TRINOMIAL OR BINOMIAL: ACCELERATING AMERICAN PUT OPTION PRICE ON TREES

JIUN HONG CHAN, MARK JOSHI, ROBERT TANG, AND CHAO YANG

ABSTRACT. We investigate the pricing performance of eight trinomial trees and one binomial tree, which was found to be most effective in an earlier paper, under twenty different implementation methodologies for pricing American put options. We conclude that the binomial tree, the Tian third order moment matching tree with truncation, Richardson extrapolation and smoothing performs better than the trinomial trees.

1. INTRODUCTION

Various types of binomial and trinomial trees have been proposed in the literature for pricing financial derivatives. Since tree models are backward methods, they are effective for pricing American-type derivatives. Joshi (2007c) conducted an empirical investigation on the performance of eleven different binomial trees on American put options using twenty different combinations of acceleration techniques; he found that the best results were obtained by using the "Tian third order moment tree" (which we refer to as Tian3 binomial tree henceforth) together with truncation, smoothing and Richardson extrapolation. Here we perform a similar analysis for trinomial trees and also compare the pricing results to the Tian3 binomial tree.

We use the same four acceleration techniques implemented by Joshi (2007c): *control variate* due to Hull and White (1988), *truncation* due to Andicropoulos, Widdicks, Duck and Newton (2004), *smoothing* and *Richardson extrapolation* due to Broadie and Detemple (1996). These four techniques can be implemented independently or collectively and therefore yield 16 combinations of acceleration techniques one can use when pricing a derivative using trees. For detailed discussion and merits of these acceleration techniques, we refer reader to Joshi (2007c).

Date: October 29, 2008.

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

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

As with binomial trees, there is freedom to choose the parameters of a trinomial tree, depending upon what characteristics one wishes to emphasize. For example, one can attempt to match higher moments, or attempt to obtain smooth convergence.

We will examine 8 of these choices in this paper. This results in 128 different ways to price an American put option using trinomial trees. We use three different error measures to determine the most accurate trinomial tree, whilst using a Leisen-Reimer tree accelerated by smoothing and Richardson extrapolation with a large number of steps as the true price.

We find that the best choice of trinomial tree depends on how one defines error, but in all cases one should use the acceleration techniques of smoothing, Richardson extrapolation and truncation. The best choices of tree parameters were the Tian fourth order moment matching tree, the Boyle tree with parameter 1.2 and the equal probability tree. (See Section 2 for the precise definitions of these trees.)

We compared their performances with the best binomial tree found by Joshi (2007c) which is the Tian third order moment-matching tree with smoothing, Richardson extrapolation and truncation: the binomial tree turns out to be substantially faster under all the error measures.

A review of trinomial trees and our eight choices of parameters are discussed in Section 2. The different ways these can be accelerated is discussed in Section 3. We discuss testing methodologies in Section 4. We present numerical results in Section 5 and conclude in Section 6.

We thank the Faculty of Economics and Commerce at the University of Melbourne for supporting this work through a faculty research grant.

2. Choices of trinomial tree parameters

We quickly review our 8 choices of trinomial trees in this section while the review for the binomial tree used in this paper was discussed by Joshi (2007c). A trinomial tree is characterized by the following five parameters:

(1) the probability of an up move p_u,
(2) the probability of an down move p_d,
(3) the multiplier on the stock price for an up move u,
(4) the multiplier on the stock price for a middle move m,
(5) the multiplier on the stock price for a down move d.

A recombining tree is computationally more efficient so we require

$$ud = m^2, \tag{2.1}$$

that is, an up move followed by a down move is the same as two middle moves; we therefore have four free parameters.

We work in the Black-Scholes model with the usual parameters: $S(0)$ is the current spot price, L is the strike price of the option, r is the continuously compounding risk-free rate, σ is the volatility and T is the maturity of the option. We define the following variables

$$M = \exp(r\Delta t), \tag{2.2}$$

$$V = \exp(\sigma^2 \Delta t), \tag{2.3}$$

$$\Delta t = T/N, \tag{2.4}$$

where N is the total number of steps of a trinomial tree. For a tree to be risk-neutral, the mean and variance across each time steps must be asymptotically correct. Since we have four parameters and two constraints, this provides us with two degree of freedom in setting the trinomial trees' parameters.

