THE CONVERGENCE OF BINOMIAL TREES FOR PRICING THE AMERICAN PUT

MARK S. JOSHI

ABSTRACT. We study 20 different implementation methodologies for each of 11 different choices of parameters of binomial trees and investigate the speed of convergence for pricing American put options numerically. We conclude that the most effective methods involve using truncation, Richardson extrapolation and sometimes smoothing. We do not recommend use of a European as a control. The most effective trees are the Tian third order moment matching tree and a new tree designed to minimize oscillations.

1. INTRODUCTION

There are three main approaches to developing the prices of derivative contracts: Monte Carlo, PDE methods and tree methods. Trees have a conceptual appeal in that they have a natural financial interpretation, are easy to explain and converge in the limit to the Black–Scholes value. They are also well-adapted to the pricing of derivatives with early exercise features. Whilst tree methods can be shown to be special cases of explicit finite difference methods, the fact that when implementing them we are trying to approximate a probability measure rather than a PDE gives rise to different ideas for acceleration and parameter choices.

Whilst it follows from a suitably modified version of the Central Limit theorem that tree prices converge to the Black–Scholes price, one would also like to know in what way the convergence occurs. In addition, one would like to be able to pick the tree in such a way as to accelerate convergence. This problem has been solved for the European call and put options with Diener and Diener, [8], and Walsh, [21], providing detailed analyzes of convergence, and their work was extended by this author, [13], to show that for a given European option, a binomial tree with arbitrarily high order of convergence exists.

Date: March 27, 2009.

1991 *Mathematics Subject Classification.* 91B24, 60G40, 60G44. JEL Classification: G13.

Key words and phrases. binomial trees, Richardson extrapolation, options, rate of convergence.

However, for American options only limited progress has been made. This is an important problem in that trading houses may need to price thousands of contracts for book revaluation and VAR calculations. The crucial issue for such calculations is to find a methodology that achieves a sufficiently accurate price quickly rather than which is asymptotically best. Staunton [18] has examined various methodologies for approximating American put prices including explicit finite differences, implicit finite differences and analytic approximations, as well as trees. He concludes that the Leisen–Reimer tree with the acceleration techniques of extrapolation and truncation is best. However, he does not consider other tree methodologies: the motivation for this tree choice seems to be that the Leisen–Reimer tree is the most effective tree without acceleration techniques and that these make it faster. However, this does not address the possibility that the tree that is best with acceleration techniques may be the worst then they are not used. Our objective here is to find an effective binomial tree by examining many choices of parameters and accelerations in order to find which is fastest.

It is known that for certain trees that the American put option has order 1 convergence, [15] [17], but higher order convergence has not been established for any choice of tree. The only real requirements on a binomial tree are that the mean and variance in the risk-neutral measure are asymptotically correct. This means that there are an infinite number of possible parameter choices which give the same limiting price. This is true even if we require that the tree be self-similar in the sense that for a given number of steps, the possible moves in log-space and the probability of an up-move is the same at all nodes. For example, one can discretize the real-world measure and then pass to the risk-neutral measure and gain a different tree for each choice of the real-world drift. These will all converge to the true price but will differ for any finite number of steps. There are by now a large number of choices of parameters for trees; in this paper, we focus on eleven of these which we believe have the most interesting features: to attempt to do all possibilities would have resulted in an impossibly bloated paper.

There is also the option of using trinomial trees and one can ask similar questions in that case. We defer that work to the sequel [3] where similar conclusions are drawn and, in particular, we see that the best binomial tree found here is better than the best trinomial tree. Of course, there are many other approaches to pricing American options such as implicit and explicit finite difference schemes and analytic approximations. However, Staunton's work [18] suggests that trees are the best available method.

Many suggestions have been made for methodologies for improving convergence for individual trees. The ability to use these is independent of the choice of tree. We now briefly discuss some of the acceleration suggestions that have been made and give a fuller description in Section 3. The first is due to Hull and White, [9]; they suggest using a European priced on the same tree as a *control* variate. i.e. they adjust the American price by the error of the European price.

Broadie and Detemple, [2], suggested two modifications. These were to *smooth* by using the Black–Scholes price on the second-last layer, and to use Richardson extrapolation (RE) to partially remove the lead order term. Broadie and Detemple showed that the two techniques of smoothing and RE together resulted in effective speed-ups for the CRR tree.

Staunton, [18], examined the convergence of binomial trees using *truncation.* In particular, the tree is pruned so that nodes more than 6 standard deviations from the mean in log space are not evaluated. This results in an acceleration since it take less time to develop the tree for a given number of steps, whilst behaviour more than six standard deviations has very little effect on the price. He shows that the Leisen–Reimer tree with Richardson extrapolation and truncation is very effective. Staunton's work followed on from that of Andicropoulos, Widdicks, Duck, and Newton, [1], who had previously suggested curtailing the range of a tree according to distance from mean and strike.

Since all these techniques can be implemented independently, we therefore have 2^4 different ways to improve each binomial tree. In addition, there is a question when using Richardson extrapolation and smoothing together of whether one matches the smoothing times between the small and large numbers of steps. This means that there are a total of 20 different ways to implement each tree.

Furthermore, there is now a large number of different ways to choose the parameters of a binomial tree, depending upon what characteristics one wishes to emphasize. For example, one can attempt to match higher moments, or to obtain smooth convergence, or achieve higher order convergence for a specific European option. We will examine 11 of these choices in this paper.

This results in 220 different ways to price an American put option. It is not at all obvious which will perform best since some trees will perform well in combination with some acceleration techniques and badly with others. In this paper, we perform a comparison of all these methods; we run a large number of options for each case, using a Leisen–Reimer tree with a large number of steps and Richardson extrapolation as a benchmark.

We find that the best choice of tree depends on how one defines error, but that the two best trees are the Tian third moment-matching tree with smoothing, Richardson extrapolation and truncation, and a new tree using a time-dependent drift with extrapolation and truncation.

The structure of binomial trees and our eleven choices of parameters is discussed in Section 2. The different ways these can be accelerated is discussed in Section 3. We present numerical results in Section 4 and conclude in Section 5.

I am grateful to Chris Beveridge, Mark Broadie, Nick Denson, Christopher Merrill, Ken Palmer and Mike Staunton for their comments on an earlier version of this paper.

2. Binomial tree parameters

In this section, we discuss the definition of a tree. A node in a tree is specified by three things:

(1) the probability of an up move p,
(2) the multiplier on the stock price for an up move, u,
(3) the multiplier on the stock price for a down move, d.

We shall say that a tree is *self-similar* if the values of p, u, d are functions of the number of the steps in the tree, n, but not the step number. Thus for a given number of steps, every node is the same in a relative sense. Only one of our choices, the split tree, will not be self-similar.

A sequence of self-similar trees is therefore a specification of p, u and d as a function of the number of steps. Note that any methodology for pricing options on trees is really a choice of a sequence of trees since we need the behaviour for varying n. Thus when we speak of a given tree this is really short-hand for saying that we are specifying such a sequence.

