Review of Derivatives Research, 7, 53–72, 2004

Lean Trees—A General Approach for Improving Performance of Lattice Models for Option Pricing

RAINER BAULE
Institute of Finance and Banking, University of Göttingen, Platz der Göttinger Sieben 5, D-37073 Göttingen, Germany

MARCO WILKENS * Marco.Wilkens@KU-Eichstaett.de
Chair of Finance and Banking, Catholic University of Eichstätt-Ingolstadt, Auf der Schanz 49, D-85049 Ingolstadt, Germany

Abstract. The well-known binomial and trinomial tree models for option pricing are examined from the point of view of numerical efficiency. Common lattices use a large part of time resources for calculations which are almost irrelevant for the solution. To avoid this waste of resources, the tree is reduced to a "lean" form which yields the same order of convergence, but with a reduction of numerical effort. In numerical tests it is shown that the proposed method leads to a significant improvement in real calculation time without loss of accuracy for a broad class of derivatives.

Keywords: Lean trees, lattice models, option pricing, numerical valuation techniques.

JEL classification: G13, C63

Lattice models are in widespread use for the valuation of American-type and exotic options for which no closed-form solutions exist. Their history dates back to the introduction of binomial trees by Cox, Ross, and Rubinstein (1979). Since then, several extensions and improvements on this fundamental approach have been worked out, of which only a small selection can be mentioned here. Boyle (1986, 1988) extended the binomial lattice to a trinomial one, gaining more flexibility for the choice of the parameters and also a better performance. Hull and White (1988) improved the accuracy by transferring control variate techniques from the Monte-Carlo method to the tree framework. Richardson extrapolation was suggested by Geske and Johnson (1984) and also used by Breen (1991), who developed the accelerated binomial model. Broadie and Detemple (1996) introduced several further improvements, particularly for the American put, and analyzed them in comparison to existing methods concerning accuracy and calculation time. Leisen and Reimer (1996) proved an order of convergence for some existing binomial lattices and constructed a new one using a slightly different choice of parameters with doubled order of convergence. Besides these and other works aiming to improve the lattice model itself, many papers have been published dealing with the numerical valuation of special derivatives, e.g., barrier or lookback options, with multivariate trees and with extensions of the framework such as the consideration of varying volatilities for instance.

* Corresponding author.

Furthermore, research has been carried out to modify the structure of the tree. Curran (1995) suggested a method of pruning the tree to avoid unnecessary calculations for American options. Chen and Yang (1999) constructed a universal trinomial lattice in which the parameters vary in time to handle almost arbitrary diffusion processes. Recently, Figlewski and Gao (1999) have proposed a further generalization of lattice methods called the Adaptive Mesh Model, which has the powerful property that the density of the tree is variable. Starting with a relatively coarse mesh, their basic idea is to insert regions with higher resolution into the tree where the behavior of the underlying is crucial. In this paper their new approach is employed, but the other way round: the proposed method starts with an already fine mesh, thinning it out or even cutting it off at regions which have a lesser importance. As these regions most often coincide with stock prices far from the initial value (and other critical values such as the strike price), we suggest concentrating on those nodes of the tree which belong to a range of stock prices around the mean value (the "main body" of the tree). The overall goal is to develop a numerical procedure which can be applied for a wide range of derivatives without the necessity for particular adaptations and which reduces the complexity of numerical calculations with negligible loss of accuracy. It will be shown that the proposed Lean Tree Model satisfies both of these objectives.

In Section 1.1 we present a method of pruning binomial and trinomial trees to avoid vast calculations with little impact. As simple pruning may lead to some inaccuracy, we concentrate on the trinomial lattice in Section 1.2, which is developed into the Lean Tree Model with a coarse mesh in the outer parts. Section 1.3 deals with the asymptotic behavior of the model. It is shown that the same order of convergence can be achieved as in a complete tree with calculation effort reduced from $\mathcal{O}(n^2)$ to $\mathcal{O}(n\sqrt{n}\log n)$.

In Section 2, numerical results are presented in the form of an analysis of the trade-off between computational speed and accuracy in comparison with the conventional model. The approach is first applied to American put options. Afterwards we consider other types of derivatives, particular barrier options, max options and power options. In all cases a significant enhancement of the calculation time can be achieved.

Section 3 is the conclusion.

1. Lean Trees

1.1. Pruning Binomial and Trinomial Trees

We make the common assumptions of an ideal market with continuous trading of the underlying and a constant and flat interest rate structure.[1] Let f_t denote the value of an American-style non-path-dependent derivative on a single stock at time t, let T be its time to maturity from the time of evaluation $t_0 := 0$. The underlying stock price S_t is assumed to follow a geometric Brownian motion with variance rate σ^2. Under risk neutrality, the drift of this process equals $r - q$, where r denotes the risk-free continuously compounded annualized interest rate and q the continuous dividend yield. Thus S_t follows the equation

$$dS_t = (r - q)S_t\,dt + \sigma S_t\,dz, \tag{1}$$

where dz represents a standard Wiener process.

In a lattice model this continuous process is discretized in such a way that the time to expiry T is divided into n equidistant[2] time steps of length $\Delta t = T/n$. For each of these steps the price of the underlying jumps to one out of two (in a binomial lattice) or three (in a trinomial lattice) possible values at the next time step. Let u (for up), d (for down) and s (for straight) denote the factors by which the jumps occur, and p_u, p_d and p_s the corresponding risk-neutral probabilities. Since the assumed distribution of the underlying is lognormal, the lattice is usually based on the logarithm of S_t, so the parameters are chosen in a way that the first central moments of the assumed continuous normal distribution coincide with the modelled discrete binomial or trinomial distributions of $\log(S_{\Delta t}/S_0)$. Therefore, several possible (and reasonable) choices exist—see (Leisen and Reimer, 1996) for an overview concerning the binomial case.

The expected value of the random variable $\log(S_{\Delta t}/S_0)$ equals $(r-q-\sigma^2/2)\Delta t$, so if the factors u and d are chosen to fulfill $ud = e^{2(r-q-\sigma^2/2)\Delta t}$, it is ensured that this expected value at each (even) time step coincides with the stock price at the middle node in the corresponding column of the tree. For the same reason, s is chosen to be $s = e^{(r-q-\sigma^2/2)\Delta t}$ in the trinomial case. Matching of the first central moments yields

$$u = e^{(r-q-\sigma^2/2)\Delta t+\sigma\sqrt{\Delta t}}, \qquad d = e^{(r-q-\sigma^2/2)\Delta t-\sigma\sqrt{\Delta t}}, \tag{2}$$

and $p_u = p_d = 1/2$ for the binomial lattice (see (Jarrow and Rudd, 1983)),

$$u = e^{(r-q-\sigma^2/2)\Delta t+\sigma\sqrt{3\Delta t}}, \qquad d = e^{(r-q-\sigma^2/2)\Delta t-\sigma\sqrt{3\Delta t}}, \tag{3}$$

and $p_u = p_d = 1/6$, $p_s = 2/3$ for the trinomial lattice (see (Figlewski and Gao, 1999)).

