benchmark is 11.149997)												
Average number of representative average prices	10.1	20.2	30.0	40.1	50.0	60.1	70.3	80.5	90.1	100.1	MAE	RMSE
Second order error criterion	0.0120000	0.0018000	0.0006500	0.0003100	0.0001780	0.0001120	0.0000760	0.0000540	0.0000406	0.0000310		
Adaptive placement method	11.21956	11.16202	11.15433	11.15204	11.15113	11.15070	11.15048	11.15033	11.15024	11.15018		
	(0.6239%)	**(0.1079%)**	**(0.0389%)**	**(0.0183%)**	**(0.0101%)**	**(0.0063%)**	**(0.0043%)**	**(0.0030%)**	**(0.0022%)**	**(0.0017%)**	**0.6239%**	**0.2007%**

Table 1 continued

Average number of representative average prices	9.6	19.5	29.2	39.2	49.1	59.1	69.1	79.2	89.1	99.1		
Second order error criterion	0.0120000	0.0018000	0.0006500	0.0003100	0.0001780	0.0001120	0.0000760	0.0000540	0.0000406	0.0000310		
Adaptive placement method (*DHL*)	11.22050 **(0.6324%)**	11.16213 **(0.1088%)**	11.15436 **(0.0391%)**	11.15206 **(0.0185%)**	11.15115 **(0.0103%)**	11.15071 **(0.0064%)**	11.15049 **(0.0044%)**	11.15034 **(0.0031%)**	11.15024 **(0.0022%)**	11.15018 **(0.0017%)**	**0.6324%**	**0.2034%**
Number of representative average prices	10	20	30	40	50	60	70	80	90	100		
Linear interpolation—Hull and White's model	13.23643 (18.7124%)	11.63889 (4.3847%)	11.29268 (1.2797%)	11.22382 (0.6621%)	11.20448 (0.4886%)	11.18645 (0.3270%)	11.17488 (0.2232%)	11.16846 (0.1656%)	11.16522 (0.1366%)	11.16249 (0.1120%)	**18.7124%**	**6.0984%**
Linear interpolation (*DHL*)	12.24294 (9.8021%)	11.27877 (1.1549%)	11.20285 (0.4740%)	11.17722 (0.2442%)	11.16835 (0.1646%)	11.16232 (0.1105%)	11.15882 (0.0791%)	11.15676 (0.0606%)	11.15509 (0.0457%)	11.15430 (0.0386%)	**9.8021%**	**3.1265%**

The values of parameters are $S_0 = 50$, $X = 50$, $T = 1$ year, $r = 10\%$, $\sigma = 80\%$, and $N = 40$, and the number of representative average prices is considered to be 10, 20, ..., 100. In addition, the *precision criterion* is fixed to 0.00001, and the *second order error criterion* is adjusted so that the average number of representative average prices per node in the adaptive placement method is comparable to the number of representative average prices employed in other methods. Panels 1 and 2 present the values of European and American arithmetic average calls, respectively. The corresponding relative error ((option value − benchmark)/benchmark) is in the underlying parentheses. The benchmarks of European and American arithmetic average calls are 9.684012 and 11.149998, which are derived from the Hull and White's model with the number of representative average prices to be 10000 and the strike price inserted into the table of representative average prices if it is between the maximum and minimum arithmetic average prices for each node. In the adaptive placement method, the average number of representative average prices per node is the total number of stored representative average prices of all nodes divided by the number of all nodes in the binomial tree. In addition, the abbreviations AMO and DHL stand for the algorithms of Aingworth et al. (2000) and Dai et al. (2002) individually. The results of the MAE (maximum absolute error) and RMSE (root–mean–square error) for the relative errors of all methods clearly demonstrate that our adaptive placement method outperforms the other methods in terms of producing smaller interpolation errors binomial-tree option pricing model, there are two more parameters in our adaptive placement method: the *second order error criterion* and the *precision criterion*. The *precision criterion* is fixed to be 0.00001 and the *second order error criterion* is adjusted so that the average number of representative average prices per node in our adaptive placement method can be comparable with the number of representative average prices per node employed in the other methods. For the adaptive placement method, the average number of representative average prices per node is defined as the total number of representative average prices of all nodes divided by the number of all nodes in the binomial tree. For each reported option value, the corresponding relative error ((option value − benchmark)/benchmark) is listed in the underlying parentheses. The benchmarks are derived from the Hull and White's model with the number of representative average prices to be 10000 and the strike price inserted as a representative average price if it is between the maximum and minimum arithmetic average prices for each node.

The reported values of the MAE (maximum absolute error) and RMSE (root–mean–square error) for the relative errors of all examined methods in Table 1 clearly demonstrate that the adaptive placement method outperforms the other methods and produces smaller interpolation errors under the same number of representative average prices. More specifically, for European arithmetic average calls, the relative error is 0.1659% when the average number of representative average prices per node is 10.1, and the relative error is 0.0006% when the average number of representative average prices per node is 100.4. However, the corresponding relative errors in the Hull and White's model are 19.9707% and 0.1792% for the number of representative average prices being 10 and 100, respectively.

As to the American arithmetic average calls, since more representative average prices are needed near the intersection of the early exercise value and the continuation value, larger values of the *second order error criterion*s than those for European arithmetic average calls are employed to let the average numbers of representative average prices be about 10, 20,…,100. Similar as the results for European arithmetic average calls, our adaptive placement method generates the most accurate option values. With the same number of representative average prices per node, the relative errors of the option values of our method is about 1.5–3% of the magnitude compared with those of the Hull and White's model.

3.2.2 Illustration of the convergence rates with respect to the number of representative average prices

In order to obtain a better understanding of the rates of convergence, an analysis on the relative error and the number of representative average prices is performed. Plots of ln(|relative error|) in relation to the number of representative average prices for the European and American arithmetic average calls are in Fig. 7a, b. In the figure, a steeper slope represents a better convergence rate, and the lower the line is located, the smaller relative error that the underlying method generates under the same number of representative average prices per node. In addition to the methods based on the piece-wise linear interpolation in Table 1, the results of applying the quadratic interpolation.[10] are also reported in Fig. 7a, b.

It is apparent that the Hull and White's model results in the poorest performance. Although the relative error decreases significantly when the algorithms of Aingworth et al. (2000) and Dai et al. (2002) are applied, our adaptive placement method outperforms them in terms of both providing a better convergence rate and generating smaller relative error with the same number of representative average prices per node. For the quadratic interpolation method, although it exhibits better performance of convergence rates than the Hull and White's model, the AMO algorithm, and the DHL algorithm, our adaptive placement method generally performs better than the method of quadratic interpolation as shown in Fig. 7a, b. Furthermore, there is a vital shortcoming for the quadratic interpolation that the option value exhibits the oscillatory convergence to the exact option value. This polynomial wiggle problem limits the application of the quadratic interpolation because it is never sure that placing more representative average prices will generate a more accurate result. Therefore, studying the relative error with respect to the number of representative average prices for the quadratic interpolation method becomes meaningless.