If we require the tree to be risk-neutral then p is determined by u and d via the usual formula

$$p = \frac{e^{r\Delta T} - d}{u - d}, \tag{2.1}$$

with

$$\Delta T = \frac{T}{n}.$$

(Only one of our trees, the Jarrow–Rudd tree, is not risk neutral.) A risk-neutral tree methodology is therefore a pair of sequences u_n and d_n. To keep p between zero and one, we must have

$$d_n < e^{r\Delta T} < u_n. \tag{2.2}$$

We work in the Black–Scholes model with the usual parameters: T is maturity, r is the continuously compounding risk-free rate, S_t is the stock price and σ is the volatility. We can also use μ, the real-world drift, when constructing the tree if we so choose: its choice may affect how convergence occurs although it does not affect the limit.

The choice of u_n and d_n is constrained to ensure that the limiting tree is the Black–Scholes model. Since p_n ensures that the mean is correct, we have one essential condition left: the variances must converge correctly. Since we have two sequences and only one condition, there is still quite a lot of flexibility.

We now discuss our 10 choices of trees that are self-similar. The Cox–Ross–Rubinstein (CRR) tree, [7], is the oldest tree:

$$u_n = e^{\sigma\sqrt{\Delta T}}, \tag{2.3}$$
$$d_n = e^{-\sigma\sqrt{\Delta T}}. \tag{2.4}$$

Note that the dependence of u_n and d_n on n enters via the fact that ΔT is a function of n.

The Tian tree, [19], uses the extra degree of freedom to match the first three moments exactly for all n rather than just the first two in the limit. It takes

$$u_n = \frac{1}{2} r_n v_n \left(v_n + 1 + (v_n^2 + 2v_n - 3)^{\frac{1}{2}} \right), \tag{2.5}$$
$$d_n = \frac{1}{2} r_n v_n \left(v_n + 1 - (v_n^2 + 2v_n - 3)^{\frac{1}{2}} \right), \tag{2.6}$$
$$r_n = e^{r\Delta T}, \tag{2.7}$$
$$v_n = e^{\sigma^2 \Delta T}. \tag{2.8}$$

The Jarrow–Rudd (JR), [10], tree is not a risk-neutral tree and, in fact, seems to be the only non-risk-neutral tree in common use:

$$u_n = e^{\mu\Delta T + \sigma\sqrt{\Delta T}}, \tag{2.9}$$
$$d_n = e^{\mu\Delta T - \sigma\sqrt{\Delta T}}, \tag{2.10}$$
$$\mu = r - \frac{1}{2}\sigma^2, \tag{2.11}$$
$$p_n = \frac{1}{2}. \tag{2.12}$$

A simple modification of the Jarrow–Rudd tree is to take the value of p that makes the tree risk-neutral. We shall refer this to this as the Jarrow–Rudd risk-neutral tree (JRRN). This has also been studied by Jarrow and Turnbull, [11].

It follows from the standard analysis of the binomial tree that one can modify the CRR tree by taking an arbitrary real-world drift μ so

$$u_n = e^{\mu\Delta T + \sigma\sqrt{\Delta T}}, \tag{2.13}$$
$$d_n = e^{\mu\Delta T - \sigma\sqrt{\Delta T}}.$$

(See for example, [12].) One choice is to take $\mu = \frac{1}{T}(\log K - \log S_0)$, thus guaranteeing that the tree is centred on the strike in log space. This was done in [13] and we shall refer to that tree as the *adjusted tree*.

A similar approach has previously been suggested by Tian, [20], who suggested moving the tree slightly so that the strike of the option would land on a node in such a way as to minimize distortion. We shall refer to this as the *flexible* tree.

Chang and Palmer, [5], also suggest a similar tree but make the strike lie half-way between two nodes to obtain smoother convergence for European options. We shall refer to this as the *CP* tree.

Leisen and Reimer, [16], suggested changing point of view to first specifying probabilities of an up move in both stock and bond measures. These two quantities then determine the up and down moves. The probabilities are chosen by using inversions of normal approximations to binomials to get binomial approximations of normals. They suggest three different trees and we will use the one they label (C) here; since that is the one which appears to be in most common use [18]. Their tree had the features of only being defined for odd numbers of steps and being approximately centred on the option strike. This tree is known to have second order convergence for European options, [14].

In [14], a new tree with almost 4th order convergence was introduced by this author. We shall refer to this tree as *J4*. It is only defined for odd numbers of steps. This tree agrees with the Leisen–Reimer (C) tree to order 2.5 in the way the probabilities are specified. Since American options typically have first order convergence, we can expect the two trees to have similar convergence behaviour.

Another choice due to Chriss, [6], is to modify u_n and d_n in the Jarrow–Rudd model. We let

$$X = \frac{2e^{r\Delta T}}{u_n + d_n}$$

and multiply u and d by X. This can be viewed as a symmetrized version of JRRN. The tree is risk-neutral.

Our final tree is the only one that is not self-similar. Our motivation is that whilst it is known that the Leisen–Reimer (C) tree has second order convergence for European options, it can actually perform worse for in-the-money American options [16]. This suggests that there is some odd interaction between the exercise boundary and the tree in the money. We therefore modify the adjusted tree above to use a time-dependent drift. In particular, if the integer part of $n/2$ is k, then we set

$$t_1 = tk/n,$$
$$\mu_1 = \frac{\log K - \log S_0}{t_1},$$
$$\mu_2 = 0,$$

and for the first k steps, we use drift μ_1, and for the rest, we use μ_2. The up and down moves are then defined as in equation (2.13). The idea here is that in the first half we use a strong time-dependence to get the centre of the tree at the same level as strike, and then in the second half, we have no drift. We shall refer to this tree as the *split* tree.

It is worth noting that the trees designed to have smooth and/or higher order convergence have node placement determined by the strike of the option, and for those trees, we therefore have to build a different tree for each option. This is not, however, true for the others including, in particular, the Tian 3rd moment matching tree.

We remark that there are other possible choices and for a review of a different set of 11 choices for pricing European options we refer the reader to [4]. Our choices here were motivated by the desire to include

- higher order convergence for Europeans trees;
- the most popular and oldest trees e.g. CRR, Jarrow–Rudd, and JRRN;
- the theoretically nicest trees, e.g. the higher order moment matching;
- trees with nice lead order terms, e.g. the Chang–Palmer tree, the adjusted tree, and the flexible tree of Tian.

Whilst 10 of our 11 trees have previously been studied, most of them have not been studied in combination with acceleration techniques, so of our 220 trees, we estimate that at least 200 have not previously been examined.

3. The implementation choices

In this section, we list the implementation choices which can be applied to any tree and define a key for our numeric results.

Our first implementation option is *truncation.* We only develop the tree as far as 6 standard deviations from the mean in log-space computed in the risk-neutral measure. At points on the edge of the truncated tree, we take the continuation value to be given by the Black–Scholes formula for a European option. The probability of a greater than six standard deviation move is $1E-9$. The difference between the European and American prices will be slight so far out-of-the money, and so far in-the-money the option will generally be behind the exercise boundary. These facts together mean that truncation has minimal effect on the price: typical effects are around 1E-12. However, for large numbers of steps it can have large effects on speed of implementation since the number of nodes no longer grows quadratically. For small numbers of nodes, it can be slightly slower because of the extra Black–Scholes evaluations. The use of truncation in tree pricing was suggested by Andicropoulos, Widdicks, Duck, and Newton, [1], and refined by Staunton, [18].