To introduce index notation, let $f_{j,i}$ be the value of the derivative in the ith node at time step j, which corresponds to the stock price $S_0u^id^{j-i}$ in the binomial case and $S_0s^id^{j-i}$ $(j \geqslant i)$ or $S_0s^{2j-i}u^{i-j}$ $(j < i)$ in the trinomial case. The valuation procedure of working backwards through the tree is well known: starting at expiry, at each time step the buyer has the choice between prematurely exercising and holding the derivative to the next time step. Thus the value at a single node is the maximum of the payoff from an immediate exercise and the continuation value. The latter is calculated as the risk-neutral expected value in the next time step, discounted by the risk-free rate, that is,

$$\tilde{f}_{j,i} = e^{-r\Delta t}(p_u f_{j+1,i+1} + p_d f_{j+1,i}) \tag{4}$$

or

$$\tilde{f}_{j,i} = e^{-r\Delta t}(p_u f_{j+1,i+2} + p_s f_{j+1,i+1} + p_d f_{j+1,i}), \tag{5}$$

respectively (the tilde indicates the pure continuation value).

During this procedure, a total number of $(n+1)(n+2)/2$ nodes in the binomial tree or $(n+1)^2$ nodes in the trinomial tree has to be visited and evaluated. Looking closer at the associated stock prices, it becomes evident that except those positioned in a certain range around the middle nodes (the "main body"), they are extremely unlikely to be reached by the supposed process for S_t. For a quantitative statement notice that the standard deviation of the binomial distribution at the jth time step equals $\sqrt{p_u\, p_d\, j}$, so for larger values of j more than 99 percent of the mass lies between the inner $3\sqrt{j}$ nodes. But this also means that for each time step j a number of about $j - 3\sqrt{j}$ nodes, being by far the major part for larger values of j, is reached with a probability less than 1 percent. For the trinomial tree a similar estimation holds.

Within this observation lies the key for an acceleration of the method. Why should the lion's share of the pricing effort be wasted on calculations which have little influence on the solution? To avoid this waste, we suggest concentrating on the main body of the tree by simply cutting the tails off to make the tree "lean." In the binomial tree the difference $\log S_{j,i+1} - \log S_{j,i}$ equals $2\sigma\sqrt{\Delta t} = 2\sigma\sqrt{T/n}$, so in the final column the number of $\sqrt{n}/2$ nodes covers one standard deviation of the log stock price, which is $\sigma\sqrt{T}$. For the trinomial tree this number is $\sqrt{n}/\sqrt{3}$. Thus we define the main body as the inner $c\sqrt{n}$ nodes, where c is a constant which depends on the type of the tree and the desired accuracy.

However, applying such brute force to the model cannot go without a snag, which is illustrated in Figure 1 for the binomial case: when the main body is detached, for an evaluation of the marked critical nodes (black circle) the option values in the succeeding time step are needed, but the outer ones of these (white circle) have not been calculated in the step before, since they are out of the main body. So if we want to proceed with

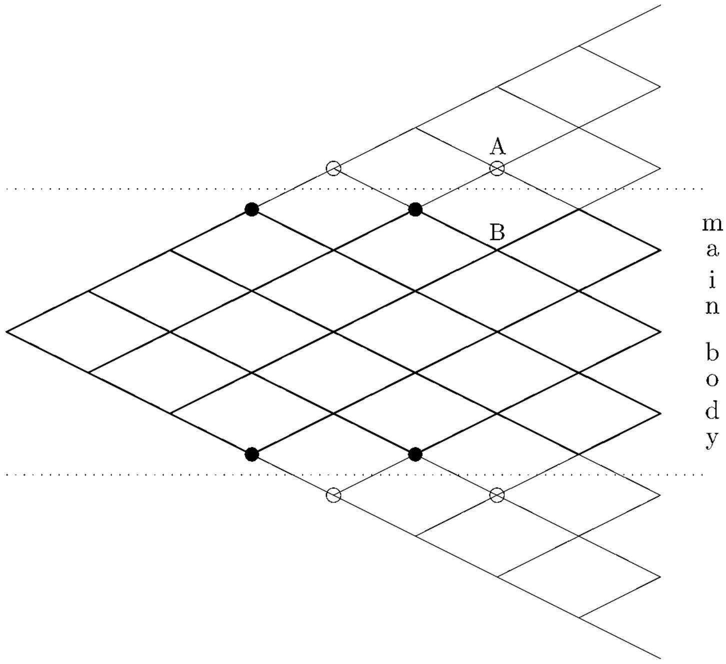

Figure 1. Binomial tree and its reduction to a lean form: The main body consists of the inner nodes. To obtain the continuation value in the marked critical nodes (black circle), estimates for the nodes with white circles have to be carried out.

our method, estimates for the option values in the white nodes must be extracted from the information the tree gives thus far. For this estimation several approaches are feasible.

A solution which suggests itself is an extrapolation method. Conceivable for instance is a linear extrapolation: $f_{j,i} \approx 2f_{j,i-1} - f_{j,i-2}$ (for the upper critical nodes). If a closed-form solution exists for the equivalent European-style derivative, besides more advanced extrapolation schemes as quadratic extrapolation, a conceptionally different approach is applicable: borrowing the idea of control variate techniques (see (Hull and White, 1988)), an estimator for $f_{j,i}$ is $f_{j,i} \approx f_{j,i-1} + f^e_{j,i} - f^e_{j,i-1}$, where the superscript ^{e} stands for European-style. Clearly, the necessity to calculate a Black–Scholes-value for each time step also increases the total calculation time. In Section 2, the question of whether the enhanced accuracy is worth this effort is examined.

1.2. Coarsening the Mesh of Trinomial Trees

Pruning binomial and trinomial trees leads to passable results in certain cases, but is not really satisfying, as the information of the outer parts of the tree is totally neglected. It would be more appropriate to have a procedure which builds a mesh whose density decreases in the outer parts. However, the binomial model leaves no degree of freedom to build such a thin mesh, as with the choice of one of the factors u or d, the requirement to match the first two moments of the normal distribution fixes the other factor.

Thus we concentrate on the trinomial version of the model in this subsection, since it has the desired flexibility to build a coarse mesh for the tails of the distribution. The procedure we suggest is the following (see also Figure 2): The main body ends with the row of critical nodes ($k = 1$). In the two rows directly above and below the main body ($k = 2$) the number of time steps is halved. The successors of the critical nodes at time t then are found at time $t + \Delta t$ or $t + 2\Delta t$, depending on which time step is engaged with a node in the outer row. In the next rows outside the main body this process is iterated, i.e., for each outer row the number of time steps is halved and their length is doubled.

As the number of time steps increases, the factors for an up-move and down-move also have to be adjusted, since otherwise it cannot be guaranteed that the trinomial distribution still matches the first central moments of the normal distribution. Let u_k, s_k, d_k denote the factors in the upper tail of the tree, $p_{u,k}$, $p_{s,k}$, $p_{d,k}$ the corresponding risk-neutral probabilities. We focus on the upper tail; the lower tail is completely analogous. From the suggested iteration it follows that the number of time steps to the next node equals either $l = 2^{k-1}$ or $l = 2^k$.