Our first three trinomial trees involve moment-matching in spot space. The Tian equal probability tree sets the up and down move probabilities to be equal to 1/3 and matches the first two moments exactly (Tian, 1993):

$$m = \frac{M(3-V)}{2}, \tag{2.5}$$

$$K = \frac{M(V+3)}{4}, \tag{2.6}$$

$$u = K + \sqrt{K^2 - m^2}, \tag{2.7}$$

$$d = K - \sqrt{K^2 - m^2}. \tag{2.8}$$

We shall call this the "EqualProb" tree.

The Tian fourth order moment matching (Tian4) tree drops the equal probability constraint of the EqualProb tree and matches the first four moments exactly (Tian, 1993):

$$m = MV^2, \tag{2.9}$$

$$K = \frac{M}{2}(V^4 + V^3), \tag{2.10}$$

$$u = K + \sqrt{K^2 - m^2}, \tag{2.11}$$

$$d = K - \sqrt{K^2 - m^2}, \tag{2.12}$$

$$p_u = \frac{md - M(m+d) + M^2V}{(u-d)(u-m)}, \tag{2.13}$$

$$p_d = \frac{um - M(u+m) + M^2V}{(u-d)(m-d)}. \tag{2.14}$$

Joshi (2007a) introduced an adjusted binomial tree where the tree is centered on the strike in log space. Similarly, here we introduce an *adjusted* trinomial tree[1] by setting the central node on the last layer of the tree to be equal to the strike price of the option, that is, we set $m = (L/S(0))^{1/N}$, and matching the first three moments precisely. The p_u, p_d, u and d have the same expressions as the Tian4 tree except

$$K = \frac{V}{2}(MV + m) + \frac{m}{2M}(m - M). \tag{2.15}$$

The next five trees involve moment-matching in the log space. The *LogSpace* tree matches the first four moments in the log space:

$$m = e^{(r-\frac{1}{2}\sigma^2)\Delta t}, \tag{2.16}$$

$$u = me^{\sqrt{3\Delta t}\sigma}, \tag{2.17}$$

$$d = me^{-\sqrt{3\Delta t}\sigma}, \tag{2.18}$$

$$p_u = \frac{1}{6}, \tag{2.19}$$

$$p_d = \frac{1}{6}. \tag{2.20}$$

[1]Since the parameters for the adjusted tree are strike dependent, pricing options with different strikes on the same underlying stock will require new tree to be constructed for each of the strikes and this will have an obvious computational disadvantage

while Boyle (1986) has chosen the parameters to be:

$$m = 1, \tag{2.21}$$

$$u = \exp(\lambda\sigma\sqrt{\Delta t}), \tag{2.22}$$

$$d = \frac{1}{u}, \tag{2.23}$$

$$p_u = \frac{md - M(m+d) + M^2V}{(u-d)(u-m)}, \tag{2.24}$$

$$p_d = \frac{um - M(u+m) + M^2V}{(u-d)(m-d)}. \tag{2.25}$$

Boyle suggested that the choice of value of λ should be greater than one and the best results were obtained when λ is approximately 1.20. For our investigation purpose, we set λ equal to 1.10, 1.20 and 1.30. We refer these three trinomial trees as Boyle1.1, Boyle1.2 and Boyle1.3.

Lastly, the Kamrad and Ritchken (KR) tree matches the first moment and variance in the log space (Kamrad and Ritchken, 1991):

$$m = 1, \tag{2.26}$$

$$u = \exp(v), \tag{2.27}$$

$$d = 1/u, \tag{2.28}$$

$$v = \lambda\sigma\sqrt{\Delta t}, \tag{2.29}$$

$$\mu = r - \frac{\sigma^2}{2}, \tag{2.30}$$

$$\lambda = 1.22474, \tag{2.31}$$

$$p_u = \frac{1}{2\lambda^2} + \frac{\mu\sqrt{\Delta t}}{2\lambda\sigma}, \tag{2.32}$$

$$p_d = \frac{1}{2\lambda^2} - \frac{\mu\sqrt{\Delta t}}{2\lambda\sigma}. \tag{2.33}$$

3. The implementation choices

The implementation choices in this paper are the same as those in Joshi (2007c). We briefly review all the choices in this section and for more detailed descriptions, we refer the reader to Joshi (2007c).