We note that the location of the truncation will vary according to volatility and time. There are clearly many other ways to carry out truncation. Our motivation here was to use a methodology that was sure to have minimal impact on price and we have therefore not examined the trade-off between location of the truncation boundary and speed. Nor have we examined the issue of whether it is better to use the intrinsic value at the boundary rather than the Black–Scholes prices. A full analysis would require one to take into account the fact that one can truncate at the edge of a narrower space when using the Black–Scholes price. We leave this issue to future work. Clearly, our analysis is very much tailored to the American put and options with different pay-offs may require a different methodology.

Our second implementation option is *control variates.* Given a binomial tree, one prices both the American put and the European put. If P_A is the tree price of the American put, P_E that of the European and P_{BS} that given by the Black–Scholes formula, we take the error controlled price to be

$$\hat{P}_A = P_A + P_{BS} - P_E.$$

Note that we can expect this to perform well when the European price is poor, but that the error will change little when it is good. It does, however, take a substantial amount of extra computational time. In particular, when the order of convergence of the European option is higher than that of the American option, we can expect little gain. This approach is due to Hull and White, [9].

Our third implementation option is *Richardson extrapolation.* If the price after n steps is

$$X_n = \text{TruePrice} + \frac{E}{n} + o(1/n), \tag{3.1}$$

then taking

$$Y_n = A_n X_n + B_n X_{2n+1}$$

with A_n and B_n satisfying

$$A_n + B_n = 1.0,$$
$$\frac{A_n}{n} + \frac{B_n}{2n+1} = 0.0,$$

we get

$$Y_n = \text{TruePrice} + o(1/n).$$

We therefore take

$$A_n = 1 - \left(1 - \frac{n}{2n+1}\right)^{-1}, \tag{3.2}$$

$$B_n = \left(1 - \frac{n}{2n+1}\right)^{-1}. \tag{3.3}$$

Whilst the error for an American put will not be of the form in (3.1), if it is of this form plus a small oscillatory term, Richardson extrapolation will still reduce the size of the error. One way to reduce the oscillations is to use smoothing. Broadie and Detemple, [2], suggested using smoothing and Richardson extrapolation together.

Our fourth implementation option is *smoothing*. Inside the tree model, there will no exercise opportunities within the final step, so the derivative is effectively European. This suggests that a more accurate price can be obtained by using the Black–Scholes formula for this final step. With this technique we therefore replace the value at each node in the second final layer with the maximum of the intrinsic and the Black–Scholes value.

Since we can use each of these techniques independently of the others, this yields 2^4 different choices. We also consider an extra choice which is relevant when doing both smoothing and Richardson extrapolation. It is possible that making the trees with n and $2n + 1$ steps smooth at the same time will result in better extrapolation than smoothing both of them at the last possible time which will be different for the two trees. We can therefore smooth at the first step after $(n - 1)T/n$. This yields an extra 4 trees which we will refer to as being *matched*.

4. Numerical results

In order to assess the speed/accuracy trade-off of various tree methodologies without being influenced by special cases, an approach based on computing the root-mean-square (rms) error was introduced by Broadie and Detemple, [2]. One picks option parameters from a random distribution and assesses the pricing error by using a model with a large number of steps as the true value. One then looks at the number of option evaluations per second against the rms error.

Key	Truncate	Control	Smooth	Extrapolate	Match
0	no	no	no	no	n/a
1	yes	no	no	no	n/a
2	no	yes	no	no	n/a
3	yes	yes	no	no	n/a
4	no	no	yes	no	n/a
5	yes	no	yes	no	n/a
6	no	yes	yes	no	n/a
7	yes	yes	yes	no	n/a
8	no	no	no	yes	n/a
9	yes	no	no	yes	n/a
10	no	yes	no	yes	n/a
11	yes	yes	no	yes	n/a
12	no	no	yes	yes	no
13	yes	no	yes	yes	no
14	no	yes	yes	yes	no
15	yes	yes	yes	yes	no
16	no	no	yes	yes	yes
17	yes	no	yes	yes	yes
18	no	yes	yes	yes	yes
19	yes	yes	yes	yes	yes

TABLE 3.1. The labelling of implementation options by number.

Since we want to be clear that our results do not depend on particular choices of random distribution, we use identical parameters to that of Leisen, [17], and proceed as follows: volatility is distributed uniformly between 0.1 and 0.6. The time to maturity is, with probability 0.75, uniform between 0.1 and 1.00 years and, with probability 0.25, uniform between 1.0 and 5.0 years. We take the strike price, K, to be 100 and take the initial asset price S_0 to be uniform between 70 and 130. The continuously compounding rate, r, is, with probability 0.8, uniform between 0.0 and 0.10 and, with probability 0.2, equal to 0.0.

Some authors, [22], [18], have suggested using a model set of 16 extreme cases. Whilst this is probably enough when comparing a small number of models, here we will be doing 220 different models and want the number of test cases to be greater than the number of models. We therefore used 2200 cases and used the same set of options for each of the 220 models.

When computing the rms error, Leisen following Broadie and Detemple suggests using the relative error and dropping any cases where the true value is below 0.5 in order to avoid small absolute errors on small values distorting the results. Whilst this is reasonable, it is also criticizable in that it is particularly lenient in the hardest cases. For a deeply out-of-the-money option, the value will often be less than 0.5 so these are neglected. For a deeply in-the-money option most of the value will be the intrinsic value, so a large error on the model-dependent part may translate into a small error in relative terms.

We therefore introduce a new error measure which is intended to retain the good features of the Broadie–Detemple approach whilst excising the not so good ones. We therefore take the modified relative error to be

$$\frac{\text{TreePrice} - \text{TruePrice}}{0.5 + \text{TruePrice} - \text{IntrinsicValue}}.$$

This has the virtue of stopping small errors in small prices appearing to be large for deeply out-of-the-money options. It also stops the model-independent intrinsic value affecting our assessment of the relative error: we are computing the relative error of the model-dependent part. Note that for out-of-the-money options subtracting the intrinsic value will have no effect.

For each of the eleven trees discussed, we run the tree with each of the 20 options according to the keys in Table 3.1. We restrict to trees with odd numbers of steps, since some trees, e.g. Leisen–Reimer, are only defined in that case. For our model prices we used the Leisen–Reimer tree with 5001 steps and Richardson extrapolation; this is following the choice of Staunton, [18]. All timings are done with a 3 GigaHertz single core Pentium 4 processor.