It can be shown that a recursive definition of the form $u_{k+1} = \sqrt{2}\, u_k$ leads to a consistent mesh where the probabilities are independent of k (see Appendix A). The factors turn out to be

$$u_k = \exp\left(\left(r - q - \frac{\sigma^2}{2}\right) l\Delta t + 2^{k/2}\sigma\sqrt{3\Delta t}\right), \tag{6}$$

$$s_k = \exp\left(\left(r - q - \frac{\sigma^2}{2}\right) l\Delta t\right), \tag{7}$$

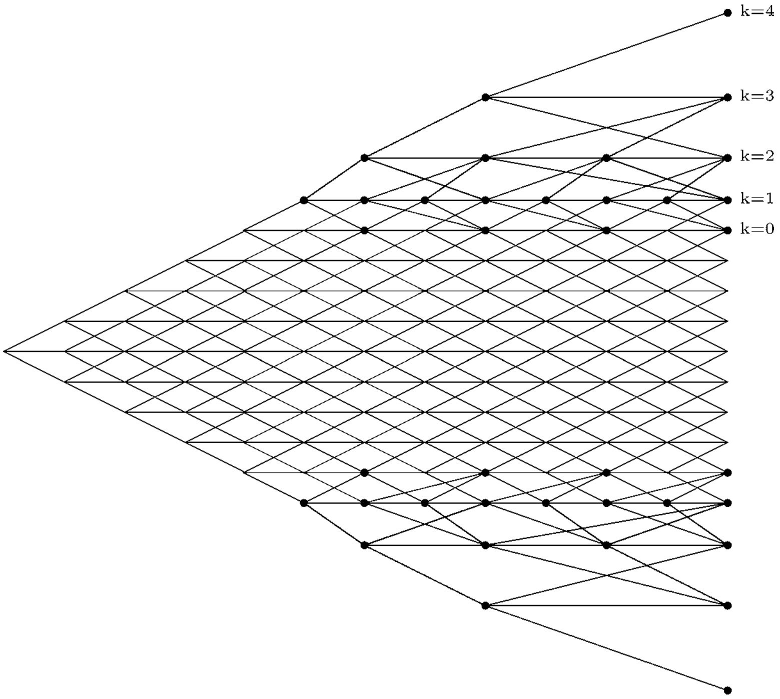

Figure 2. Lean trinomial tree: In the outer parts of the tree the length of the time steps as well as the distance between two nodes is enlarged to build a coarse mesh. (See the text for further explanation.)

$$d_k = \exp\left(\left(r - q - \frac{\sigma^2}{2}\right)l\Delta t - 2^{(k-1)/2}\sigma\sqrt{3\Delta t}\right), \tag{8}$$

the risk-neutral probabilities

$$p_{d,k} = \sqrt{2}p_{u,k}, \qquad p_{s,k} = 1 - p_{u,k} - p_{d,k}, \tag{9}$$

where $p_{u,k}$ is one of the values

$$p_{u,k} = p_u^{(1)} = \frac{1}{3(2+\sqrt{2})} \quad \text{if } l = 2^{k-1}, \tag{10}$$

$$p_{u,k} = p_u^{(0)} = \frac{2}{3(2+\sqrt{2})} \quad \text{if } l = 2^k. \tag{11}$$

In Figure 2 the procedure is demonstrated for $n = 12$ time steps. The tree does not appear very “lean;” this desired property only becomes evident for larger values of n. In Figure 3 a lean tree with $n = 40$ is portrayed, where the attribute “lean” is much more obvious.

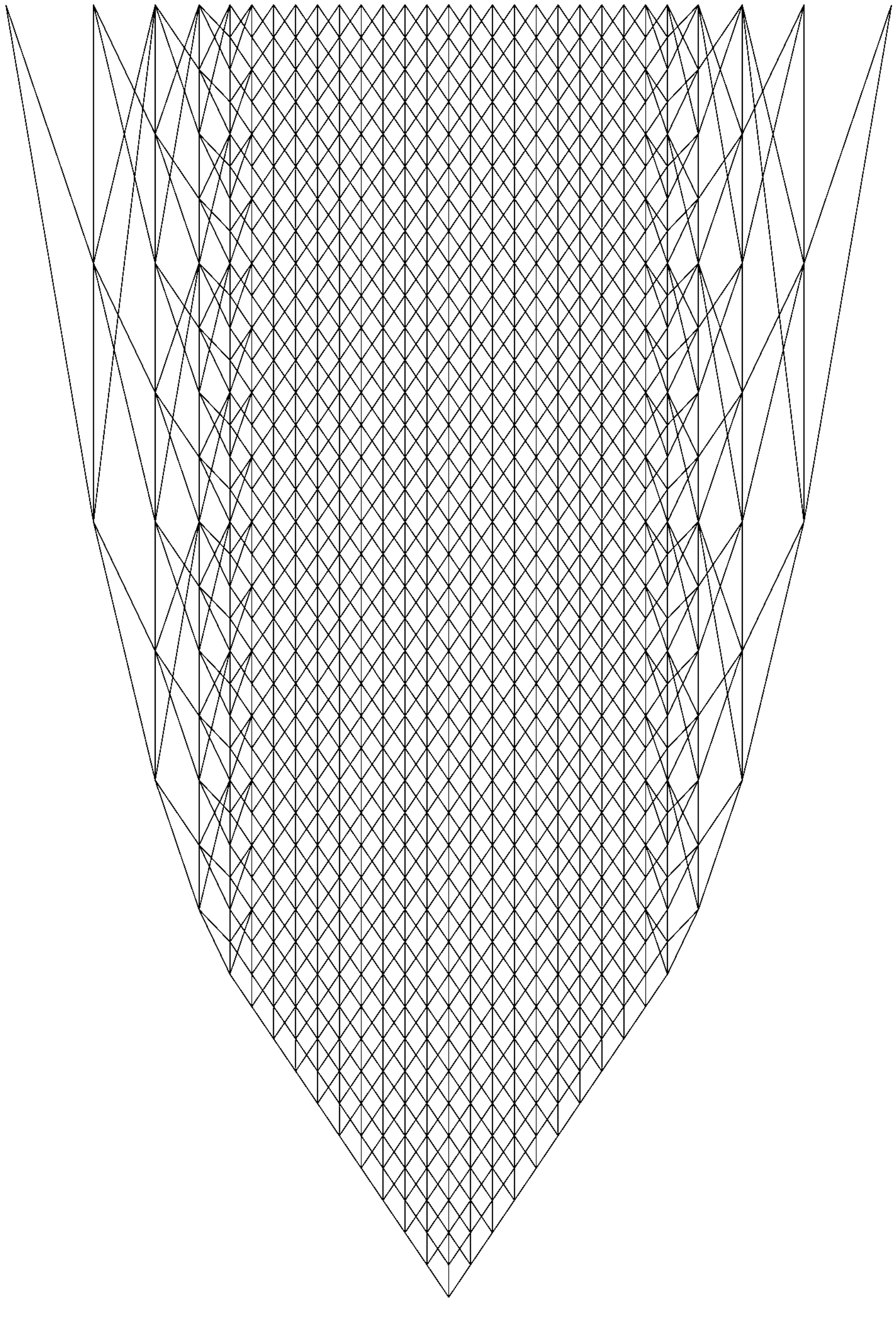

Figure 3. Lean trinomial tree with 40 time steps.

1.3. Asymptotic Behavior

In this subsection a theoretical result concerning the convergence of the option value $\widehat{\hat{f}}_n$ obtained by a lean tree with n time steps against the true value f is presented. It is well known that the option value obtained with a conventional Cox–Ross–Rubinstein tree $\hat{f}_n$ converges to f with order 1, that is, there exists a positive constant a so that (see (Leisen, 1998))

$$\left|f - \hat{f}_n\right| \leqslant an^{-1}. \tag{12}$$

As shown in Appendix B, with a choice of c in dependence on n so that $c \sim \log n$, any desired order of convergence of the lean tree value against the complete tree value can be achieved, particularly the order of convergence of the latter against the true option value. So there exists a further constant a' with

$$\left|\hat{f}_n - \widehat{\hat{f}}_n\right| \leqslant a' n^{-1}, \tag{13}$$

which yields by using the triangle inequality also

$$\left|f - \widehat{\hat{f}}_n\right| \leqslant \left(a + a'\right) n^{-1}, \tag{14}$$

that is convergence of order 1 of the lean tree value against the true value. In summary it can be emphasized that with the right choice of c, the same order of convergence of the lean tree value can be achieved as that of the complete tree value.