3.2.3 The effect of incorporating the AMO and DHL algorithms

For European fixed-strike-price arithmetic average calls, Aingworth et al. (2000) suggest that the range $[A_{min}(i, j), A_{max}(i, j)]$ can be curtailed to $[\min(A_{min}(i, j), (N+1) X/(i+1)), \min(A_{max}(i, j), (N+1)X/(i+1))]$ and whenever the average price is above the new upper bound, there is a closed-form formula to derive the corresponding expected option value without suffering any interpolation error. For American arithmetic average options, Dai et al. (2002) suggest a two-phase backward induction method to tighten the range of representative average prices for each node. During the first backward induction, the value of $\overline{A}(i, j)$ for each node is determined according to the critical early exercise boundary for the arithmetic average price. While the second backward induction is processed, $[A_{min}(i, j), A_{max}(i, j)]$ is replaced by $[\min(A_{min}(i, j), \overline{A}(i, j)), \min(A_{max}(i, j), \overline{A}(i, j))]$ and once the average price is higher than the new upper bound, the exact option value will be the exercise value.

In both Fig. 7 and Table 1, it is obvious that the algorithms of AMO and DHL improve significantly the convergence rate of the interpolation error of the Hull and White's model, whereas the improvement is minor when combining the AMO and DHL algorithms with our adaptive placement method. Specifically, the average change of the relative error from incorporating the AMO and DHL algorithms into our adaptive placement method is merely 0.0003% and 0.0009% for European and American arithmetic average options. The reason behind this phenomenon is analyzed as follows. The common idea of the AMO and DHL algorithms is to derive the option values for

[10] In this paper, for any average price A, we use the closest three $A(i, j, k)$'s for the quadratic interpolation. More specifically, suppose the average price A is between $A(i, j, k-1)$ and $A(i, j, k)$ and is closer to $A(i, j, k-1)$ than $A(i, j, k)$, we choose the $(A(i, j, k-2), C(i, j, k-2))$, $(A(i, j, k-1), C(i, j, k-1))$ and $(A(i, j, k), C(i, j, k))$ for the quadratic interpolation. Otherwise, $(A(i, j, k-1), C(i, j, k-1))$, $(A(i, j, k), C(i, j, k))$ and $(A(i, j, k+1), C(i, j, k+1))$ are used for the quadratic interpolation.

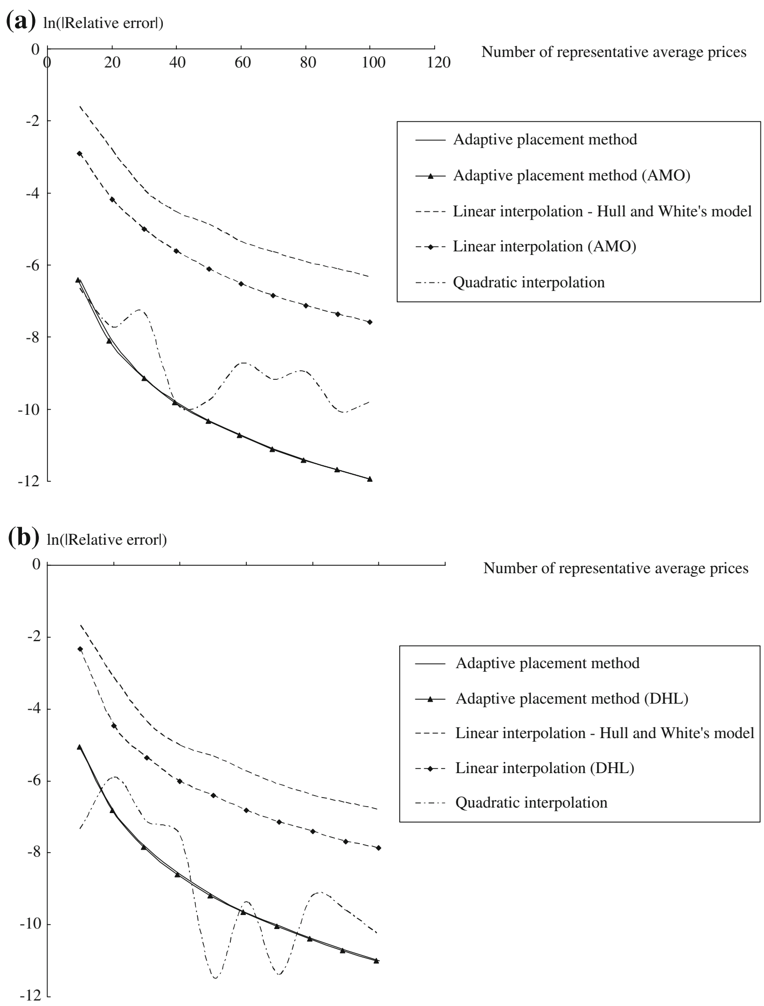

Fig. 7 **a** The rates of convergence of different methods of pricing European arithmetic average calls. The values of parameters in our numerical example are as follows: S_0 is 50, X is 50, T is 1 year, r is 10%, σ is 80%, and N is 40. It is obvious that our adaptive placement method converges faster than the other methods with respect to the number of representative average prices. In addition, the convergence rate of the Hull and White's model can be improved when the AMO algorithm is applied. However, the effect of incorporating the AMO algorithm into our adaptive placement method seems very small. **b** The rates of convergence of different methods of pricing American arithmetic average calls. The numerical settings for this figure are as follows: $S_0 = X = 50$, $T = 1$, $r = 10\%$, $\sigma = 80\%$, and $N = 40$. It is obvious that our adaptive placement method converges faster than the other methods with respect to the number of representative average prices. We also observe that the DHL algorithm does improve the convergence rates of the Hull and White's model, whereas this algorithm almost does not affect the results of our adaptive placement method. Due to the polynomial wiggle problem, the convergence rate of the quadratic interpolation method is so unstable that employing more representative average prices could sometimes derive worse results for the arithmetic average calls

the arithmetic average prices higher than some threshold without incurring any interpolation error and meanwhile concentrate the resource of representative average prices on a smaller range to further reduce the interpolation error. However, for the region above the threshold, the interpolation error is in fact very small. It can be observed in Table 1 that under the same value of the *second order error criterion*, incorporating the AMO or DHL algorithm to tighten the range of the maximum and minimum arithmetic average prices does not save much for the average number of representative average prices per node in our adaptive placement method. This is because our adaptive placement method already places fewer representative average prices in the region above the threshold, and automatically concentrates on dealing with the region below the threshold. According to the results that introducing the AMO and DHL algorithms neither improve the performance nor save the average number of representative average prices per node in our adaptive placement method, it is believed that although our adaptive placement method does not figure out the threshold directly, the concept of the AMO and DHL algorithm is already nested in our adaptive placement method.

3.2.4 Extrapolation with respect to the second order error criterion

According to the description of the adaptive placement method in Sect. 2, one may be interested in a question whether our adaptive placement method can generate theoretically the exact option values for arithmetic average options when the *second order error criterion* approaches zero. Based upon the results for European and American arithmetic average calls of our adaptive placement method in Table 1, the illustrations of the option value in relation to the value of the *second order error criterion* for the European and American arithmetic average calls are in Fig. 8a, b, respectively. Meanwhile, the regressions of the option values on the *second order error criterion* are performed as well.