We ran each tree with the following numbers of steps

$$25, 51, 101, 201, 401, 801.$$

We then used linear interpolation of log time against log error to estimate the time required to find an absolute rms error of 1E-3, a modified relative rms error of 1E-3 and a relative rms error (Broadie-Detemple) of 0.5E-4. The difference in target values expressing the fact that the Broadie-Detemple measure is more lenient. We note the the rms error is not truly a linear function so the interpolation will sometimes introduce a small amount of error; however, in most cases it will be interpolation not extrapolation, and the amount of curvature will by slight so any error will be small. In any case, our objective is to assess which methods are good and bad, and the level of accuracy is more than good enough to do so. We will look at the most effective methods in more detail.

From studying Tables 4.1, 4.2, and 4.3, we see various effects. The most marked one is that Richardson extrapolation is very effective when the tree has been smoothed either by adapting the tree to the strike, or by using the BS formula. In particular, the unadapted trees CRR, JR, JRRN, Tian

key	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
CP	0.9	2.7	8.0	19.5	7.8	17.3	27.1	53.5	44.4	73.1	27.0	48.7	42.2	70.2	26.0	47.3	43.7	72.5	26.3	47.8
CRR	2.0	5.4	9.0	21.3	7.5	16.6	26.4	52.4	0.3	0.9	1.4	4.0	45.8	75.4	32.0	56.4	47.3	77.7	32.9	57.7
J4	7.2	16.3	5.0	12.9	1.3	3.7	16.0	34.4	44.4	73.2	31.1	54.9	42.1	70.1	29.1	51.9	43.5	72.3	30.0	53.4
JR	1.8	4.9	6.4	16.0	8.4	18.4	16.9	36.0	0.23	0.7	1.3	3.7	42.7	70.8	30.1	53.4	44.9	74.3	31.4	55.5
JRRN	1.8	4.9	6.4	16.1	8.4	18.4	16.9	36.0	0.24	0.7	1.3	3.7	42.7	71.0	30.1	53.6	45.0	74.4	30.9	55.6
LR	7.2	16.2	4.9	12.9	1.3	3.7	16.0	34.4	44.2	72.9	31.1	54.8	42.0	69.8	29.1	51.9	43.5	72.2	30.0	53.3
Tian	0.3	1.0	0.8	2.8	0.3	1.1	0.9	3.1	0.27	0.8	1.5	4.3	143.6	200.5	98.5	143.7	125.6	179.4	85.0	127.0
adjusted	1.1	3.1	4.3	11.5	7.8	17.4	12.9	28.6	44.2	72.9	31.0	54.7	41.7	69.3	29.2	52.1	43.5	72.1	29.9	53.3
Chriss	1.9	5.1	5.8	14.7	8.9	19.3	15.4	33.3	0.24	0.7	1.3	3.7	42.8	71.0	30.1	53.4	44.9	74.2	31.5	55.6
flexible	1.0	3.0	2.1	6.3	7.1	16.1	25.4	50.5	44.3	72.9	29.4	52.3	41.2	68.7	28.0	50.2	43.1	71.0	28.8	51.7
split	1.1	3.1	3.5	9.6	1.9	5.2	6.1	15.3	101.7	149.2	74.7	17.6	83.3	126.4	62.9	49.2	99.3	146.7	65.8	48.9

TABLE 4.1. Number of option evaluations a second with an absolute rms error of 1E-3.

key	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
CP	23	45	219	305	179	257	460	573	659	750	441	511	574	669	398	469	677	770	401	473
CRR	61	103	320	420	163	238	635	756	14	27	34	59	592	680	401	468	629	727	421	496
J4	465	601	323	428	46	82	859	991	910	1007	634	715	780	694	405	460	680	747	470	522
JR	43	76	123	187	110	171	145	216	8	17	57	91	1647	1711	1227	1259	1896	1919	1272	1293
JRRN	43	76	124	188	110	171	146	217	8	17	57	91	1703	1796	1242	1278	1847	1895	1239	1279
LR	468	593	324	424	46	82	869	981	911	1009	635	718	774	876	486	562	825	935	537	588
Tian	10	22	25	50	13	26	27	53	8	16	42	70	1516	1556	1120	1147	1334	1361	936	970
adjusted	25	47	339	442	184	264	831	953	889	988	437	490	742	839	417	598	806	897	425	607
Chriss	44	79	123	187	122	187	145	216	8	17	57	91	1399	1437	1037	1074	1488	1503	1250	1292
flexible	35	63	75	125	154	228	701	821	716	819	381	452	730	824	397	467	593	697	339	410
split	34	62	229	77	228	321	570	231	3332	3330	253	26	1528	1625	216	71	2337	2365	207	72

TABLE 4.2. Number of option evaluations a second obtainable with a modified relative error of 1E-3 using 0.5 additional weighting.

key	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
CP	22	42	231	319	175	252	566	682	296	375	414	484	283	362	380	451	290	370	362	435
CRR	41	73	255	347	166	241	583	704	5	12	46	76	716	808	501	569	753	855	506	582
J4	445	578	309	412	38	69	658	786	1254	1344	888	963	997	930	527	586	906	977	595	650
JR	31	58	92	148	94	150	107	167	5	11	47	77	1236	1305	876	922	1150	1205	788	835
JRRN	31	58	93	148	94	150	107	167	5	11	47	78	1349	1453	931	992	1242	1330	825	898
LR	446	570	311	409	38	69	665	778	1282	1371	893	970	1021	1119	609	686	1014	1124	655	684
Tian	13	28	33	63	17	34	32	62	5	11	47	77	1652	1675	1164	1187	1266	1302	859	901
adjusted	27	52	220	305	22	43	441	553	924	1023	450	496	873	965	433	658	973	1058	430	693
Chriss	31	58	92	147	96	152	107	167	5	11	47	77	1150	1214	820	875	1069	1136	832	900
flexible	14	29	76	125	163	239	607	726	289	368	384	456	282	359	354	423	283	364	351	422
split	3	7	78	69	2	5	119	183	453	557	207	24	443	533	181	67	532	635	182	67

TABLE 4.3. Number of option evaluations a second obtainable with a relative error of 0.5E-4 with 0.5 cut-off.

steps	Error Split 8	Time Split 8	Error Split 9	Time Split 9
25	8.491E-04	4.687E-04	8.491E-04	4.622E-04
51	5.292E-04	1.698E-03	5.292E-04	1.447E-03
101	1.108E-04	6.868E-03	1.108E-04	4.946E-03
201	4.710E-05	2.743E-02	4.710E-05	1.615E-02
401	2.089E-05	1.092E-01	2.089E-05	5.342E-02
801	6.916E-06	4.402E-01	6.915E-06	1.831E-01

TABLE 4.4. Detailed data for split 8 and split 9. Error is modified relative error. The time is the average time to price one option.

and Chriss do very badly in cases 8 through 11, but do much better in cases 12 and higher, reflecting the Black–Scholes smoothing.

The control methodology is useful when the error is large, but when the price is accurate without it, adding it in merely slows things down. This suggests it is no longer a worthwhile technique for this problem. In particular, the key of 15 almost always does worse than the key of 13 with the only exceptions being the Chang–Palmer and flexible trees using the Broadie–Detemple error measure.