This has to be compared with the savings in computational speed, which can be measured by the number of nodes which have to be visited and evaluated. The number of nodes in the main body of the tree is clearly bounded by $c\, n\sqrt{n}$. In the coarse mesh of the outer parts, a number of additional nodes exists, which can be bounded by

$$2 \sum_{k=0}^{[\log_2 n]} 2^k \leqslant 2^{[\log_2 n]+2} = \mathcal{O}(n), \tag{15}$$

so the total number of nodes equals $\mathcal{O}(cn\sqrt{n} + n) = \mathcal{O}(cn\sqrt{n})$. To achieve the desired convergence results, c has to be chosen proportional to $\log n$. Thus the overall costs are $\mathcal{O}(n\sqrt{n}\log n)$, which has to be compared with the costs of $\mathcal{O}(n^2)$ for a conventional tree. It can be stressed that the Lean Tree Model makes an asymptotic improvement on the performance behavior without losing or worsening the convergence property.

Note that the asymptotical convergence is independent of the particular structure of the lean tree; the argumentation remains true if the value in all critical nodes is set to zero. This is because convergence is not achieved by better estimates in the critical nodes, but by a growing share of the main body and thus by a decreasing probability that a critical node is reached. However, good estimates as obtained by the structure of the lean tree still make a great deal of sense, as for practical considerations not only the asymptotic behavior, but the actual errors for usual values of n are of main interest, and these are fairly small as we will demonstrate in the next section.

2. Numerical Results

2.1. American Plain-Vanilla Options

General Sample In this section the behavior of the binomial as well as the trinomial version of the model with regard to computational speed and accuracy is analyzed and compared with the conventional approach. We follow in large parts the method of measuring performance of numerical models proposed by Broadie and Detemple (1996). Therefore, a sample of 2500 sets of random parameters for an American put option has been generated, according to the following restrictions:

- Strike price: fix at 100;
- Initial stock price: uniformly distributed between 70 and 130;
- Time to maturity: with probability 0.75 uniform between 0.1 and 1.0 years; with probability 0.25 uniform between 1 and 5 years;
- Volatility: uniform between 10% and 60%;
- Riskless interest rate: uniform between 0% and 10%.

As error measures we consider the maximum relative error (*MRE*) as well as the root mean squared relative error (*RMSE*), defined as

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\hat{f}_i - f_i}{f_i}\right)^2}, \tag{16}$$

where f_i denotes the ith "true" (obtained with a 20000-step trinomial tree) and $\hat{f}_i$ the ith estimated option value. To make the relative error meaningful, those sets of parameters which lead to a (true) option value lower than 0.5 have been removed, leaving a number of $N = 2326$ options.

In Table 1 the error measures together with the speed for different versions of the model with $n = 1000$ time steps are given. Speed is measured in option prices per second. It becomes evident that the Lean Tree Model saves a factor 5–10 in calculation time. The choice

Table 1. Performance of Several Conventional and Lean Trees with $n = 1000$ Time Steps for American Put Options. Speed Is Measured in Option Prices per Second. The Error Values Refer to the Sample Described in the Text

Model	**Subtype**	***c***	**Speed**	***RMSE***	***MRE***
Binomial			0.503	$2.71 \cdot 10^{-4}$	$3.62 \cdot 10^{-3}$
Lean Binomial	Lin. extrapolation	2.0	4.16	$26.2 \cdot 10^{-4}$	$42.3 \cdot 10^{-3}$
Lean Binomial	Lin. extrapolation	2.5	3.34	$4.00 \cdot 10^{-4}$	$3.62 \cdot 10^{-3}$
Lean Binomial	Control-variate	2.0	2.57	$6.71 \cdot 10^{-4}$	$14.1 \cdot 10^{-3}$
Lean Binomial	Control-variate	2.5	2.24	$2.71 \cdot 10^{-4}$	$3.62 \cdot 10^{-3}$
Trinomial			0.235	$2.01 \cdot 10^{-4}$	$3.86 \cdot 10^{-3}$
Lean Trinomial	Lin. extrapolation	2.0	2.84	$11.4 \cdot 10^{-4}$	$14.2 \cdot 10^{-3}$
Lean Trinomial	Lin. extrapolation	2.5	2.25	$2.81 \cdot 10^{-4}$	$3.86 \cdot 10^{-3}$
Lean Trinomial	Coarse mesh	2.0	2.65	$2.89 \cdot 10^{-4}$	$3.86 \cdot 10^{-3}$
Lean Trinomial	Coarse mesh	2.5	2.16	$2.01 \cdot 10^{-4}$	$3.86 \cdot 10^{-3}$

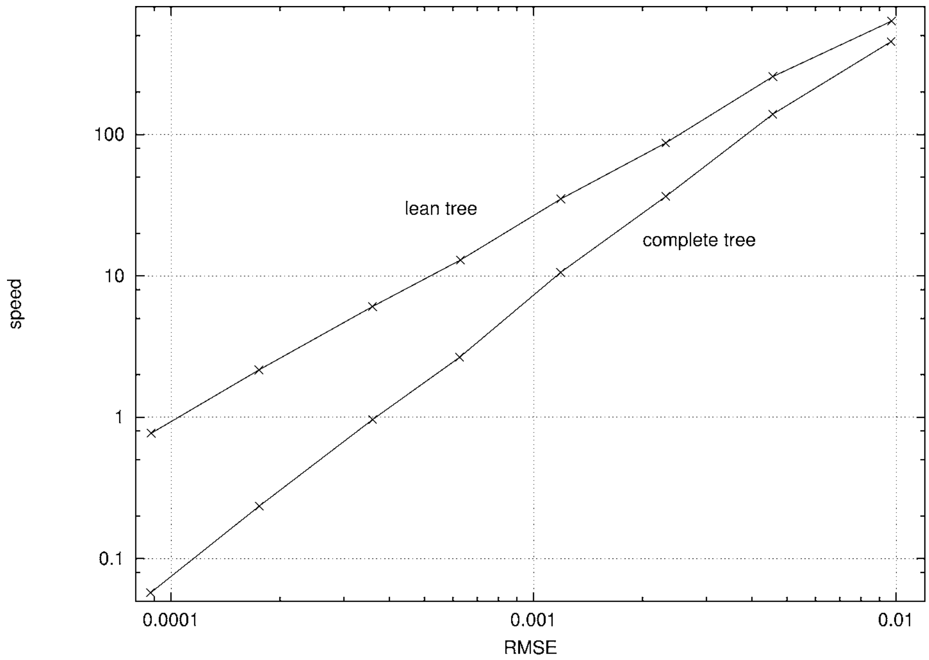

Figure 4. Trade-off between speed and accuracy for the conventional and the lean trinomial model with coarse mesh and $c = 2.5$ for plain vanilla options. Speed is measured in option prices per second. The root mean squared error refers to the sample described in the text. Marks are set for $n = 20, 40, 80, 150, 300, 500, 1000$ and 2000 time steps.

of $c = 2.0$ leads, in the binomial case, to an unacceptable growth in the error, whereas with $c = 2.5$ almost the same error measures can be achieved as in the conventional model. The control-variate-technique seems to be superior to the extrapolation method, although the necessity to calculate Black–Scholes-values in each time step has a significant impact on the calculation time. The reason is that linear extrapolation may lead to negative option values in some nodes. Comparing the two fundamental approaches, the trinomial model outperforms the binomial one. All in all the best model is the trinomial lattice with the coarse mesh.