It is worth noting that the values of R^2 for the regressions for the European and American arithmetic average calls are 0.99992 and 0.999468, respectively. The extremely high R^2 indicates that the option values and the *second order error criterion* are in a perfectly linear relation for our adaptive placement method. This fact gives us an opportunity to derive very accurate approximations for the exact option values by the linear extrapolation under the scenario in which the *second order error criterion* is zero. Hence, the values of the intercept terms can stand for the linearly extrapolated approximation of exact option values and they are 9.684022 and 11.150269 for the European and American arithmetic average calls respectively, which are very close to the benchmarks 9.684010 and 11.149997 in Table 1.

3.3 Analysis of the performance under the same computational time

The analysis in the above subsection shows that the adaptive placement method reduces the interpolation error effectively in the sense that it is able to derive a more accurate result under the same number of representative average prices per node. However, our adaptive placement method costs inevitably more computational time conditional on the same number of representative average prices per node. Here we provide another

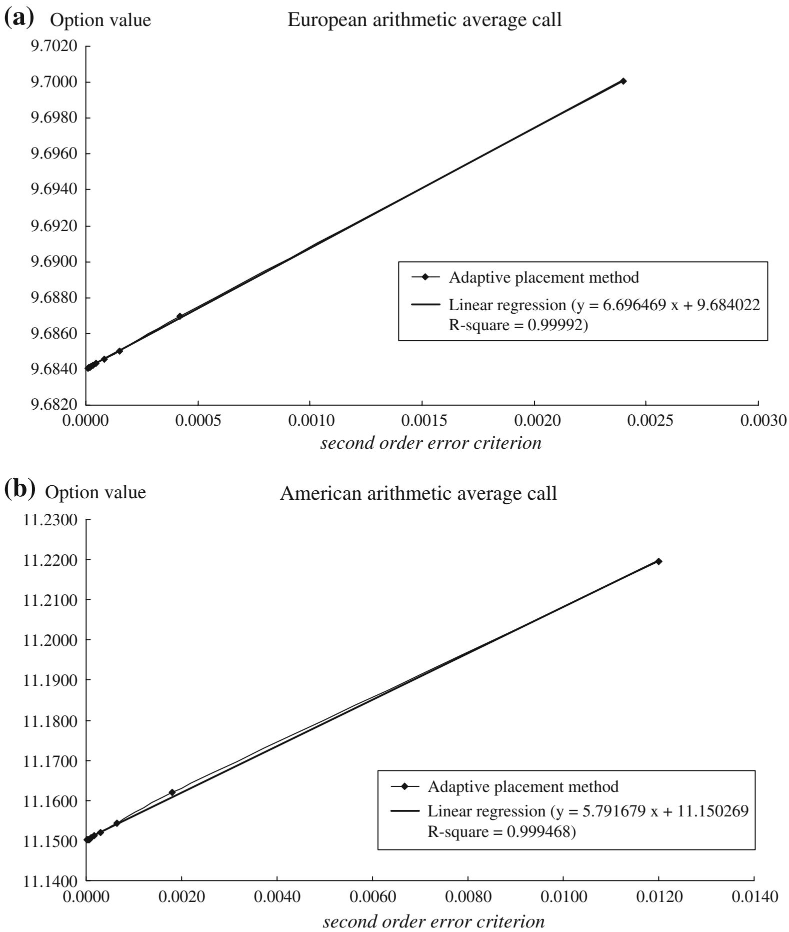

Fig. 8 **a** The linear regression of the option values of European arithmetic average calls on the *second order error criterion*. Based upon the option values for European arithmetic average calls of our adaptive placement method in Table 1, the regression of the option values on the *second order error criterion* is performed. The value of R^2 is extremely high such that the option values and the *second order error criterion* are in a perfectly linear relation for our adaptive placement method. In addition, the intercept term is 9.684010, which is the approximate exact option value in the case of the *second order error criterion* being zero. **b** The linear regression of the option values of American arithmetic average calls on the *second order error criterion*. For the results of American arithmetic average call prices of our adaptive placement method in Table 1, the regression of the option values on the *second order error criterion* is performed. The extremely high value of R^2 means that the option values and the *second order error criterion* are in a perfectly linear relation for our adaptive placement method. In addition, the intercept term is 11.150269, which stands for the approximate exact option value in the case of the *second order error criterion* being zero

measure to demonstrate the superiority of our adaptive placement method by analyzing the computational time and the performance of reducing the interpolation error for each method. Suppose we focus on spending about 4 s to produce the value for an arithmetic average option. The relative errors generated by each method are shown in Table 2. It is obvious that our adaptive placement method still exhibits the smallest relative

Table 2 The analysis of the relative error under the same computational time

	Relative error %	Number of representative average prices	Computational time
Panel 1: European arithmetic average calls under about 4 s computational time			
Adaptive placement method	0.0104	30.5*	4.0
Adaptive placement method (*AMO*)	0.0050	41.4*	3.8
Linear interpolation—Hull and White's model	0.1792	100.0	4.2
Linear interpolation (*AMO*)	0.0159	180.0	4.1
Panel 2: American arithmetic average calls under about 4 s computational time			
Adaptive placement method	0.0389	30.0*	4.0
Adaptive placement method (*DHL*)	0.0257	34.3*	3.9
Linear interpolation—Hull and White's model	0.1120	100.0	4.2
Linear interpolation (*DHL*)	0.0606	80.0	4.4

The values of parameters used in this table are the same as those in Table 1. Under the condition in which the computational time for all examined methods is limited to about 4 s, Panels 1 and 2 present the relative error and the number of representative average prices of each method for European and American arithmetic average calls, respectively. It is apparent that with the same computational time, our adaptive placement method outperforms the Hull and White's model, the AMO algorithm, and the DHL algorithm in terms of generating smaller relative pricing error

* These values are the average numbers of representative average prices per node

error than other methods with the same computational time in both cases of European and American arithmetic average calls. Therefore, we can conclude that even taking both the average number of representative average prices per node and the required computational time into account, our adaptive placement method still exhibits the strongest performance among all examined methods.

3.4 Comparisons with the CJV, CJEV, and CMR algorithms

Figure 9a, b compare the results of our adaptive placement method and those of Chalasani et al. (1998), Chalasani et al. (1999), and Costabile et al. (2006), which have been proven to be efficient on pricing arithmetic average options. Since the above three algorithms do not depend on the number of representative average prices per node, the results of option prices of these three algorithms are all horizontal lines in Fig. 9a, b. It is obvious that due to the rapid convergence rate of our model, the prices of arithmetic average options derived by our model are inside the range between the upper and lower bounds of the CJV and CJEV algorithms as long as the average number of representative average prices per node is large than 25. Although the CMR algorithm does outperform the CJV and CJEV algorithms, our adaptive placement method can generate option prices closer to the benchmark than the CMR algorithm when the average number of representative average prices per node is large than 40. The above facts demonstrate that our adaptive placement method can generate more accurate arithmetic average option prices than these three algorithms without difficulty under a moderate average number of representative average prices per node.