Depending upon on our error methodology the most effective trees for this test are Tian 13 (absolute and Broadie-Detemple) and split 8 (modified relative.) Note, however, that split 9 (i.e. with truncation) is almost as good as split 8, and, in fact, on detailed analysis, Table 4.4, we see that the reason is that 25 steps is too many to get an error of $1E-3$. The time has therefore been extrapolated giving the appearance that the untruncated tree is better when, in fact, it is not. For every case run, the errors are indistinguishable whilst the split 9 tree is better on time.

Other points to note are that Leisen–Reimer and J4 give almost identical results as expected, and that the adjusted tree with RE is also very similar to these trees with RE.

Another curiosity is that in certain cases the combination of truncation and control does very badly for the split tree. This suggests that the truncated split tree is doing a poor job of pricing the European option.

If one takes a key of 0, that is with no acceleration techniques, it is, in fact, the LR and J4 trees that are best, and Tian that is the worst. This demonstrates that the accuracy in the zero case is a poor predictor of accuracy after acceleration.

The contents of the final four columns and the previous four suggest that the precise choice of time to smooth is not important in that the columns are qualitatively similar with no clear trends.

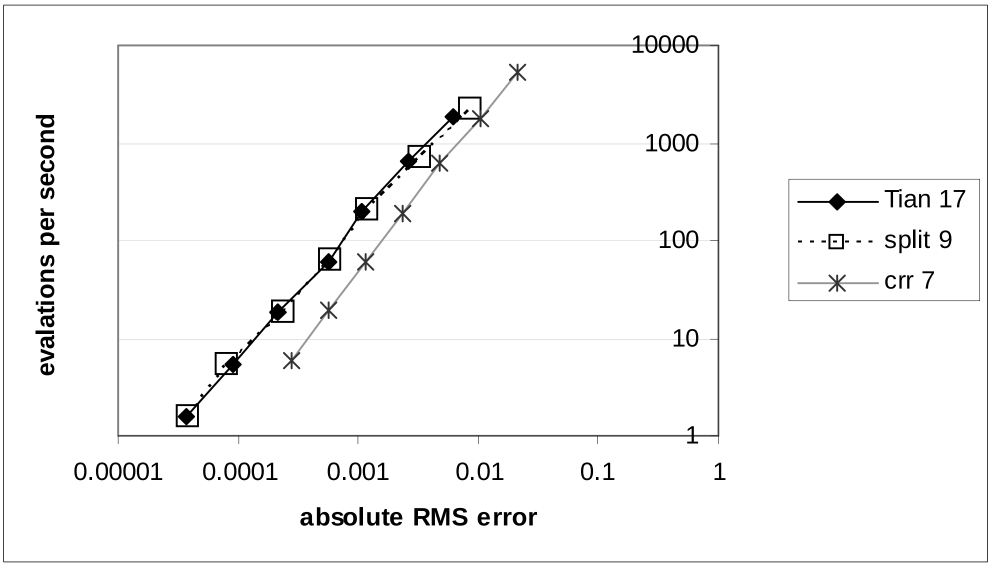

FIGURE 1. Number of evaluations per second against rms absolute error for three trees with log scale.

Whilst these tests have been effective for seeing how much time is required to get a good level of accuracy, they do not answer the question of which tree to use when a very high level of accuracy is required. A second set of tests was therefore run for the most accurate combinations of choices of tree parameters and acceleration techniques. The selection was made by examining which cases performed best in the first tests. In this case, the model prices were obtained from the Leisen–Reimer tree with 10001 steps and extrapolation.

The number of steps run were

$$101, 201, 401, 801, 1601.$$

The number of option prices run was $12,000$.

In Tables 4.5, 4.6, and 4.7, we present results for the three error methodologies. In each case, we have used the same 27 choices but we have reordered so that the error with 1601 steps is in ascending order in each case.

Examining Table 4.5, we see from the column with 1601 steps that Tian 17 achieves the smallest error with split 9 close behind. The only methods which are faster with that number of steps are the last four which do not involve Richardson extrapolation. Their errors are much larger, however. We need to compare with different number of steps, this is done in Figure

Tian	15	3.88E-05	1.31	9.16E-05	4.42	2.24E-04	14.45	5.58E-04	46.02	1.03E-03	145.08
Tian	13	4.00E-05	1.56	9.35E-05	5.48	2.24E-04	18.68	5.56E-04	61.76	1.03E-03	201.07
split	13	4.10E-05	1.55	9.11E-05	5.45	2.58E-04	18.61	6.33E-04	61.42	1.34E-03	200.53
adjusted	17	5.57E-05	1.55	1.57E-04	5.46	3.78E-04	18.60	8.86E-04	61.47	1.96E-03	199.46
J4	9	5.58E-05	1.56	1.57E-04	5.46	3.75E-04	18.66	8.78E-04	61.95	1.97E-03	203.82
J4	8	5.58E-05	0.57	1.57E-04	2.31	3.75E-04	9.11	8.78E-04	36.32	1.97E-03	147.28
LR	9	5.58E-05	1.56	1.57E-04	5.47	3.75E-04	18.70	8.78E-04	62.03	1.98E-03	203.82
LR	8	5.58E-05	0.57	1.57E-04	2.29	3.75E-04	9.12	8.78E-04	36.26	1.98E-03	146.11
J4	17	5.63E-05	1.55	1.58E-04	5.45	3.78E-04	18.54	8.78E-04	61.35	1.97E-03	199.28
LR	17	5.63E-05	1.55	1.58E-04	5.46	3.77E-04	18.58	8.78E-04	61.37	1.97E-03	199.46
JRRN	19	6.68E-05	1.31	1.36E-04	4.42	3.12E-04	14.42	8.23E-04	45.83	1.73E-03	143.41
Chriss	17	6.70E-05	1.56	1.36E-04	5.47	3.13E-04	18.60	8.23E-04	61.50	1.72E-03	197.87
JRRN	17	6.70E-05	1.56	1.36E-04	5.47	3.13E-04	18.62	8.24E-04	61.57	1.72E-03	199.10
JR	17	6.70E-05	1.55	1.36E-04	5.46	3.13E-04	18.58	8.24E-04	61.40	1.72E-03	198.93
Chriss	13	6.93E-05	1.56	1.44E-04	5.46	3.30E-04	18.59	8.51E-04	61.66	1.80E-03	200.35
JR	13	6.93E-05	1.55	1.44E-04	5.46	3.31E-04	18.62	8.52E-04	61.71	1.80E-03	201.08
JRRN	13	6.93E-05	1.55	1.44E-04	5.46	3.31E-04	18.61	8.52E-04	61.71	1.80E-03	200.90
flexible	9	1.01E-04	1.55	2.06E-04	5.46	3.67E-04	18.67	8.54E-04	62.00	2.42E-03	202.90
flexible	13	1.02E-04	1.55	2.11E-04	5.45	3.83E-04	18.43	8.98E-04	61.32	2.49E-03	200.18
CP	17	1.03E-04	1.55	2.05E-04	5.46	3.53E-04	18.59	8.56E-04	61.42	2.25E-03	198.92
CRR	7	2.83E-04	5.79	5.58E-04	18.97	1.14E-03	61.44	2.33E-03	192.60	4.72E-03	630.97
flexible	7	2.84E-04	5.79	5.63E-04	19.00	1.16E-03	61.47	2.38E-03	192.11	4.89E-03	629.16
LR	7	3.45E-04	5.75	7.08E-04	18.91	1.38E-03	61.39	2.80E-03	192.28	5.94E-03	622.11
J4	7	3.45E-04	5.79	7.08E-04	18.97	1.38E-03	61.42	2.80E-03	192.43	5.94E-03	630.97

TABLE 4.5. rms error in absolute terms and number of option evaluations per second for 27 good cases using 12,000 evaluations.