Concerning the error measure, *MRE* is the same as in the respective complete model for all versions of the lean tree with $c = 2.5$. This guarantees that the average error behavior is a good indicator for the performance of the model, so we will concentrate on the measure *RMSE* in the following.

Figure 4 shows the trade-off between speed and accuracy in a log-log-scale for the conventional and the lean trinomial model with $c = 2.5$. It becomes clear that the prices calculated with the Lean Tree Model have almost the same quality as those with the complete model, but are obtained in much less time. The slope of the lean model curve is significantly smaller than that of the other, which asymptotically equals 2. This means that the order of convergence in terms of *calculation time* rather than number of time steps is

Table 2. Performance of the Lean Trinomial Model with Coarse Mesh and Different c-Values for Deep-Out-of-the-Money Options ($n = 1000$): To Achieve a Satisfying Accuracy, c Has to Be Enlarged to 4.0

Model	***c***	**Speed**	***RMSE***
Trinomial		0.235	$7.97 \cdot 10^{-4}$
Lean Trinomial	2.5	2.16	$207 \cdot 10^{-4}$
Lean Trinomial	3.0	1.93	$81.5 \cdot 10^{-4}$
Lean Trinomial	3.5	1.67	$21.6 \cdot 10^{-4}$
Lean Trinomial	4.0	1.48	$8.34 \cdot 10^{-4}$

enhanced. Nothing else could have been expected according to the analysis of the preceding section; theoretically, the slope should asymptotically be equal 1.5.[3]

Options with Strike Prices Close to the Boundary According to the last subsection, the lean trinomial model with $c = 2.5$ works well in the general case. However, problems might occur when the strike price is close to the boundary of the main body of the tree. In these cases an error behavior which is significantly worse can be expected. To enhance the accuracy one should use larger values for c, that is, a larger part of the main body. Since this adaption also increases the calculation time, a closer look at the performance is necessary.

For this reason a second sample of 2500 options has been created. The parameters are the same as in the preceding subsection, with the exception that the initial stock prices has been chosen so that the strike price ($X = 100$) lies near the lower boundary of the main body with $c = 2.5$. Thus the regarded options are deep out of the money.[4]

The results are shown in Table 2. It becomes evident that the relative errors of the complete tree are larger than in the general sample, which is a consequence of the small absolute values of options which are deep out of the money (the average option value in the sample is as small as 0.20). The errors of the lean tree with $c = 2.5$ are not satisfying. To achieve better results, the parameter c has to be increased to a value of about $c = 4.0$. Clearly this also increases the calculation time, so the time-saving factor is reduced from 10.9 to 6.3. As a conclusion, c should be chosen larger than 2.5 for options which are deep out of the money. With a choice of

$$c = \max\left\{2.5;\, 2.0 + \frac{\log(S_0/X)}{\sigma\sqrt{T}}\right\} \tag{17}$$

the higher speed with $c = 2.5$ is achieved in normal cases, whereas in critical cases (option deep out of the money) the required accuracy is preserved.

2.2. *Path-Dependent Options*

One strength of the model is its generality, which allows the pricing of other than plain-vanilla options. In this section we will show the performance for path-dependent derivatives, particularly American-style barrier options. As an example we choose the down-and-out call.

Pricing barrier option with lattices is a non-trivial task, since the approximated barrier in the tree does not fit the true barrier correctly (see (Boyle and Lau, 1994) for a discussion of this problem). Thus, depending on the number of time steps, the true barrier sometimes lies a little bit above the nearest row of nodes and sometimes a little bit below, which yields a very slow convergence. The option price as a function of the number of time steps has a typical jagged shape (see, e.g., (Boyle and Lau, 1994)).

Several approaches have been carried out to deal with this problem. Boyle and Lau (1994) suggest using only a restricted set of integers for the number of time steps. They calculate a sequence of reasonable values for n so that the approximated barrier is as near as possible to the true barrier. The problem of this sequence is that the smallest value can be quite large if the barrier is close to the strike price. Ritchken (1995) adjusts the parameters of the tree by introducing a stretch parameter so that one row of nodes always coincides with the true barrier H. A similar yet slightly different approach is to adjust only the first time step, whereas in all succeeding steps the values $u = e^{\sigma\sqrt{3\Delta t}}$, $s = 1.0$, and $d = 1/u$ are applied. In the first step these parameters are multiplied with a constant factor $b = H/S_0 u^{i_0}$ with $i_0 = \lfloor \log(S_0/H)/\log(u) + 0.5 \rfloor$.[5] In general, the corresponding risk-neutral probabilities which match the first central moments of $\log S_{\Delta t}$ are given by

$$p_u = \frac{(\sigma^2 \Delta t + s'd')(s' - d')}{u'^2(s' - d') + d'^2(u' - s') - s'^2(u' - d')}, \tag{18}$$

$$p_d = \frac{s' + p_u(u' - s')}{s' - d'}, \tag{19}$$

$$p_s = 1 - p_u - p_d, \tag{20}$$

where

$$u' = \log u - \left(r - q - \frac{\sigma^2}{2}\right)\Delta t, \tag{21}$$

$$s' = \log s - \left(r - q - \frac{\sigma^2}{2}\right)\Delta t, \tag{22}$$

$$d' = \log d - \left(r - q - \frac{\sigma^2}{2}\right)\Delta t. \tag{23}$$

It should be noticed that in contrast to the parameters in Section 1 the middle row of the tree no longer coincides with the expected value of $\log(S_{j\Delta t}/S_0)$. Thus it might be necessary to enlarge the main body by choosing a larger value of c.

Barrier option pricing is particularly critical when the initial stock price is close to the barrier. In these cases also the suggested procedure may lead to poor results, except for large values of n. The reason is that i_0 becomes zero for small values of n, which means that the central row of the tree coincides with the barrier. However, this pitfall can be avoided if i_0 is bounded below by 1. As a consequence, in the first time step only a down-step can lead to a knock-out, which improves the performance enormously. The advantage over Ritchken's approach is that no minimum number of n is required. Nevertheless, a relatively large number of time steps might still be needed to achieve a certain accuracy. Here the

Table 3. Pricing a Down-and-Out Call with a Barrier Close to the Initial Stock Price: The Approximate Number of Time Steps to Achieve a Relative Error of 0.1% or 0.01% Is Given in the Respective Column. The Speed in Option Prices per Second for the Full Trinomial Tree and the Lean Trinomial Tree with Coarse Mesh and $c = 2.5$ Are Compared. The Option Parameters Are *Strike Price* $= 100$, *Barrier* $= 90$, *Volatility* $= 0.3$, *Risk* $-$ *Free Rate* $= 0.05$, *Maturity* $= 1$

		Accuracy 0.1%			**Accuracy 0.01%**		
			Speed			**Speed**	
Stock Price	**Option Value**	***n***	**Full Tree**	**Lean Tree**	***n***	**Full Tree**	**Lean Tree**
95	4.79	50	280	289	300	8.92	26.3
92	1.94	100	78.5	118	400	5.03	17.1
91	0.977	300	8.92	26.3	600	2.24	9.79
90.5	0.490	600	2.24	9.79	1500	0.360	3.05
90.2	0.196	1500	0.360	3.05	5000	0.0322	0.446
90.1	0.0983	2500	0.129	1.25	10000	0.00689	0.164

Lean Tree Model can display its power of moderate increasing calculation time, as Table 3 shows.