3.1. **Truncation.** At each layer (step), the trinomial tree is developed up to 6 standard deviations away from the mean in log space computed under the risk-neutral measure. For nodes at the upper and lower edge

Key	Truncate	Control	Extrapolate	Smooth
0	no	no	no	no
1	yes	no	no	no
2	no	yes	no	no
3	yes	yes	no	no
4	no	no	yes	no
5	yes	no	yes	no
6	no	yes	yes	no
7	yes	yes	yes	no
8	no	no	no	yes
9	yes	no	no	yes
10	no	yes	no	yes
11	yes	yes	no	yes
12	no	no	yes	yes
13	yes	no	yes	yes
14	no	yes	yes	yes
15	yes	yes	yes	yes

TABLE 3.1. The labelling of implementation options by number.

of the truncated tree, we take the continuation value of the American put to be equal to the Black-Scholes price for the corresponding European put.

3.2. **Control variates.** We price the American put and the corresponding European put simultaneously. If P_A and P_E are the tree prices of the American and European put respectively, and P_{BS} is the true price of the European put given by the Black-Scholes formula, we take the error controlled price to be

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

3.3. **Richardson extrapolation.** [2]

If X_n is the price after n steps then we use the estimate

$$Y_n = 2X_{2n} - X_n.$$

as the price of the put option. Our motivations is that if

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

[2]This idea can be extended to extrapolation using three different prices obtained from n, $2n$ and $4n$ steps in an attempt to eliminate the second order term. However, this was not found to affect the final conclusions of this paper. This is probably because the two-point extrapolation makes the lead error term small but not zero in these cases.

then

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

Whilst this will not be wholly, we shall accelerate to the extent that it is true.

3.4. **Smoothing.** On the second last layer of tree, an American put within the disrectized model is a European put option, we therefore set the price of the American put on that layer to be the maximum of the intrinsic value and the Black-Scholes price.

We list keys describing all the possible choices in Table 3.1.

4. The Implementation of Numerical Tests

Our numerical tests follow those introduced by Joshi (2007c) which were inspired by Broadie and Dutemple (1996). We pick the option parameters from a random distribution and assess the pricing error by using a 14001 step Leisen-Reimer binomial tree with smoothing, truncation and Richardson extrapolation as a proxy to the true price. Prices obtained using a 14001 step binomial tree should be sufficiently close to the true price as 14001 steps is significantly larger than 5001 steps used by Staunton (2005) for similar analysis. The results in Joshi (2007c) suggest that the error with 1601 steps is around $5E-6$ (with slight variation according to error measure) and using $14,001$ steps should reduce the error by a factor of 8.5 (assuming $1/n$ convergence in the number of steps which is roughly correct.)

The option parameters were selected based on the following distribution: the current spot price, $S(0)$, has a uniform distribution between 70 and 130; the strike price, L, is equal to 100; the continuously compounding interest rate, r, is with probability 0.8, uniform between 0.0 and 0.10 and with probability 0.2 equal to 0.0; the volatility, σ, is distributed uniformly between 0.1 and 0.6; the time to maturity, T, 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.

To test the accuracy of various trinomial trees, we use three different error measures. First, we use the root-mean-squared (rms) absolute error. Next, we use the Broadie–Detemple relative error measure. This measures the relative errors but excludes cases where the true price is less than 0.5 in order to avoid distortions due to errors on small values. However, this error measure does not assess the accuracy of the trees in pricing deep out-of-the-money options as options with prices less than 0.5 are excluded. It is also soft on deep-in-the-money options since most of the value will be the intrinsic value which is model independent.

Finally, we adopt the Joshi modified-relative error measure introduced by Joshi (2007c)

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

The main advantage of Joshi modified-relative error measure is that it incorporates both deep-out-of-the-money and deep-in-the-money options without distorting the test results. Hence, we expect that it might give different conclusions to the absolute error measure and the Broadie-Detemple relative error measure (Joshi, 2007c).

For each of the eight trinomial trees discussed, we run the trinomial tree to price 3000 randomly generated put options using all of the keys in Table 3.1 with the following number of steps

$$100, 200, 400, 800, 1600.$$

In addition, we also price the 3000 put options using Tian third order moments matching binomial tree[3] (Tian3) with the same keys and number of steps. 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 Joshi modified-relative rms error of 1E-3 and a relative rms error (Broadie–Detemple) of 0.25E-4. The difference in target values expressing the fact that the Broadie–Detemple error measure is more lenient.