Chriss	17	5.07E-06	1.56	1.17E-05	5.47	2.49E-05	18.60	6.94E-05	61.50	1.45E-04	197.87
JR	17	5.07E-06	1.55	1.17E-05	5.46	2.49E-05	18.58	6.94E-05	61.40	1.44E-04	198.93
JRRN	17	5.07E-06	1.56	1.17E-05	5.47	2.49E-05	18.62	6.94E-05	61.57	1.44E-04	199.10
JRRN	19	5.07E-06	1.31	1.13E-05	4.42	2.44E-05	14.42	7.22E-05	45.83	1.46E-04	143.41
split	13	5.45E-06	1.55	1.33E-05	5.45	3.07E-05	18.61	7.11E-05	61.42	1.65E-04	200.53
Chriss	13	6.27E-06	1.56	1.51E-05	5.46	3.27E-05	18.59	7.75E-05	61.66	1.85E-04	200.35
JR	13	6.27E-06	1.55	1.52E-05	5.46	3.27E-05	18.62	7.76E-05	61.71	1.86E-04	201.08
JRRN	13	6.27E-06	1.55	1.52E-05	5.46	3.27E-05	18.61	7.76E-05	61.71	1.86E-04	200.90
J4	9	6.65E-06	1.56	1.41E-05	5.46	3.22E-05	18.66	7.27E-05	61.95	1.62E-04	203.82
J4	8	6.65E-06	0.57	1.41E-05	2.31	3.22E-05	9.11	7.27E-05	36.32	1.62E-04	147.28
LR	9	6.65E-06	1.56	1.41E-05	5.47	3.23E-05	18.70	7.30E-05	62.03	1.63E-04	203.82
LR	8	6.65E-06	0.57	1.41E-05	2.29	3.23E-05	9.12	7.30E-05	36.26	1.63E-04	146.11
adjusted	17	6.71E-06	1.55	1.43E-05	5.46	3.08E-05	18.60	7.43E-05	61.47	1.73E-04	199.46
J4	17	6.74E-06	1.55	1.43E-05	5.45	3.10E-05	18.54	7.44E-05	61.35	1.74E-04	199.28
LR	17	6.74E-06	1.55	1.43E-05	5.46	3.10E-05	18.58	7.43E-05	61.37	1.74E-04	199.46
CP	17	9.04E-06	1.55	1.88E-05	5.46	5.52E-05	18.59	1.33E-04	61.42	4.44E-04	198.92
flexible	9	1.09E-05	1.55	2.24E-05	5.46	4.55E-05	18.67	1.01E-04	62.00	3.42E-04	202.90
flexible	13	1.17E-05	1.55	2.46E-05	5.45	5.22E-05	18.43	1.08E-04	61.32	3.93E-04	200.18
CRR	7	4.75E-05	5.79	8.95E-05	18.97	1.64E-04	61.44	3.31E-04	192.60	7.39E-04	630.97
LR	7	5.10E-05	5.75	9.51E-05	18.91	1.81E-04	61.39	3.34E-04	192.28	6.12E-04	622.11
J4	7	5.10E-05	5.79	9.51E-05	18.97	1.81E-04	61.42	3.34E-04	192.43	6.12E-04	630.97
flexible	7	5.22E-05	5.79	9.04E-05	19.00	1.80E-04	6.15E+01	3.40E-04	192.11	7.23E-04	629.16

TABLE 4.6. rms error in modified relative terms with additional weight of 0.5 and number of option evaluations per second for 27 good cases using 12,000 evaluations.

split	9	3.50E-06	1.56	8.53E-06	5.51	2.09E-05	18.87	5.71E-05	62.27	1.71E-04	204.19
split	17	3.57E-06	1.56	8.42E-06	5.47	1.97E-05	18.66	5.11E-05	61.32	1.40E-04	198.76
JRRN	19	3.86E-06	1.31	8.69E-06	4.42	2.09E-05	14.42	4.83E-05	45.83	1.14E-04	143.41
Chriss	17	3.92E-06	1.56	8.76E-06	5.47	2.10E-05	18.60	4.86E-05	61.50	1.13E-04	197.87
JR	17	3.92E-06	1.55	8.76E-06	5.46	2.10E-05	18.58	4.86E-05	61.40	1.13E-04	198.93
JRRN	17	3.92E-06	1.56	8.76E-06	5.47	2.10E-05	18.62	4.86E-05	61.57	1.13E-04	199.10
split	13	4.16E-06	1.55	1.01E-05	5.45	2.46E-05	18.61	6.42E-05	61.42	1.82E-04	200.53
Chriss	13	4.39E-06	1.56	1.03E-05	5.46	2.29E-05	18.59	5.23E-05	61.66	1.15E-04	200.35
JR	13	4.39E-06	1.55	1.03E-05	5.46	2.29E-05	18.62	5.23E-05	61.71	1.15E-04	201.08
JRRN	13	4.39E-06	1.55	1.03E-05	5.46	2.29E-05	18.61	5.23E-05	61.71	1.15E-04	200.90
LR	9	5.47E-06	1.56	1.23E-05	5.47	2.48E-05	18.70	5.35E-05	62.03	1.16E-04	203.82
LR	8	5.47E-06	0.57	1.23E-05	2.29	2.48E-05	9.12	5.35E-05	36.26	1.16E-04	146.11
J4	9	5.47E-06	1.56	1.23E-05	5.46	2.48E-05	18.66	5.35E-05	61.95	1.16E-04	203.82
J4	8	5.47E-06	0.57	1.23E-05	2.31	2.48E-05	9.11	5.35E-05	36.32	1.16E-04	147.28
adjusted	17	5.48E-06	1.55	1.24E-05	5.46	2.47E-05	18.60	5.42E-05	61.47	1.18E-04	199.46
J4	17	5.51E-06	1.55	1.24E-05	5.45	2.48E-05	18.54	5.42E-05	61.35	1.19E-04	199.28
LR	17	5.51E-06	1.55	1.24E-05	5.46	2.48E-05	18.58	5.43E-05	61.37	1.20E-04	199.46
CP	17	8.08E-06	1.55	2.00E-05	5.46	4.74E-05	18.59	1.02E-04	61.42	3.00E-04	198.92
flexible	9	8.49E-06	1.55	1.87E-05	5.46	4.47E-05	18.67	1.22E-04	62.00	2.97E-04	202.90
flexible	13	8.81E-06	1.55	1.96E-05	5.45	4.64E-05	18.43	1.26E-04	61.32	3.00E-04	200.18
CRR	7	2.81E-05	5.79	5.36E-05	18.97	1.03E-04	61.44	2.38E-04	192.60	4.43E-04	630.97
flexible	7	2.86E-05	5.79	5.38E-05	19.00	1.05E-04	61.47	2.45E-04	192.11	4.44E-04	629.16
J4	7	3.81E-05	5.79	7.13E-05	18.97	1.32E-04	61.42	2.40E-04	192.43	4.34E-04	630.97
LR	7	3.81E-05	5.75	7.13E-05	18.91	1.32E-04	61.39	2.40E-04	192.28	4.34E-04	622.11

TABLE 4.7. rms error in Broadie–Detemple relative terms with cut-off of 0.5 and number of option evaluations per second for 27 good cases using 12,000 evaluations.