2.3. *Multivariate Options*

The convergence analysis in Section 1.3 shows that the Lean Tree Model saves an exponent 0.5 concerning the asymptotic behavior with respect to n, the number of time steps. If the concept of the model is generalized to a multivariate setting, the definition of the main body allows the saving of this exponent 0.5 in each space dimension. Unfortunately the suggested method of coarsening the mesh of the tree cannot be adapted for the multivariate case in a straightforward way.[6] Thus it is more appropriate to use the method of pruning for multidimensional problems.

To demonstrate the behavior in the bivariate case, we have examined the performance of the model for max options.[7] A max call option is a derivative with two underlyings S^1, S^2, which is equivalent to a call on the most valuable of both underlyings. For European-style max options closed-form solutions are given by Stulz (1982). We have chosen to price European-style max options with the conventional and the lean trinomial tree to have this analytic formula as a benchmark. (See (Boyle, 1988; Cho and Lee, 1995) on how to build multivariate trinomial lattices.)

To analyze the behavior, another sample has been generated according to the same parameters as in Section 2.1. The initial stock price of the second underlying S_0^2 is identical to S_0^1, whereas its volatility is independent. Furthermore, the correlation is uniformly distributed between 0 and 1.

The results are shown in Figure 5. Obviously, the slope of the line for the lean tree equals only half the slope of the line for the complete tree. This doubled order of convergence in terms of calculation time could have been expected according to the theoretical analysis. Even for a small number of time steps the savings are significant, and for $n = 500$ the Lean Tree Model outperforms the conventional model by a factor close to 100.

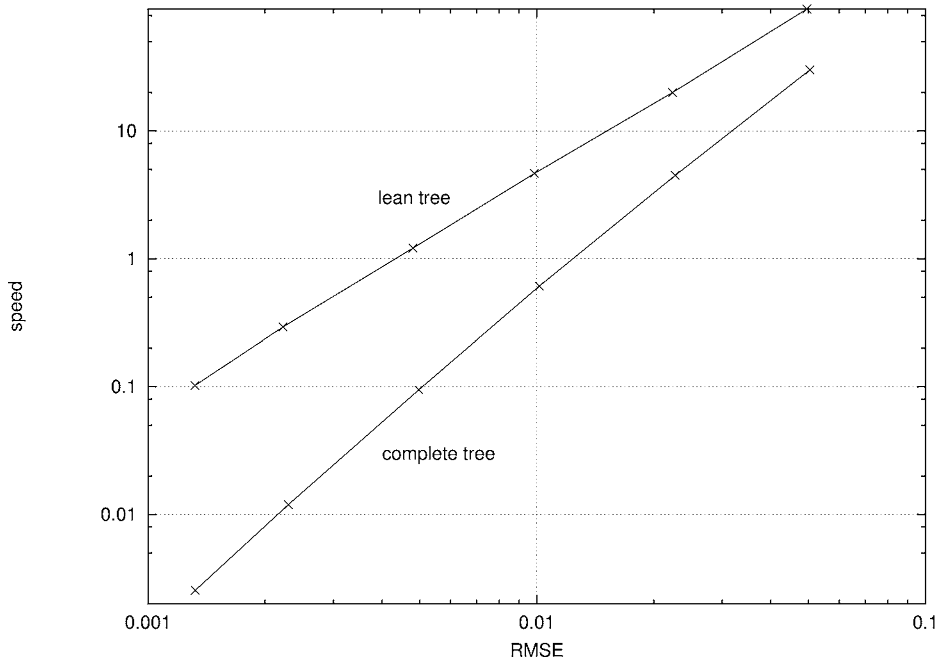

Figure 5. Trade-off between speed and accuracy for the conventional and the lean trinomial model with pruning for max options. Speed is measured in option prices per second. The root mean squared error refers to the sample described in the text. Marks are set for $n = 20, 40, 80, 150, 300$ and 500 time steps.

2.4. Other Exotic Options

The Lean Tree Model can be used to price a wide range of exotic derivatives. As an example we will demonstrate its application to American-style capped power options. The payoff of a capped power option is given by

$$\min\{(S_T - X)^2 1_{\{S_T > X\}}, (Cap - X)^2\}. \tag{24}$$

Since the maximum possible payoff is $(Cap - X)^2$, the option should clearly be exercised if the underlying reaches the cap level.[8] If no dividend payments have to be considered, it is also clear that it should otherwise never be exercised early. Thus the cap level plays the same critical role as the barrier for knock-out options.[9] Indeed, if a power option is priced with a naive Cox–Ross–Rubinstein tree, a similar jagged curve can be observed (see Figure 6).

If we apply our suggested adaptation, that is, multiply the parameters u, s, d in the first time step with the factor $b = H/S_0 u^{-i_0}$ where $i_0 = \lfloor \log(H/S_0)/\log(u) + 0.5 \rfloor$, the convergence behavior can be smoothed dramatically. Using a lean tree preserves this high accuracy, but again leads to a significant reduction in calculation time.

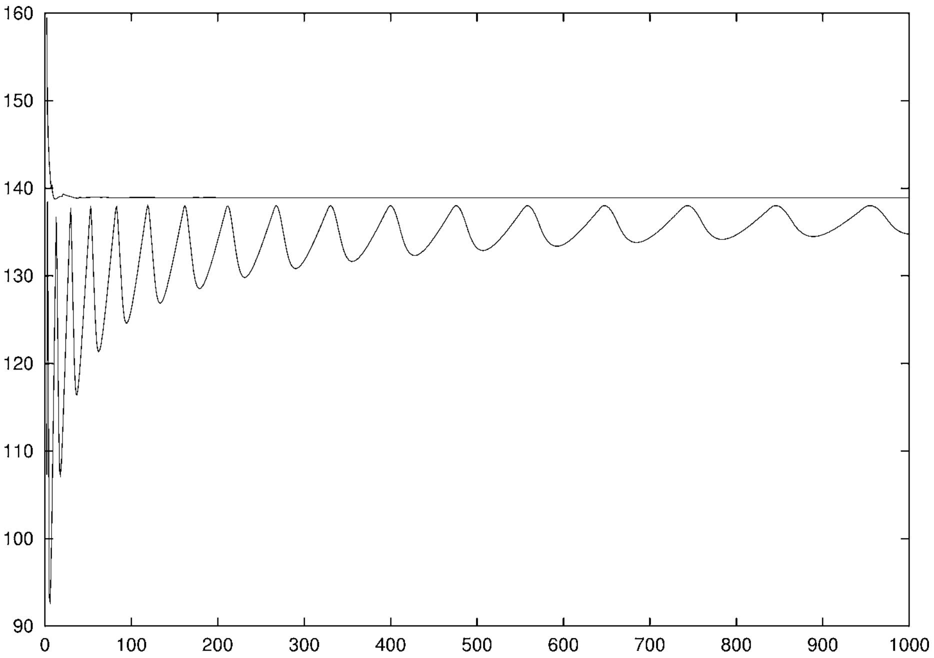

Figure 6. Option price as a function of *n* for a capped power option with a naive trinomial tree (jagged curve) and the adaption described in the text.

3. Summary and Conclusions

Numerical methods have to be applied for option pricing whenever a closed-form solution fails to exist, which is the case for a large share of American-type and exotic options. One fundamental approach is the lattice model, which was originally developed in the context of plain-vanilla options, but can easily be adapted for more complex derivatives. In the present paper, this conventional tree model has been developed into the Lean Tree Model, which achieves the same accuracy in a calculation time decreased from $\mathcal{O}(n^2)$ to $\mathcal{O}(n\sqrt{n}\log n)$.