5. Numerical results

Tables 5.1, 5.2 and 5.3 show the number of option evaluations per second required to obtain a given level of accuracy while Tables 5.4, 5.5, and 5.6 show the accuracy and speed of the Tian3 binomial and three interesting trinomial trees (which includes the most accurate of trinomial trees) for a given error definition and number of steps. Figures 1, 2 and 3 graph the log of errors against the log of time for selected cases with key 13.

Depending upon our methodology for error measurement, Tables 5.1, 5.2 and 5.3 show that the most effective trinomial trees without acceleration technique (key 0) are KR (absolute error), Boyle1.3 (Broadie-Detemple error) and Boyle1.1 (modified relative error). While the most effective trinomial trees with acceleration techniques are Tian4 with key 13 (absolute error), EqualProb with key 13 (Broadie-Detemple error) and Boyle1.2 with key 9 (Joshi's modified relative error). However, the Tian3 binomial tree with key 13 outperforms all the trinomial trees in all cases.

[3]The specification of the model parameters of this tree are given in Joshi (2007c)

key	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Boyle1.2	7	52	83	301	13	92	65	212	7	54	82	307	103	371	63	213
LogSpace	16	90	15	77	3	23	11	69	28	149	16	83	35	182	12	72
EqualProb	3	29	64	255	11	81	51	181	3	30	68	270	126	425	52	189
Adjusted	16	91	15	80	3	24	12	71	28	150	16	84	29	162	12	72
Tian4	20	109	22	110	3	26	17	90	44	227	25	125	188	553	19	98
Boyle1.1	5	40	97	333	20	124	76	235	5	42	103	357	173	525	80	248
Boyle1.3	10	67	61	245	10	70	47	173	11	74	60	248	77	307	46	174
KR	47	234	71	271	12	87	56	192	75	361	67	268	108	382	52	187
Tian3	3	39	10	57	3	31	8	41	3	41	9	57	1039	1435	8	41

TABLE 5.1. 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
Boyle1.2	84	385	822	1291	62	260	614	864	97	429	662	1201	636	1210	482	796
LogSpace	104	441	177	493	55	241	138	343	286	885	161	479	389	883	122	328
EqualProb	69	334	216	559	55	242	167	386	76	363	178	512	1051	1665	135	350
Adjusted	98	424	126	394	14	98	99	278	207	714	117	388	147	470	91	270
Tian4	108	452	209	547	13	93	163	380	360	1029	202	556	825	1428	153	380
Boyle1.1	62	304	741	1211	80	309	566	822	67	332	732	1281	832	1436	512	828
Boyle1.3	124	495	709	1178	42	203	537	795	164	610	570	1090	543	1094	413	720
KR	55	273	109	359	14	100	86	255	86	393	98	344	148	472	76	242
Tian3	49	206	97	266	16	85	74	190	63	242	93	262	3166	3172	71	187

TABLE 5.2. Number of option evaluations a second obtainable with a relative error of 0.25E-4 with 0.5 cut-off.

key	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Boyle1.2	570	1327	116	374	106	370	90	261	583	1410	103	357	106	378	79	247
LogSpace	158	581	63	252	33	175	50	179	166	614	93	333	36	188	72	232
EqualProb	395	1051	143	429	132	426	112	301	368	1043	132	420	135	444	101	290
Adjusted	103	439	54	226	26	150	42	162	119	490	67	267	29	160	52	189
Tian4	255	794	104	348	102	362	82	247	248	807	139	437	174	525	107	301
Boyle1.1	600	1371	123	389	198	556	96	272	567	1385	113	379	212	598	88	264
Boyle1.3	492	1209	119	381	85	320	92	265	494	1265	107	366	87	333	82	253
KR	420	1093	115	372	105	368	90	262	472	1228	107	365	106	380	82	253
Tian3	415	884	196	427	202	455	151	306	459	939	317	606	1016	1413	238	425

TABLE 5.3. Number of option evaluations a second obtainable with a Joshi modified relative error of 1E-3 using 0.5 additional weighting.

Similarly, Tables 5.4, 5.5, and 5.6 show that the most effective trinomial trees using 1600 steps are Boyle1.2 tree with key 13 for absolute error measure and EqualProb tree with key 13 for both the Broadie-Detemple relative error measure and Joshi modified relative error measure. However, the Tian3 binomial tree again outperforms any of the trinomial trees in both accuracy and speed of pricing regardless of the error definition; in particular, when using 1600 steps in pricing, the Tian3 binomial tree achieves double the speed and half the error.