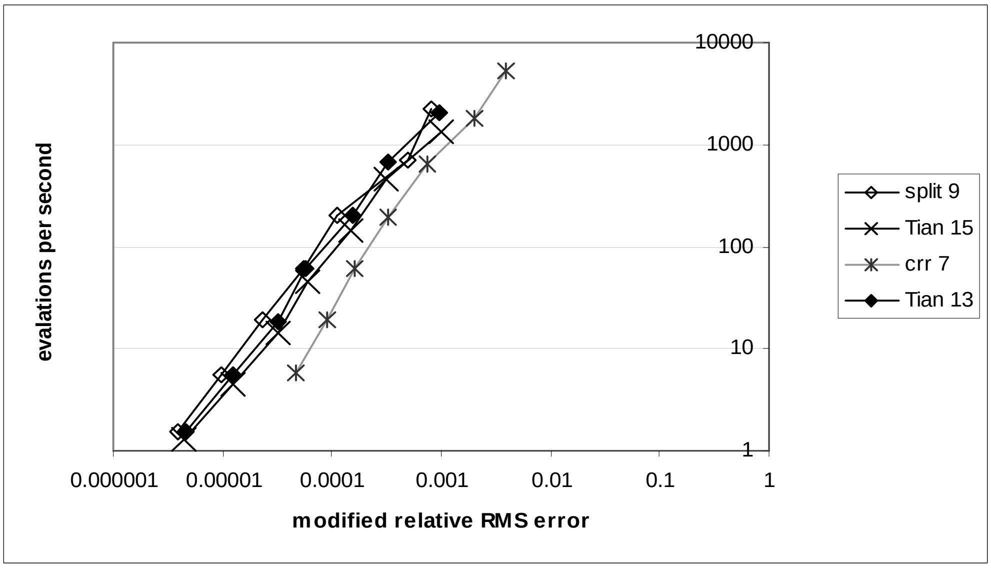

FIGURE 2. Number of evaluations per second against modified relative rms error for four trees with log scale.

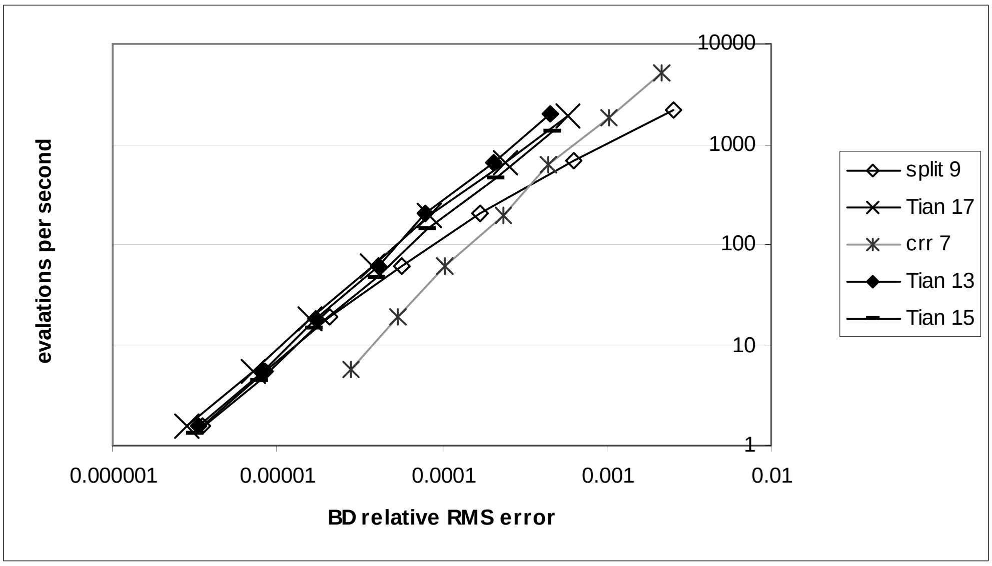

FIGURE 3. Number of evaluations per second against Broadie–Detemple relative rms error for five trees with log scale.

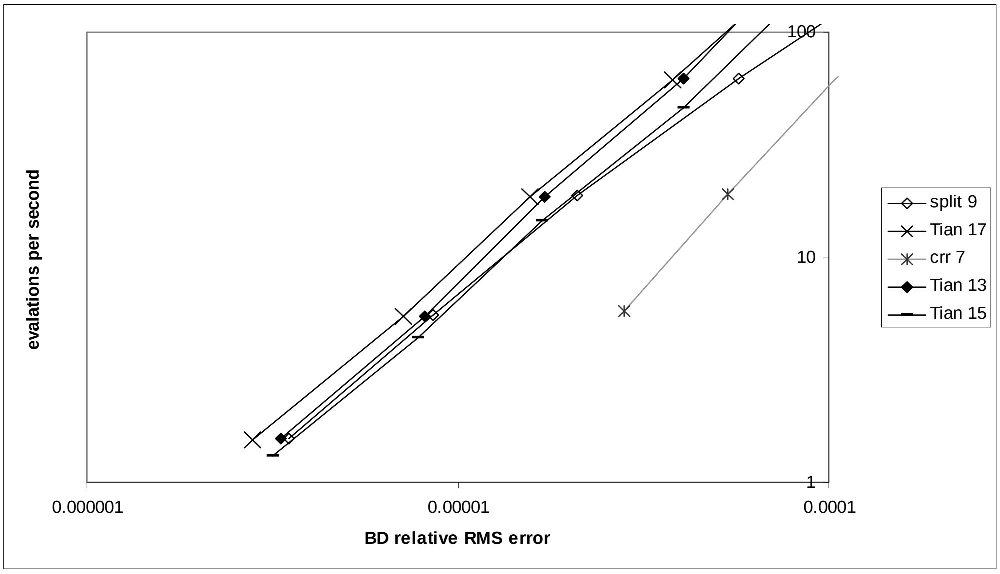

FIGURE 4. Number of evaluations per second against Broadie–Detemple relative rms error for five trees with log scale.

1. We see clearly that CRR 7 is substantially worse than Tian 17 and split 9.

If one's objective is to minimize absolute error then it is clear that we should use Tian 17: that is third moment matching with smoothing, Richardson extrapolation, truncation and matching smoothing times. The choice of split 9 is also competitive. Note that the smallest error varies with number of steps and with 401 steps, it is split 9 that wins. This suggests that the trees are essentially the same in accuracy.