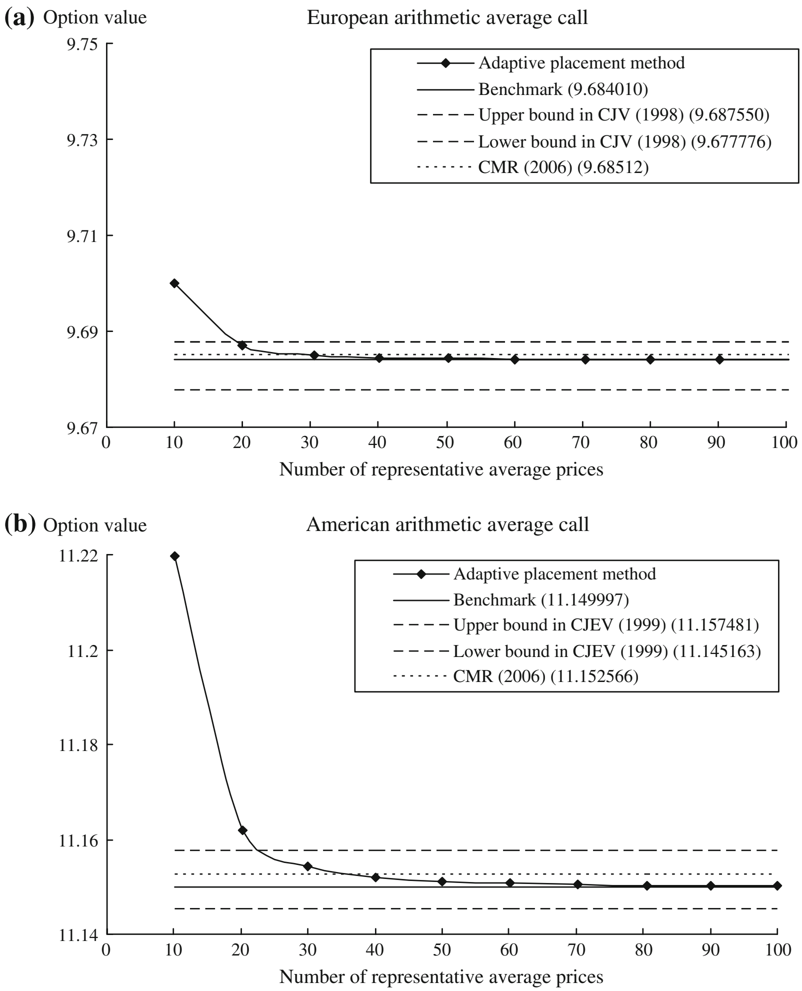

Fig. 9 **a** The comparisons among our method and the algorithms of Chalasani et al. (1998) and Costabile et al. (2006) for European arithmetic average calls. The numerical settings for this figure are as follows: $S_0 = 50$, $X = 50$, $T = 1$, $r = 10\%$, $\sigma = 80\%$, and $N = 40$. For European arithmetic average calls, about 20 representative average prices per node are enough for our method to derive sufficiently accurate option values to be inside the upper and lower bounds of CJV (1998). In addition, as long as the average number of representative average prices per node is larger than 30, our model can generate more accurate option values than CMR (2006). The above results show that our adaptive placement method can generate more accurate European arithmetic average option prices than these two methods with small computational resource. **b** The comparisons among our method and the algorithms of Chalasani et al. (1999) and Costabile et al. (2006) for American arithmetic average calls. The values of parameters in our numerical example are as follows: S_0 is 50, X is 50, T is 1 year, r is 10%, σ is 80%, and N is 40. The American arithmetic average call prices of our adaptive placement model will be inside the range between the upper and lower bounds of CJEV (1999) when the average number of representative average prices per node is larger than 30. Furthermore, for the average number of representative average prices per node larger than 40, the option values derived by our model are more accurate than that of CMR (2006). These facts demonstrate that our adaptive placement method can generate sufficiently accurate American arithmetic average option prices than these two methods with small computational resource

4 Apply adaptive placement method to pricing GARCH options

The adaptive placement method is in fact a general method for all option pricing algorithms with the techniques of augmenting state variables and applying the piece-wise linear interpolation in the backward induction. The GARCH option pricing models in Ritchken and Trevor (1999) and Cakici and Topyan (2000) are typical examples for this kind of algorithm. In this section, we show that the efficiency improvement for pricing GARCH options is still impressive when incorporating the adaptive placement method into the Cakici and Topyan's model, which is a modification of the Ritchken and Trevor's model. In the following subsections, the Ritchken and Trevor's GARCH option pricing model will be stated first. After that, several numerical experiments are conducted to show the performance improvement of the adaptive placement method over the original GARCH option pricing models.

4.1 Ritchken and Trevor's and Cakici and Topyan's models

In Ritchken and Trevor (1999), the stock price is assumed to follow a NGARCH process, and the logarithmic stock price return between day t and $t+1$ under the risk neutral measure is

$$\begin{aligned}\ln\left(\frac{S_{t+1}}{S_t}\right) &= (r - h_t/2) + \sqrt{h_t}\varepsilon_{t+1},\\ h_{t+1} &= \beta_0 + \beta_1 h_t + \beta_2 h_t(\varepsilon_{t+1} - c^\star)^2,\end{aligned} \tag{5}$$

where r is the constant daily risk free return, h_t and ε_{t+1} are the daily conditional variance and a standard normal random variable given the information at t, and the time step is fixed to be 1 day. In addition, β_0, β_1, and β_2 are nonnegative GARCH parameters.

For the option pricing framework in Ritchken and Trevor (1999), a lattice space for logarithmic stock prices $y_t = \ln(S_t)$ is established by introducing a constant gap between adjacent logarithmic stock prices $\gamma_n = \gamma/\sqrt{n}$. The parameter γ is defined as $\sqrt{h_0}$, where h_0 is the initial conditional variance, and n is a parameter for the lattice model so that there are $2n+1$ discrete points to approximate the distribution of the next day's logarithmic stock price, that is

$$y_{t+1} = y_t + \theta\eta\gamma_n, \text{ for } \theta = 0, \pm 1, \pm 2, \ldots, \pm n,$$

where η is a jump parameter to ensure that the probabilities over the $2n+1$ logarithmic stock prices are feasible, and it is defined as a positive integer satisfying $\eta - 1 < \sqrt{h_t}/\gamma \leq \eta$. Consequently, given the information of y_t and h_t, the realized value of ε_{t+1} corresponding to possible y_{t+1}'s can be calculated via

$$\varepsilon_{t+1} = (\theta\eta\gamma_n - (r - h_t/2))/\sqrt{h_t}. \tag{6}$$

By substituting Eq. 6 into Eq. 5, the evolution of the conditional variance over the $2n+1$ logarithmic stock prices can be represented as

$$h_{t+1} = \beta_0 + \beta_1 h_t + \beta_2 h_t \left(\frac{\theta\eta\gamma_n - (r - h_t/2)}{\sqrt{h_t}} - c^\star \right)^2, \text{ for } \theta = 0, \pm 1, \pm 2, \ldots, \pm n. \tag{7}$$

The probability for each logarithmic stock price y_{t+1}, conditional on the information of y_t and h_t, is given as $\text{Prob}(y_{t+1} = y_t + \theta\eta\gamma_n) = P(\theta)$, for $\theta = 0, \pm 1, \pm 2, \ldots, \pm n$, where

$$P(\theta) = \sum_{j_u, j_m, j_d} \binom{n}{j_u\, j_m\, j_d} p_u^{j_u} p_m^{j_m} p_d^{j_d}$$

with nonnegative integer j_u, j_m, and j_d satisfying $n = j_u + j_m + j_d$ and $\theta = j_u - j_d$. Matching the mean and variance of the logarithmic stock price, the formulae of p_u, p_m, and p_d can be derived as follows.