Regardless of the choice of the tree parameters, if only one acceleration technique is to be used, it is obvious that truncation and control variate perform significantly better than smoothing and Richardson extrapolation individually (comparing keys 1, 2, 4 and 8). However, the combination of smoothing and Richardson performs significantly better
.

Overall, the best combination of acceleration techniques is key 13 which include smoothing, Richardson extrapolation and truncation. The control methodology is useful when the error is large, but when the price is accurate without it, adding it in merely slows the pricing. In particular, the key of 15 generally produces significantly slower results (for a given level of accuracy) than the key of 13.

Truncation always increases the speed of computation. Looking at Tables 5.4, 5.5, and 5.6, we could see that truncation speeds up the pricing by four and seven times for the Tian3 binomial tree when using steps of 800 and 1600 respectively and speeds up the pricing by seven and eight times for the trinomial trees whilst having a negligible effect on accuracy.

In order to compute the orders of convergence, we regress the log RMS error against log time taken by fitting the best straight line through the cases with 800, 1600 and 3200 steps (the last three points on the right side for each of the trees in Figures 1, 2 and 3). Since the graphs exhibit significant non-linearity, we drop cases with less than 800 steps from the regression. Table 5.7 shows the orders of convergence for selected cases. We display results for absolute errors, relative errors with modification, and the Broadie–Detemple relative errors.

Although our previous tests suggests that the Tian3 binomial tree with key 13 is the best tree, Table 5.7 seems to suggest otherwise as the selected trinomial trees appear to have a slightly higher orders of convergence. This may lead us to believe that if the number of time steps is large enough, eventually the trinomial trees will outperform the Tian3 binomial tree. However, for “reasonable” numbers of time steps they do worse, see Figures 1, 2 and 3. A naive extrapolation for

		1600	1600	800	800	400	400	200	200	100	100
name	key	error	speed	error	speed	error	speed	error	speed	error	speed
Tian3	13	3.22E-05	47	7.25E-05	124	1.79E-04	301	4.37E-04	731	1.06E-03	1510
Tian3	12	3.22E-05	7	7.25E-05	31	1.79E-04	115	4.37E-04	386	1.06E-03	1119
Boyle1.2	13	7.51E-05	26	1.95E-04	96	6.35E-04	240	1.52E-03	556	2.52E-03	1176
Boyle1.2	12	7.51E-05	3	1.95E-04	13	6.35E-04	53	1.52E-03	189	2.52E-03	608
Tian4	12	7.97E-05	3	1.97E-04	13	4.37E-04	53	1.00E-03	189	2.26E-03	608
Tian4	13	7.97E-05	26	1.97E-04	96	4.37E-04	240	1.00E-03	556	2.26E-03	1176
EqualProb	13	8.44E-05	26	1.92E-04	96	5.90E-04	240	1.28E-03	556	2.13E-03	1176
EqualProb	12	8.44E-05	3	1.92E-04	13	5.90E-04	53	1.28E-03	189	2.13E-03	608

TABLE 5.4. rms error in absolute terms and number of option evaluations per second for 8 good cases using 3,000 evaluations.

name	key	1600 error	1600 speed	800 error	800 speed	400 error	400 speed	200 error	200 speed	100 error	100 speed
Tian3	13	3.03E-06	47	6.84E-06	124	1.66E-05	301	3.80E-05	731	8.61E-05	1510
Tian3	12	3.03E-06	7	6.84E-06	31	1.66E-05	115	3.80E-05	386	8.61E-05	1119
EqualProb	13	5.78E-06	26	1.29E-05	96	3.48E-05	240	7.45E-05	556	1.43E-04	1176
EqualProb	12	5.78E-06	3	1.29E-05	13	3.48E-05	53	7.45E-05	189	1.43E-04	608
Tian4	12	6.31E-06	3	1.62E-05	13	3.67E-05	53	8.65E-05	189	1.89E-04	608
Tian4	13	6.31E-06	26	1.62E-05	96	3.67E-05	240	8.65E-05	556	1.89E-04	1176
Boyle1.2	13	6.50E-06	26	1.54E-05	96	4.41E-05	240	1.44E-04	556	2.39E-04	1176
Boyle1.2	12	6.50E-06	3	1.54E-05	13	4.41E-05	53	1.44E-04	189	2.39E-04	608

TABLE 5.5. rms error in Broadie–Detemple relative terms with cut-off of 0.5 and number of option evaluations per second for 8 good cases using 3,000 evaluations.