The convergence property is independent of the particular structure of the lean tree, but the suggested coarse mesh yields the best numerical results. For common values of the number of time steps, the calculation time can be reduced by a factor 10. This factor reduces but is still larger than 5 when options are considered which are deep out of the money.

One strength of the model is its generality, since it can be applied to a wide range of derivatives. With a simple modification of known methods for barrier options, very good results can be obtained even if the initial stock price is close to the barrier. The same modification allows the efficient pricing of capped power options. Furthermore, the use of lean trees ameliorates the exploding calculation time in multivariate settings, since it saves an exponent 0.5 in asymptotic convergence behavior for each space dimension.

Appendix A. Derivation of the Parameters for the Coarse Mesh

In this Appendix the parameters (6)–(11) are derived. We consider the variable

$$X_t := \log(S_t) - \left(r - q - \frac{\sigma^2}{2}\right)t, \tag{A.1}$$

since its expected value equals X_0 for every t. Let $\overline{u}_k$ and $\overline{d}_k$ denote the (additive) values for an up-move and down-move of X_t in the kth upper row of the lean trinomial tree (see Figure 2), $p_{u,k}$ and $p_{d,k}$ the corresponding risk-neutral probabilities (for the straight-move $\overline{s}_k = 0$ holds). Let l be the number of time steps to the next node. From the structure of the coarse mesh it follows that there are two constellations with different parameter sets: $l = 2^{k-\iota}$ with either $\iota = 1$ or $\iota = 0$. We use the existing degree of freedom to pose the condition that in each constellation the probabilities are constant (independent of k), that is $p_{u,k} = p_u^{(\iota)} = const$ and $p_{d,k} = p_d^{(\iota)} = const$.

To ensure moment matching, it has to be guaranteed that

$$E[X_{t+l\Delta t} - X_t] = p_u^{(\iota)}\overline{u}_k + p_d^{(\iota)}\overline{d}_k = 0 \tag{A.2}$$

and

$$E\big[(X_{t+l\Delta t} - X_t)^2\big] = p_u^{(\iota)}\overline{u}_k^2 + p_d^{(\iota)}\overline{d}_k^2 = 2^{k-\iota}\Delta t\sigma^2. \tag{A.3}$$

A necessary condition for recombining is, furthermore,

$$\overline{d}_{k+1} = -\overline{u}_k. \tag{A.4}$$

Our goal is a recursive procedure for $\overline{u}_k$ of the form:

$$\overline{u}_{k+1} = \lambda\overline{u}_k \tag{A.5}$$

with some $\lambda > 1$. Combining (A.4) and (A.5) with (A.2) and (A.3) (with $k+1$ instead of k) yields

$$p_u^{(\iota)}\lambda\overline{u}_k - p_d^{(\iota)}\overline{u}_k = 0 \tag{A.6}$$

and

$$p_u^{(\iota)}\lambda^2\overline{u}_k^2 + p_d^{(\iota)}\overline{u}_k^2 = 2^{k+1-\iota}\Delta t\sigma^2. \tag{A.7}$$

As (A.3) is also valid for $k+2$ instead of k, furthermore,

$$p_u^{(\iota)}\lambda^4\overline{u}_k^2 + p_d^{(\iota)}\lambda^2\overline{u}_k^2 = 2^{k+2-\iota}\Delta t\sigma^2 \tag{A.8}$$

holds. From (A.6) it follows

$$p_d^{(\iota)} = \lambda p_u^{(\iota)}, \tag{A.9}$$

leaving

$$p_d^{(\iota)}\lambda\overline{u}_k^2 + p_d^{(\iota)}\overline{u}_k^2 = 2^{k+1-\iota}\Delta t\sigma^2, \tag{A.10}$$

$$p_d^{(\iota)}\lambda^3\overline{u}_k^2 + p_d^{(\iota)}\lambda^2\overline{u}_k^2 = 2^{k+2-\iota}\Delta t\sigma^2. \tag{A.11}$$

It follows

$$2(\lambda + 1) = \lambda^3 + \lambda^2 \quad \Rightarrow \quad \lambda = \sqrt{2}. \tag{A.12}$$

Since $\overline{u}_0 = \sqrt{3\Delta t}\sigma$, the recursion (A.5) yields

$$\overline{u}_k = 2^{k/2}\sqrt{3\Delta t}\sigma, \tag{A.13}$$

$$\overline{d}_k = -2^{(k-1)/2}\sqrt{3\Delta t}\sigma. \tag{A.14}$$

The probabilities can be calculated using (A.2) und (A.3):

$$p_u^{(\iota)} = \frac{2^{k-\iota}}{3(2^k + \sqrt{2}2^{k-1})} = \frac{2^{1-\iota}}{3(2+\sqrt{2})}, \tag{A.15}$$

$$p_d^{(\iota)} = \frac{\sqrt{2}2^{k-\iota}}{3(2^k + \sqrt{2}2^{k-1})} = \frac{2^{1-\iota}\sqrt{2}}{3(2+\sqrt{2})}. \tag{A.16}$$

Back to the process for S_t, the factors are

$$u_k = \exp\left(\left(r - \frac{\sigma^2}{2}\right)l\Delta t + 2^{k/2}\sigma\sqrt{3\Delta t}\right), \tag{A.17}$$

$$s_k = \exp\left(\left(r - \frac{\sigma^2}{2}\right)l\Delta t\right), \tag{A.18}$$

$$d_k = \exp\left(\left(r - \frac{\sigma^2}{2}\right)l\Delta t - 2^{(k-1)/2}\sigma\sqrt{3\Delta t}\right). \tag{A.19}$$

Appendix B. Convergence of Lean Tree Option Values

Let (j, i) be the index pair of the ith node at time step j, which corresponds to the stock price $S_{j,i} = S_0 u^i d^{j-i}$ in the binomial case and $S_0 s^i d^{j-i}$ $(j \geqslant i)$ or $S_0 s^{2j-i} u^{i-j}$ $(j < i)$ in the trinomial case. We consider an American-style option with payoff function π. Let $\epsilon_{j,i} = |f_{j,i} - \hat{f}_{j,i}|$ denote the absolute difference between the option values in the complete and the lean tree at node (j, i), $q_{j,i}$ the risk-neutral probability that the stock price reaches one of the critical nodes at the boundary of the main body from that node. The aim of the error analysis presented here is a bound for $\epsilon_{0,0}$ in dependence on n, that is $\epsilon_{0,0} \leqslant const \cdot n^{-\alpha}$ with some $\alpha > 0$, which is usually referred to as convergence of order α (of the lean tree value to the complete tree value). Suppose the main body of the tree covers c standard deviations of $\log(S_T/S_0)$ and denote the critical nodes at the boundary with (j, j^+) and (j, j^-), respectively. Define

$$\epsilon := \max_j \left\{\epsilon_{j,j^+}, \epsilon_{j,j^-}\right\} \tag{B.1}$$

as the largest error in a critical node. Then by induction it can be shown that $\epsilon_{j,i} \leqslant q_{j,i}\epsilon$ for all $j \leqslant n$, $j^- \leqslant i \leqslant j^+$: this is trivial for $j = n$. For some $j < n$, it is clear for $i = j^+$

and $i = j^-$, and by induction hypothesis $\epsilon_{j+1,i} \leqslant q_{j+1,i}\epsilon$ for $(j+1)^- \leqslant i \leqslant (j+1)^+$, it follows also for $j^- < i < j^+$:

$$\begin{aligned}\epsilon_{j,i} &= \left|f_{j,i} - \hat{f}_{j,i}\right|\\ &= \left|\max\left\{e^{-r\Delta t}E\left[f_{j+1}|S_{j,i}\right]; \pi(S_{j,i})\right\} - \max\left\{e^{-r\Delta t}E\left[\hat{f}_{j+1}|S_{j,i}\right]; \pi(S_{j,i})\right\}\right|\\ &\leqslant \left|E\left[f_{j+1}|S_{j,i}\right] - E\left[\hat{f}_{j+1}|S_{j,i}\right]\right|\\ &= \left|p_u\left(f_{j+1,i+1} - \hat{f}_{j+1,i+1}\right) + (1-p_u)\left(f_{j+1,i} - \hat{f}_{j+1,i}\right)\right|\\ &\leqslant p_u\epsilon_{j+1,i+1} + (1-p_u)\epsilon_{j+1,i}\\ &\leqslant p_u q_{j+1,i+1}\epsilon + (1-p_u)q_{j+1,i}\epsilon\\ &= q_{j,i}\epsilon \end{aligned} \tag{B.2}$$

in the binomial case and analogously:

$$\begin{aligned}\epsilon_{j,i} &\leqslant p_u\epsilon_{j+1,i+2} + p_s\epsilon_{j+1,i+1} + p_d\epsilon_{j+1,i}\\ &\leqslant p_u q_{j+1,i+2}\epsilon + p_s q_{j+1,i+1}\epsilon + p_d q_{j+1,i}\epsilon = q_{j,i}\epsilon\end{aligned} \tag{B.3}$$

in the trinomial case.

Thus $q_{0,0}$ has to be evaluated. It is the risk-neutral probability that the stock price reaches one of the critical nodes, which can be asymptotically bounded by twice the probability that a standardized Wiener process reaches a barrier $c/2$ in the interval $[0, T]$. This probability can be calculated using the reflection principle (see, e.g., (Karatzas and Shreve, 1991)) as $2N(-c/2)$. Applying the approximation formula for the normal integral (see (Abramowitz and Segun, 1964):

$$N(-x) \leqslant \frac{1}{x\sqrt{2\pi}}e^{-x^2/2} \quad (x > 2.2), \tag{B.4}$$

the error is bounded by

$$\epsilon_{0,0} \leqslant \frac{4}{\sqrt{2\pi}c}e^{-c^2/8}\epsilon. \tag{B.5}$$

If c is now chosen to satisfy $c^2 = 8\alpha\log n$ with a constant $\alpha > 0$, and ϵ is bounded independently of n (which could very roughly be realized with the strike price in the case of a put), it follows

$$\epsilon_{0,0} \leqslant \frac{\epsilon}{\sqrt{\pi\alpha\log n}}n^{-\alpha}, \tag{B.6}$$

that is convergence of order α. Note that even if ϵ depends linearly or polynomially on n, this can be overcompensated by an adequate choice of c.

Notes

1. As shown by several authors, these assumptions can be generalized. The restriction to the root framework is made in this paper, because it aims at numerical treatment rather than a most universal setting. The principles developed here can easily be applied to more general situations.

2. The equidistance is not necessary. In Section 1.2 we will partially use different time steps.
3. The error ϵ is bounded by $\epsilon \leqslant const \cdot n^{-1}$ and the calculation time τ by $\tau \leqslant const \cdot n^{3/2} \log n$. As the logarithm grows slower than every power, for every $\beta > 0$ it follows that the speed in terms of $1/\tau$ is bounded below by $1/\tau \geqslant const \cdot n^{-3/2-\beta} \geqslant const \cdot \epsilon^{3/2+\beta}$ for large values of n.
4. An analogous sample with put options deep in the money is meaningless, since they would be exercised immediately.
5. The approach is very similar to (Hull, 2003, p. 486 f).
6. If the main body is defined for each dimension separately, difficulties occur for nodes which are in the main body for one dimension, but out of it for a different dimension. Similar problems arise when a node is defined as belonging to the main body if it is within the inner nodes for *all* dimensions.
7. A study of quanto options has led to very similar results.
8. For simple capped calls this has already been stressed by Broadie and Detemple (1995).
9. See also (Boyle and Lau, 1994) for pricing simple capped call options with a binomial tree.

Acknowledgements

We thank the editors and two anonymous referees for valuable comments and suggestions.

References

Abramowitz, M. and I.A. Segun. (1964). *Handbook of Mathematical Functions*. New York: Dover.

Boyle, P.P. (1986). "Option Valuation Using a Three-Jump Process," *International Options Journal* 3, 7–12.

Boyle, P.P. (1988). "A Lattice Framework for Option Pricing with Two State Variables," *The Journal of Financial and Quantitative Analysis* 23, 1–12.

Boyle, P.P. and S.H. Lau. (1994). "Bumping up against the Barrier with the Binomial Method," *The Journal of Derivatives* 1, Summer, 6–14.

Breen, R. (1991). "The Accelerated Binomial Option Pricing Model," *The Journal of Financial and Quantitative Analysis* 26, 153–164.

Broadie, M. and J. Detemple. (1995). "American Capped Call Options on Dividend-Paying Assets," *The Review of Financial Studies* 8, 161–191.

Broadie, M. and J. Detemple. (1996). "American Option Valuation: New Bounds, Approximations, and a Comparison of Existing Methods," *The Review of Financial Studies* 9, 1211–1250.

Chen, R.-R. and T.T. Yang. (1999). "A Universal Lattice," *Review of Derivatives Research* 3, 115–133.

Cho, H.Y. and K.W. Lee. (1995). "An Extension of the Three-Jump Process for Contingent Claim Valuation," *The Journal of Derivatives* 3, Fall, 102–108.

Cox, J.C., S.A. Ross, and M. Rubinstein. (1979). "Option Pricing: A Simplified Approach," *Journal of Financial Economics* 7, 229–263.

Curran, M. (1995). "Accelerating American Option Pricing in Lattices," *The Journal of Derivatives* 3, Winter, 8–18.

Figlewski, S. and B. Gao. (1999). "The Adaptive Mesh Model: A New Approach to Efficient Option Pricing," *Journal of Financial Economics* 53, 313–351.

Geske, R. and H.E. Johnson. (1984). "The American Put Option Valued Analytically," *The Journal of Finance* 39, 1511–1524.

Hull, J.C. (2003). *Options, Futures, and Other Derivatives*, 5th ed. Upper Saddle River: Prentice Hall.

Hull, J.C. and A. White. (1988). "The Use of the Control Variate Technique in Option Pricing," *The Journal of Financial and Quantitative Analysis* 23, 237–251.

Jarrow, R. and A. Rudd. (1983). *Option Pricing*. Homewood, IL: Dow Jones-Irwin.

Karatzas, I. and S.E. Shreve. (1991). *Brownian Motion and Stochastic Calculus*, 2nd ed. New York: Springer.

Leisen, D. (1998). "Pricing the American Put Option: A Detailed Convergence Analysis for Binomial Models," *Journal of Economic Dynamics and Control* 22, 1419–1444.

Leisen, D. and M. Reimer. (1996). “Binomial Models for Option Pricing—Examining and Improving Convergence,” *Applied Mathematical Finance* 3, 319–346.

Ritchken, P. (1995). “On Pricing Barrier Options,” *The Journal of Derivatives* 3, Winter, 19–28.

Stulz, R.M. (1982). “Options on the Minimum or the Maximum of Two Risky Assets,” *Journal of Financial Economics* 10, 161–185.