$$\begin{aligned} p_u &= \frac{h_t}{2\eta^2\gamma^2} + \frac{(r - h_t/2)\sqrt{1/n}}{2\eta\gamma}, \\ p_m &= 1 - \frac{h_t}{\eta^2\gamma^2}, \\ p_d &= \frac{h_t}{2\eta^2\gamma^2} - \frac{(r - h_t/2)\sqrt{1/n}}{2\eta\gamma}. \end{aligned}$$

Based upon the above lattice model, it is possible to derive the exact values of GARCH options through a naive pricing method by recording all possible conditional variances reaching each node. However, due to the exponential growth of the number of variance paths with respect to the time step N (equivalent to the number of days to maturity) in the Ritchken and Trevor's model, the naive pricing method is intractable even for a small number of N. In Ritchken and Trevor (1999), instead of keeping track of all possible conditional variances at each node, M interpolated representative conditional variances are equally-spaced placed from the maximum to minimum conditional variances reaching each node. Specifically, for $node(i, j)$ with the logarithmic stock price $y(i, j)$, the table of representative conditional variances are constructed as follows.

$$h(i, j, k) = \frac{M - k}{M - 1} h_{max}(i, j) + \frac{k - 1}{M - 1} h_{min}(i, j), \text{ for } k = 1, \ldots, M,$$

where $h_{max}(i, j)$ and $h_{min}(i, j)$ denote the maximum and minimum conditional variances reaching $node(i, j)$.

Once the lattice model for the underlying asset price and the variance table for each node have been constructed during the lattice-building process, the standard backward induction procedure is applied to calculate option values. According to Eq. 7, for each $h(i, j, k)$, the evolutions of the conditional variance on the next day are

$$h^{next}(\theta) = \beta_0 + \beta_1 h(i, j, k) + \beta_2 h(i, j, k) \left(\frac{\theta \eta \gamma_n - (r - h(i, j, k)/2)}{\sqrt{h(i, j, k)}} - c^\star \right)^2, \quad \text{for } \theta = 0, \pm 1, \pm 2, \ldots, \pm n.$$

Suppose that $h^{next}(\theta)$ is inside the range $[h(i+1, j+\theta\eta, k_\theta), h(i+1, j+\theta\eta, k_\theta - 1)]$. By the linear interpolation method in Eq. 1, the option value C_θ for the conditional variance $h^{next}(\theta)$ can be approximated as $C_\theta = w_\theta C(i+1, j+\theta\eta, k_\theta) + (1 - w_\theta) C(i+1, j+\theta\eta, k_\theta - 1)$, where $w_\theta = (h(i+1, j+\theta\eta, k_\theta - 1) - h^{next}(\theta))/(h(i+1, j+\theta\eta, k_\theta - 1) - h(i+1, j+\theta\eta, k_\theta))$.

Finally, the continuation value for each $h(i, j, k)$ is

$$C(i, j, k) = e^{-r} \sum_{\theta=-n}^{n} P(\theta) C_\theta. \tag{8}$$

If the feature of early exercise is taken into account, taking vanilla put options as examples, the option value corresponding to $h(i, j, k)$ becomes

$$\max(C(i, j, k), X - e^{y(i,j)}). \tag{9}$$

Following the above procedure for all $h(i, j, k)$'s backward over the lattice model, the value of $C(0, 0, 1)$ will be the GARCH option price derived by Ritchken and Trevor (1999).

Cakici and Topyan (2000) argue that each node's maximum and minimum conditional variances are almost contributed from the maximum and minimum conditional variances of predecessor nodes. Therefore, they suggest tracking the evolution of the maximum and minimum conditional variances reaching each node instead of the evolution of all representative conditional variances at each node during the lattice-building process. Representative conditional variances are only employed during the backward induction phase. However, with this modification, it is possible to have some $h^{next}(\theta)$ outside the range $[h(i+1, j+\theta\eta, M), h(i+1, j+\theta\eta, 1]$ during the phase of backward induction. If this situation occurs, they suggest to use the option price corresponding to the minimum (or maximum) conditional variance for the option value of $h^{next}(\theta)$.

4.2 Integrate the adaptive placement method in Cakici and Topyan's model

It is straightforward to integrate the adaptive placement method into the Cakici and Topyan's model. First, after the lattice-building process in Cakici and Topyan (2000), if the initial conditional variance h_0 is between the maximum and minimum conditional variances for some node, h_0 is inserted into the conditional variance table for that node. As a consequence, there will be two or three representative conditional variances for each node after this procedure. Besides, for the nodes at the first four points in time, the above rule is not used and we simply record all possible conditional variances for these nodes during the lattice-building process.

During the backward induction, for each node, the goal of adaptively placing representative conditional variances is to let the range between each pair of adjacent representative conditional variances be small enough such that the linear interpolation error throughout the range is smaller than the *second order error criterion*. For any $node(i, j)$, the linear interpolation error for each pair of adjacent representative conditional variances $[h(i, j, k), h(i, j, k+1)]$ is estimated by **Function *Error Term of the Linear Interpolation*** with the inputs of $(h(i, j, k), C(i, j, k))$, $(h = (h(i, j, k) + h(i, j, k+1))/2, C_h)$, and $(h(i, j, k+1), C(i, j, k+1))$.[11] If it is smaller than the *second order error criterion*, the linear interpolation error for the range $[h(i, j, k), h(i, j, k+1)]$ is considered to be small enough and no further processing will be conducted. Otherwise, the pair (h, C_h) is inserted into the table of representative conditional variances of $node(i, j)$, and the subsets $[h(i, j, k), h]$ and $[h, h(i, j, k+1)]$ are examined separately to check whether the linear interpolation errors inside them are small enough. The above examining-and-dividing process is repeated until the linear interpolation error for every pair of adjacent representative conditional variances is less than the *second order error criterion*. As to the nodes at the first four points in time, this adaptive placement process is not necessary, and only the corresponding option values for all conditional variances of each node are derived through Eqs. 8 and 9. After completing the above backward induction procedure, the GARCH option value is in $C(0, 0, 1)$.

4.3 Analysis of the interpolation error and the convergence rate

Following the numerical example in Ritchken and Trevor (1999), we assume $S_0 = 100$, $X = 100$, the annual risk free rate is 10%, $T = 10$ days, and $N = 10$. In addition, the NGARCH parameters are $\beta_0 = 6.575\text{E}{-06}$, $\beta_1 = 0.9$, $\beta_2 = 0.04$, $c^\star = 0$, and the initial daily variance $h_0 = 0.0001096$, equivalent to an annualized volatility of 20%. As for the lattice model parameter n, the case of $n = 1$ is examined in this paper. It is straightforward to apply the adaptive placement method to the Ritchken and Trevor's and Cakici and Topyan's models for T longer than 10 days and n larger than 1. The reason of merely considering the case of $T = 10$ days and $n = 1$ is because we can employ the naive pricing method, which keeps track of all possible conditional variances reaching each node, to derive exact GARCH option values without any interpolation error to be benchmarks for comparison in this case.