		1600	1600	800	800	400	400	200	200	100	100
name	key	error	speed	error	speed	error	speed	error	speed	error	speed
Tian3	13	3.35E-05	47	6.70E-05	124	1.70E-04	301	4.33E-04	731	1.09E-03	1510
Tian3	12	3.35E-05	7	6.70E-05	31	1.70E-04	115	4.33E-04	386	1.09E-03	1119
EqualProb	13	6.23E-05	26	1.28E-04	96	5.87E-04	240	1.21E-03	556	1.60E-03	1176
EqualProb	12	6.23E-05	3	1.28E-04	13	5.87E-04	53	1.21E-03	189	1.60E-03	608
Boyle1.2	13	7.31E-05	26	1.58E-04	96	6.47E-04	240	1.44E-03	556	1.73E-03	1176
Boyle1.2	12	7.31E-05	3	1.58E-04	13	6.47E-04	53	1.44E-03	189	1.73E-03	608
Tian4	13	1.15E-04	26	2.33E-04	96	4.98E-04	240	1.05E-03	556	2.22E-03	1176
Tian4	12	1.15E-04	3	2.33E-04	13	4.98E-04	53	1.05E-03	189	2.22E-03	608

TABLE 5.6. rms error in Joshi modified relative terms with additional weight of 0.5 and number of option evaluations per second for 8 good cases using 3,000 evaluations.

	absolute		B&D		Joshi	
	slope	constant	slope	constant	slope	constant
Tian3 key 13	-0.796	-7.092	-0.822	-9.364	-0.734	-7.340
Tian4 key 13	-0.831	-5.529	-0.868	-7.895	-0.747	-5.627
EqualProb key 13	-0.863	-5.402	-0.886	-8.002	-0.784	-6.088
Boyle1.2 key 13	-0.876	-5.377	-0.899	-7.796	-0.709	-6.192

TABLE 5.7. Order of convergence as expressed as a power of time for a selected few interesting cases.

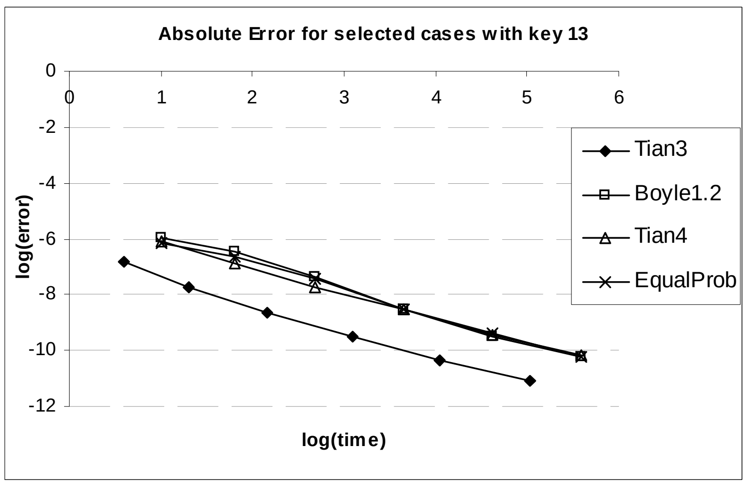

FIGURE 1. Graphs of log root-mean-square absolute error against log time for selected cases with smoothing, truncation and extrapolation.

the case of rms absolute error suggests that it would take 225 years for the trinomial trees to win.

There are a few possible reasons as to why Tian's binomial tree with key 13 outperform all the trinomial trees. One possibility is that the form of the oscillations is such that the errors with n and $2n$ points have similar phase and so are largely removed by extrapolation; whereas the equivalent term in the asymptotic expansions of the other trees does not. However, this is beyond the scope of our paper.