For modified relative error, we examine Table 4.6, we see from the column with 1601 steps that split 9 has the smallest error with split 17, Tian 13, Tian 15 and Tian 17 almost as good. Again the last 4 are faster with larger errors so we plot error against speed in Figure 2. We see clearly that CRR 7 is substantially worse than Tian 15, Tian 13 and split 9. We also see that Tian 15 is worse than Tian 13. The comparison between Tian 15 and Tian 13 suggests that although the use of a control does reduce error in this case, the additional computational effort is not worth the improvement.

If one's objective is to minimize modified rms error then it is clear that we should use split 9; Tian 13 is also a good choice.

Examining Table 4.7, we see from the column with 1601 steps that Tian 17 achieves the smallest error with Tian 15, Tian 13 and split 9 almost as

name	key	order absolute	order modified	order BD
CRR	0	-0.508	-0.454	-0.506
CRR	12	-0.505	-0.598	-0.676
CRR	13	-0.575	-0.684	-0.770
LR	9	-0.738	-0.756	-0.710
Split	9	-0.922	-0.790	-0.925
Tian	13	-0.829	-0.672	-0.724
Tian	17	-0.856	-0.906	-0.766

TABLE 4.8. Order of convergence as expressed as a power of time for a selected few interesting cases.

good. The only methods which are faster with that number of steps are yet again the four last ones which do not involve extrapolation and we compare with different number of steps, in Figure 3 and in Figure 4. We see clearly that CRR 7 is substantially worse than Tian 17, Tian 15, Tian 13 and split 9. We also see that Tian 15 is worse than Tian 13 and Tian 17. Again, the comparison between Tian 15 and Tian 13 suggests that although the use of a control does reduce error, the additional computational effort is not worth the improvement.

If one's objective is to minimize Broadie-Detemple rms error then it is clear that we should use Tian 17; Tian 13 and split 9 are also viable choices.

The reader may be interested in the order of convergence as well as the size of the errors. These were estimated by regressing the log RMS error against log time taken and fitting the best straight line through the cases with 201, 401 and 801 steps. The slopes are displayed in Table 4.8. We display results for absolute errors, relative errors with modification, and the Broadie–Detemple relative errors.

CRR 0 corresponds to the original tree of Cox, Ross and Rubinstein with no acceleration techniques, and its order is roughly -0.5. The CRR 12 tree corresponds to the BBSR method of Broadie and Detemple. Its convergence order is about $-2/3$ as a function of time, and so $-4/3$ as a function of the number of steps (when using the BD error measure.) Curiously, the order of convergence for absolute errors does not appear to improve above that of CRR 0 although the constant is, of course, much lower. The Tian 13 and 17 methods, and the split 9 method again display more rapid convergence than the other methods.

5. Conclusion

Pulling all these results together, we see that for pricing an American put option in the Black–Scholes model with high accuracy and speed, we should always use truncation and extrapolation. We should also use a technique which reduces the oscillations in the European case: that is smoothing or modifying the lattice to take account of strike.

The best overall results have been obtained using the Tian third moment matching tree together with truncation, smoothing and extrapolation, and the new split tree which uses a time-dependent drift to minimize oscillations, together with extrapolation and truncation. We have not investigated in this paper the optimal level of truncation but have instead adopted a level that has minimal effect on price. The Tian tree has the added bonus that the node placement does not depend on strike so there is the additional possibility of pricing many options simultaneously.

Interestingly, neither of the preferred trees are amongst those in popular use at the current time. This is despite the fact that the Tian tree was first introduced fifteen years ago. A possible explanation is that its virtue, matching three moments, does not have much effect when the pay-off is not smooth, and so initial tests without smoothing and extrapolation showed it to be poor.

References

[1] A.D. Andicropoulos, M. Widdicks, P.W. Duck, D.P. Newton, Curtailing the range for lattice and grid methods, *Journal of Derivatives,* Summer 2004, 55–61

[2] M. Broadie, J. Detemple, American option valuation: new bounds, approximations, and a comparison of existing methods, *The Review of Financial Studies,* Winter 1996 Vol. 9, No. 4, pp. 1211–1250

[3] J-H Chan, M. Joshi, R. Tang, C. Yang, Trinomial or binomial: accelerating American put option pricing on trees, to appear in *Journal of Futures Markets,*

[4] D. Chance, A synthesis of binomial options pricing models, preprint Feb 2007

[5] L-B. Chang, K. Palmer, smooth convergence in the binomial model, *Finance and Stochastics,* Vol 11, No 2, (2007), 91–105

[6] N. Chriss, *Black–Scholes and beyond: Option Pricing Models,* McGraw–Hill, New York 1996

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

[8] F. Diener, M. Diener, Asymptotics of the price oscillations of a European call option in a tree model, *Mathematical Finance,* Vol. 14, No. 2, (April 2004), 271–293

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

[10] R. Jarrow, A. Rudd, *Option pricing,* Homewood, IL: Richard D. Irwin, (1993)

[11] R. Jarrow, S. Turnbull, *Derivative Securities,* 2nd ed. Cincinnati: SouthWestern College Publishing. (2000)

[12] M. Joshi, *The Concepts and Practice of Mathematical Finance*, Cambridge University Press (2003)

[13] M. Joshi, Achieving Smooth Convergence for The Prices of European Options In Binomial Trees, to appear in *Quantitative Finance,*

[14] M. Joshi, Achieving Higher Order Convergence for The Prices of European Options In Binomial Trees, to appear in *Mathematical Finance,*

[15] D. Lamberton, Error estimates for the binomial approximation of American put option, *Annals of Applied Probability,* Volume 8, Number 1 (1998), 206–233.

[16] D.P. Leisen, M. Reimer, Binomial models for option valuation-examining and improving convergence, *Applied Mathematical Finance,* 3, 319–346 (1996)

[17] D.P. Leisen, Pricing the American put option, a detailed convergence analysis for binomial models, *Journal of Economic Dynamics and Control,* 22, 1419–1444, (1998).

[18] M. Staunton, Efficient estimates for valuing American options, *the Best of Wilmott 2,* John Wiley and Sons Ltd (2005)

[19] Y. Tian, A modified lattice approach to option pricing, *Journal of Futures Markets,* 13(5), 563–577, (1993)

[20] Y. Tian, A flexible binomial option pricing model, *Journal of Futures Markets* 19: 817–843, (1999)

[21] J. Walsh, The rate of convergence of the binomial tree scheme, *Finance and Stochastics,* 7, 337–361 (2003)

[22] M. Widdicks, A.D. Andricopoulos, D.P. Newton, P.W. Duck, On the enhanced convergence of standard lattice models for option pricing, *The Journal of Futures Markets,* Vol. 22, No. 4, 315–338 (2002)

Centre for Actuarial Studies, Department of Economics, University of Melbourne, Victoria 3010, Australia

E-mail address: mark@markjoshi.com