The values of European and American GARCH puts is reported in Table 3. In addition to the aforementioned parameters of the Ritchken and Trevor's GARCH option pricing model, in our adaptive placement method, the *precision criterion* is fixed to be 1.00E−10, and the examined values of *second order error criterion* are 1.00E−02, 1.00E−03, ..., 1.00E−10. For the adaptive placement method, the average number of representative conditional variances per node is defined as the total

[11] We derive the corresponding option values $C(i, j, k)$, C_h, and $C(i, j, k+1)$ following the Ritchken and Trevor's model, i.e. via Eqs. 8 and 9.

Table 3 The values of European and American GARCH puts

Panel 1: European GARCH puts (The benchmark is 1.1977462)											
Average number of representative conditional variances	5.9	22.9	43.9	54.0	72.6	122.4	293.3	842.5	2444.5	MAE	RMSE
Second order error criterion	1.00E−02	1.00E−03	1.00E−04	1.00E−05	1.00E−06	1.00E−07	1.00E−08	1.00E−09	1.00E−10		
Adaptive placement method	1.1946096	1.1975930	1.1978180	1.1977532	1.1977470	1.1977463	1.1977462	1.1977462	1.1977462		
	(−0.26187%)	**(−0.01279%)**	**(0.00599%)**	**(0.00058%)**	**(0.00007%)**	**(0.00001%)**	**(0.00000%)**	**(0.00000%)**	**(0.00000%)**	**0.2619%**	**0.0874%**
Number of representative conditional variances	6	23	44	54	73	123	294	843	2445		
Ritchken and	1.1715867	1.1772875	1.1964036	1.1942081	1.1973859	1.1951119	1.1971423	1.1970741	1.1975718		
Trevor's model	(−2.18406%)	(−1.70810%)	(−0.11209%)	(−0.29540%)	(−0.03009%)	(−0.21994%)	(−0.05042%))	(−0.05612%)	(−0.01456%)	**2.1841%**	**0.9335%**
Cakici and Topyan's	1.1715888	1.1772871	1.1959526	1.1942114	1.1974719	1.1955277	1.1970461	1.1971520	1.1975454		
model	(−2.18388%)	(−1.70813%)	(−0.14975%)	(−0.29512%)	(−0.02290%)	(−0.18523%)	(−0.05845%)	(−0.04961%)	(−0.01677%)	**2.1839%**	**0.9332%**
Panel 2: American GARCH puts (The benchmark is 1.2202429)											
Average number of representative conditional variances	5.9	16.9	37.6	49.1	66.5	110.5	262.9	768.4	2240.9	MAE	RMSE
Second order error criterion	1.00E−02	1.00E−03	1.00E−04	1.00E−05	1.00E−06	1.00E−07	1.00E−08	1.00E−09	1.00E−10		
Adaptive placement	1.2168270	1.2204853	1.2203160	1.2202491	1.2202435	1.2202429	1.2202429	1.2202429	1.2202429		
method	**(−0.27993%)**	**(0.01987%)**	**(0.00599%)**	**(0.00051%)**	**(0.00005%)**	**(0.00001%)**	**(0.00000%)**	**(0.00000%)**	**(0.00000%)**	**0.2799%**	**0.0936%**

Table 3 continued

Number of representative conditional variances	6	17	38	50	67	111	263	769	2241		
Ritchken and Trevor's model	1.1942903 (−2.12684%)	1.2012994 (−1.55243%)	1.2134564 (−0.55616%)	1.2166733 (−0.29253%)	1.2172445 (−0.24572%)	1.2172066 (−0.24883%)	1.2179495 (−0.18795%))	1.2201150 (−0.01048%)	1.2197917 (−0.03697%)	**2.1268**%	**0.9121**%
Cakici and Topyan's model	1.1942921 (−2.12669%)	1.2152594 (−0.40840%)	1.2134565 (−0.55615%)	1.2166733 (−0.29253%)	1.2176253 (−0.21451%)	1.2174298 (−0.23053%)	1.2180294 (−0.18140%)	1.2202211 (−0.00178%)	1.2198343 (−0.03348%)	**2.1267**%	**0.7614**%

The values of parameters are $S_0 = 100$, $X = 100$, the annual risk free rate is 10%, $T = 10$ days, $N = 10$, and $n = 1$. The NGARCH parameters are $\beta_0 = 6.575\text{E}{-06}$, $\beta_1 = 0.9$, $\beta_2 = 0.04$, $c^\star = 0$, and the initial daily variance $h_0 = 0.0001096$, equivalent to an annualized volatility of 20%. The *precision criterion* is fixed to 1.00E−10, and the examined values of *second order error criterion* are 1.00E−02, 1.00E−03, ..., 1.00E−10. In the adaptive placement method, the average number of representative conditional variances per node is the total number of stored representative conditional variances for all nodes divided by the number of all nodes in the GARCH lattice model. In addition, the number of representative conditional variances per node for the Ritchken and Trevor's and Cakici and Topyan's models are chosen to be comparable with the average number of representative conditional variances per node in the adaptive placement method. Panels 1 and 2 present the values of European and American GARCH puts, respectively. The corresponding relative error ((option value − benchmark)/benchmark) is in the underlying parentheses. Based upon the Ritchken and Trevor's model, by recording all possible conditional variances for each node, the exact values (without interpolation errors) of European and American GARCH puts are derived to be the benchmarks for comparison. The benchmarks of European and American GARCH puts are 1.1977462 and 1.2202429, respectively. The results of the MAE (maximum absolute error) and RMSE (root–mean–square error) for the relative errors of all methods clearly demonstrate that our adaptive placement method outperforms the other methods in terms of producing smaller interpolation errors number of representative conditional variances for all nodes divided by the number of all nodes in the GARCH lattice model. For the Ritchken and Trevor's and Cakici and Topyan's models, the number of representative conditional variances per node is set to be comparable with the average number of representative conditional variances per node employed in our adaptive placement method. The corresponding relative error ((option value − benchmark)/benchmark) is listed in the underlying parentheses for each reported option value. The benchmarks are derived from the naive pricing method, which is based on the Ritchken and Trevor's lattice model and tracks all possible conditional variances reaching each node. The benchmarks are the exact GARCH put values for $T = 10$ days and $n = 1$ without any interpolation error.

The option values derived from the Ritchken and Trevor's and Cakici and Topyan's models are very similar, that is consistent with the result in Cakici and Topyan (2000). Our adaptive placement method apparently generates the most accurate option values. For European GARCH puts, when the average number of representative conditional variances per node is 43.9, the relative pricing error of our adaptive placement method is about 5% of the magnitude of the counterpart results in the Ritchken and Trevor's and Cakici and Topyan's models. Similarly, for American GARCH puts, in the case of the average number of representative conditional variances per node to be 37.6, the relative pricing error of our adaptive placement method is about 1% of the magnitude of the counterpart results in the Ritchken and Trevor's and Cakici and Topyan's models. Finally, the MAE and RMSE of our adaptive placement method is about one-tenth magnitude of those of the Ritchken and Trevor's and Cakici and Topyan's models, that clearly demonstrate that our adaptive placement method outperforms the other methods in terms of producing smaller interpolation errors.