6. Conclusion

In general, when pricing American puts with trees, the acceleration techniques of smoothing, Richardson extrapolation and truncation should be used to increase accuracy and speed, regardless of the choice of parameters or error definition. Amongst the trinomial trees,

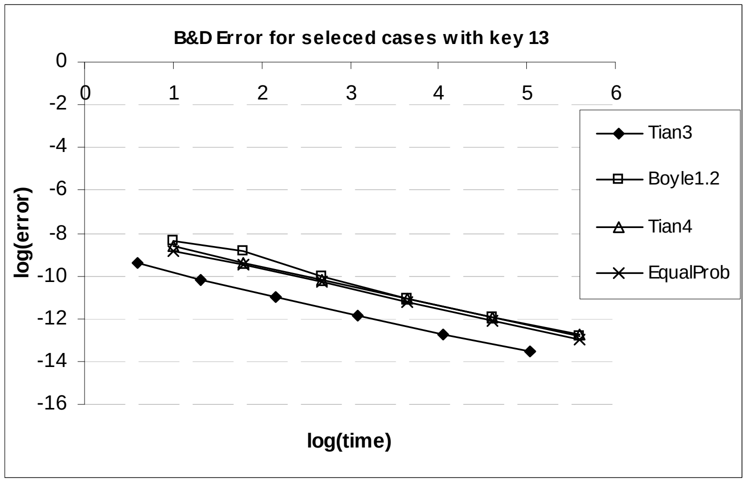

FIGURE 2. Graphs of log root-mean-square Broadie–Detemple relative error against log time for selected cases with smoothing, truncation and extrapolation.

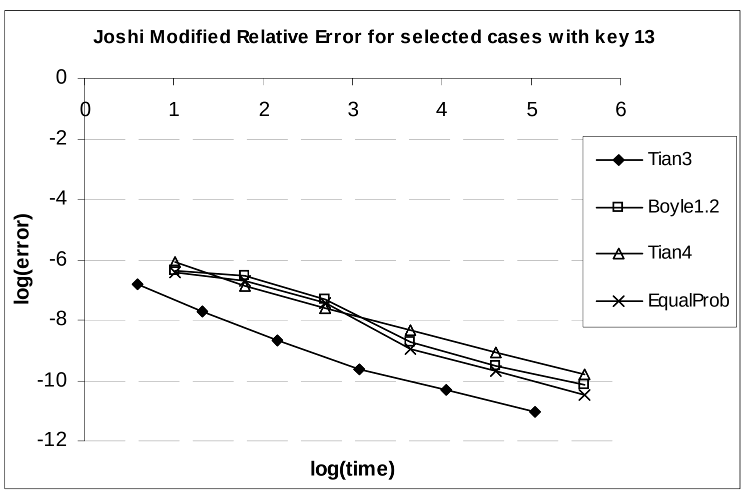

FIGURE 3. Graphs of log root-mean-square modified relative error against log time for selected cases with smoothing, truncation and extrapolation.

Boyle1.2, EqualProb, Tian4 with key 13 all perform well. However, none are as good as the Tian third moment matching binomial tree with the key of 13. We therefore believe that the use of trinomial trees does not offer any improvements to the pricing of an American put option.

References

[1] A.D. Andicropoulos, M. Widdicks, P.W. Duck & D.P. Newton. (2004). Curtailing the range for lattice and grid methods. Journal of Derivatives, 11, 55–61.

[2] P. Boyle. (1986). Option valuation using a three-jump process. International options journal, 3, 7 – 12.

[3] M. Broadie & J. Detemple. (1996). American option valuation: new bounds, approximations, and a comparison of existing methods. The Review of Financial Studies, 9, 1211–1250.

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

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

[6] R. Jarrow & Stuart Turnbull. (2000). Derivative Securities (2nd ed.) Cincinnati: South-Western College Publishing.

[7] M. Joshi. (2003). The Concepts and Practice of Mathematical Finance. Cambridge: Cambridge University Press.

[8] M. Joshi. (2007a). Achieving smooth convergence for the prices of European options in binomial trees, to appear in Quantitative Finance.

[9] M. Joshi (2007b). Achieving higher order convergence for the prices of European options In binomial trees, to appear in Mathematical Finance.

[10] M. Joshi. (2007c). The convergence of binomial trees for pricing the American put. Working paper. Retrieved from:
http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1030143.

[11] B. Kamrad & P. Ritchken. (1991). Multinomial approximating models for options with k state variables. Management Science, 37, 1640 – 1652.

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

[13] M. Staunton. (2005). Efficient estimates for valuing American options. The Best of Wilmott 2. US: Wiley.

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

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

E-mail address: `j.chan23@pgrad.unimelb.edu.au`

E-mail address: `mark@markjoshi.com`

E-mail address: `c.yang7@pgrad.unimelb.edu.au`

E-mail address: `r.tang4@pgrad.unimelb.edu.au`