In addition, plots of ln(|relative error|) in relation to the number of representative conditional variances for the European and American GARCH puts are in Fig. 10a, b, respectively. It is apparent that our adaptive placement method significantly outperforms the Ritchken and Trevor's and Cakici and Topyan's models in terms of both providing a better convergence rate and generating smaller relative error with the same number of representative conditional variances per node. Furthermore, there is a shortcoming for the Ritchken and Trevor's and Cakici and Topyan's models that the option values do not converge to the exact option value monotonically. Due to this problem, it is always not sure that employing more representative conditional variances per node (equivalent to consuming more computational resource) can generate more accurate GARCH option values.

Based upon the results for European and American GARCH puts of the adaptive placement method in Table 3, the illustrations of the option value in relation to the value of the *second order error criterion* for the European and American GARCH puts are plotted in Fig. 11a, b, respectively. Meanwhile, the regressions of the option values on the *second order error criterion* are performed as well.[12] It is worth noting that the values of R^2 for the regressions for the European and American GARCH puts are extremely high, that indicates a perfectly linear relation between the option values

[12] Since the results of the adaptive placement method start to converge to the exact option value monotonically when the *second order error criterion* is smaller than 1.00E−04, Fig. 11a, b are based on the results of the *second order error criterion* to be 1.00E−04, 1.00E−05, . . . , 1.00E−10.

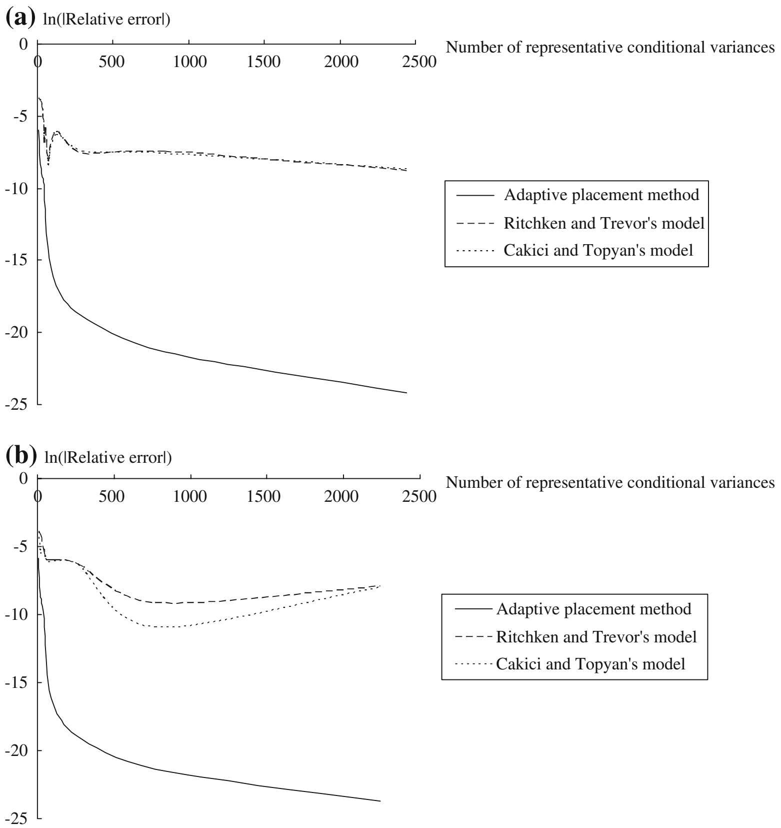

Fig. 10 **a** The rates of convergence of different methods of pricing European GARCH puts. The values of parameters are $S_0 = 100$, $X = 100$, $r = 10\%$ annually, $T = 10$ days, $N = 10$, and the lattice model parameter $n = 1$. In addition, the NGARCH parameters are $\beta_0 = 6.575\text{E}{-}06$, $\beta_1 = 0.9$, $\beta_2 = 0.04$, $c^\star = 0$, and the initial daily variance $h_0 = 0.0001096$. It is obvious that our adaptive placement method exhibits the highest convergence rate than the other methods. Moreover, our adaptive placement method can generate option values convergent to the exact option value monotonically when the average number of representative conditional variances is larger than 40 for this case, whereas the results of Ritchken and Trevor's and Cakici and Topyan's models exhibit the oscillatory convergence to the exact option value. **b** The rates of convergence of different methods of pricing American GARCH calls. The numerical settings for this figure are $S_0 = 100$, $X = 100$, $r = 10\%$ annually, $T = 10$ days, $N = 10$, and the lattice model parameter $n = 1$. In addition, the NGARCH parameters are $\beta_0 = 6.575\text{E}{-}06$, $\beta_1 = 0.9$, $\beta_2 = 0.04$, $c^\star = 0$, and the initial daily variance $h_0 = 0.0001096$. The results indicate that our adaptive placement method shows the strongest performance than the other methods. Moreover, our adaptive placement method can generate option values convergent to the exact option value monotonically when the average number of representative conditional variances is larger than 40 for this case, but the results of Ritchken and Trevor's and Cakici and Topyan's models exhibit the oscillatory convergence to the exact option value. Due to this irregular convergence, it cannot be sure that employing more representative conditional variances could derive more accurate results in Ritchken and Trevor's and Cakici and Topyan's models

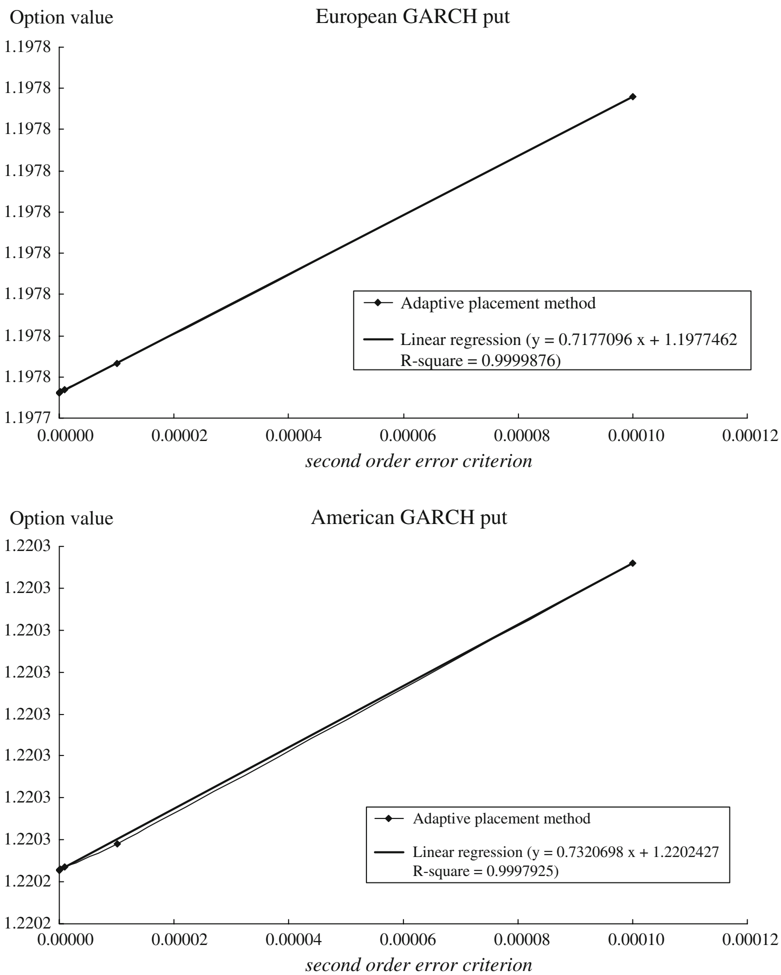

Fig. 11 **a** The linear regression of the option values of European GARCH puts on the *second order error criterion*. Based upon the option values for European GARCH puts of our adaptive placement method in Table 3, the regression of the option values on the *second order error criterion* is performed. Since the characteristic of monotonic convergence emerges when the *second order error criterion* is smaller or equal to 1.00E−04, the regression is based on the option values of the *second order error criterion* being 1.00E−04, 1.00E−05, ..., 1.00E−10. The value of R^2 is extremely high such that the option values and the *second order error criterion* are in a perfectly linear relation for our adaptive placement method. In addition, the intercept term is 1.1977462 for the case of the *second order error criterion* being zero, which is extremely close to the exact value derived by the naive pricing method. **b** The linear regression of the option values of American GARCH puts on the *second order error criterion*. For the results of American GARCH put prices of our adaptive placement method in Table 3, the regression of the option values on the *second order error criterion* is performed. Since the characteristic of monotonic convergence emerges when the *second order error criterion* is smaller or equal to 1.00E−04, the regression is based on the option values of the *second order error criterion* being 1.00E−04, 1.00E−05, ..., 1.00E−10. The extremely high R^2 indicates that the option values and the *second order error criterion* are in a perfectly linear relation for our adaptive placement method. Finally, for the case of the *second order error criterion* being zero, the intercept term is 1.2202427, which is very close to the exact value derived by the naive pricing method

and the *second order error criterion* in our adaptive placement method. This fact gives us an opportunity to derive very accurate estimates for the exact option values by the linear extrapolation under the scenario in which the *second order error criterion* is zero. Since the values of the intercept terms represent the extrapolated option values for the *second order error criterion* equal to zero, they can be viewed as accurate estimates for exact option values. In Fig. 11a, b, the values of the intercept terms are 1.1977462 and 1.2202427 for the European and American GARCH puts, which are extremely close to the benchmarks 1.1977462 and 1.2202429 in Table 3.

5 Conclusion

In this paper, the adaptive placement method is proposed to price arithmetic average options, in which the error of the linear interpolation between each pair of adjacent representative average prices is examined such that representative average prices are placed proportional to the degree of curvature of the option value as the function of the arithmetic average stock price.

From the numerical experiments, our adaptive placement method shows great superiority of reducing the interpolation error over other methods. Even taking the computational time into consideration, the results still suggest that the adaptive placement method is the most efficient one among examined methods. In addition, due to the fact that introducing the AMO and DHL algorithms into the adaptive placement method neither improves the performance nor saves the resource of representative average prices, we believe that the adaptive placement method already incorporates the idea of these two algorithms implicitly. According to the comparisons among our adaptive placement method and the algorithms of CJV, CJEV, and CMR, it is concluded that our adaptive placement method can outperform these three algorithms without difficulty. Finally, it is our finding that a pricing model which can guarantee the convergence of the interpolation error should employ different placement rules of representative average prices in dealing with the European and American arithmetic average options.

Since the adaptive placement method can apply to any numerical algorithm with the techniques of augmented state variables and the piece-wise linear interpolation approximation, we also demonstrate how to integrate the adaptive placement method into the GARCH option pricing algorithm in Cakici and Topyan (2000). The numerical results again prove the superior performance of the adaptive placement method over the original GARCH option pricing algorithms on generating more accurate option values with less interpolation errors. The great performance improvement for applying the adaptive placement method on the pricing algorithms for arithmetic average options and GARCH options suggests the potential applications of this novel method to a broad class of numerical pricing algorithms for exotic options and complex underlying processes.

References

Aingworth, D., Motwani, R., & Oldham, J. D. (2000). Accurate approximations for Asian options. *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*.

Cakici, N., & Topyan, K. (2000). The GARCH option pricing model: A lattice approach. *Journal of Computational Finance, 3*, 71–85.

Carverhill, A. P., & Clewlow, L. J. (1990). Valuing average rate (Asian) options. *Risk, 3*, 25–29.

Chalasani, P., Jha, S., Egriboyun, F., & Varikooty, A. (1999). A refined binomial lattice for pricing American Asian options. *Review of Derivatives Research, 3*, 85–105.

Chalasani, P., Jha, S., & Varikooty, A. (1998). Accurate approximations for European-style Asian options. *Journal of Computational Finance, 1*, 11–29.

Cho, H. Y., & Lee, H. Y. (1997). A lattice model for pricing geometric and arithmetic average options. *Journal of Financial Engineering, 6*, 179–191.

Costabile, M., Massabo, I., & Russo, E. (2006). An adjusted binomial model for pricing asian options. *Review of Quantitative Finance and Accounting, 27*, 285–296.

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

Dai, T.-S., Huang, G.-S., & Lyuu, Y.-D. (2002). Extremely accurate and efficient algorithms for Asian options with range bounds. *Quantitative Methods in Finance 2002 Conference, Cairns, Australia*.

Dai, T.-S., & Lyuu, Y.-D. (2002). Efficient, exact algorithms for Asian options with multiresolution lattices. *Review of Derivatives Researches, 5*, 181–203.

Faires, J. D., & Burden, R. (2003). *Numerical methods* (3rd ed.). Brooks/Cole Press.

Figlewski, S., & Gao, B. (1999). The adaptive Mesh Model: A new approach to efficient option pricing. *Journal of Financial Economics, 53*, 313–351.

Forsyth, P. A., Vetzal, K. R., & Zvan, R. (2002). Convergence of numerical methods for valuing path-dependent options using interpolation. *Review of Derivatives Researches, 5*, 273–314.

Geman, H., & Yor, M. (1993). Bessel processes, Asian options, and perpetuities. *Mathematical Finance, 3*, 349–375.

Hull, J. C., & White, A. (1993). Efficient procedures for valuing European and American path-dependent options. *Journal of Derivatives, 1*, 21–31.

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

Klassen, T. R. (2001). Simple, fast and flexible pricing of Asian options. *Journal of Computational Finance, 4*, 89–124.

Lin, J., & Ritchken, P. (2006). On pricing derivatives in the presence of auxiliary state variables. *Journal of Derivatives, 14*, 29–46.

Milevsky, M. A., & Posner, S. E. (1998). Asian option, the sum of lognormals and the reciprocal gamma distribution. *Journal of Financial and Quantitative Analysis, 33*, 409–422.

Neave, E. H., & Turnbull, S. M. (1994). Quick solutions for arithmetic average options on a recombining random walk. *Actuarial Approach for Financial Risks, 2*, 717–740.

Ritchken, P., & Trevor, R. (1999). Pricing options under generalized GARCH and stochastic volatility processes. *Journal of Finance, 54*, 377–402.

Turnbull, S. M., & Wakeman, L. M. (1991). A quick algorithm for pricing European average option. *Journal of Financial and Quantitative Analysis, 26*, 377–389.