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, [11], has 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 will also compare the pricing results to the Tian3 binomial tree.

We use the same four acceleration techniques implemented by Joshi, [11]: *control variate* due to [5], *truncation* due to [1], *smoothing* and *Richardson extrapolation* due to [3]. These four techniques can be implemented independently or collectively and therefore yield 16 combinations of acceleration techniques. In addition, one can match the smoothing time when one uses Richardson extrapolation and smoothing together. This yields an extra four combinations. This means that there are a total of 20 different combinations of acceleration techniques

Date: September 1, 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.

one can use when pricing a derivative using trees. For detailed discussion and merits of these acceleration techniques, we refer reader to [11].

As for 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 160 different ways to price an American put option using trinomial trees. We use three different error measures to determine the most accurate trinomial tree, and using a Leisen-Reimer tree with a large number of steps and Richardson extrapolation 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 in [11] which is the Tian third order moment-matching tree with smoothing, Richardson extrapolation and truncation: the binomial tree turns out to be substantially faster.

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 present numerical results in Section 4 and conclude in Section 5.

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 could be found in [11]. 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.

The first three trinomial trees that we will introduce involve moment-matching in the spot space. The Tian equal probability tree, [15], sets the up and down move probabilities to be equal to $1/3$ and matches the first two moments exactly:

$$m = \frac{M(3-V)}{2}, \tag{2.5}$$
$$K = \frac{M(V+3)}{4}, \tag{2.6}$$
$$u = K + \sqrt{K^2 - m^2}, \tag{2.7}$$
$$d = K - \sqrt{K^2 - m^2}. \tag{2.8}$$

We shall call this the "EqualProb" tree.

The Tian fourth order moment matching (Tian4) tree, [15], drops the equal probability constraint of the EqualProb tree and matches the first four moments exactly:

$$m = MV^2, \tag{2.9}$$
$$K = \frac{M}{2}(V^4 + V^3), \tag{2.10}$$
$$u = K + \sqrt{K^2 - m^2}, \tag{2.11}$$
$$d = K - \sqrt{K^2 - m^2}, \tag{2.12}$$
$$p_u = \frac{md - M(m+d) + M^2V}{(u-d)(u-m)}, \tag{2.13}$$
$$p_d = \frac{um - M(u+m) + M^2V}{(u-d)(m-d)}. \tag{2.14}$$

Joshi, [9], introduced an adjusted binomial tree where the tree is centered on the strike in log space. Similarly, here we introduce an *adjusted* trinomial tree 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:

$$m = e^{(r - \frac{1}{2}\sigma^2)\Delta t}, \tag{2.16}$$
$$u = me^{\sqrt{3\Delta t}\sigma}, \tag{2.17}$$
$$d = me^{-\sqrt{3\Delta t}\sigma}, \tag{2.18}$$
$$p_u = \frac{1}{6}, \tag{2.19}$$
$$p_d = \frac{1}{6}. \tag{2.20}$$

while Boyle, [2], 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), [12], tree matches the first moment and variance in the log space:

$$m = 1, \tag{2.26}$$
$$u = \exp(v), \tag{2.27}$$
$$d = 1/u, \tag{2.28}$$
$$v = \lambda\sigma\sqrt{\Delta t}, \tag{2.29}$$
$$\mu = r - \frac{\sigma^2}{2}, \tag{2.30}$$
$$\lambda = 1.22474, \tag{2.31}$$
$$p_u = \frac{1}{2\lambda^2} + \frac{\mu\sqrt{\Delta t}}{2\lambda\sigma}, \tag{2.32}$$
$$p_d = \frac{1}{2\lambda^2} - \frac{\mu\sqrt{\Delta t}}{2\lambda\sigma}. \tag{2.33}$$

3. The implementation choices

The implementation choices in this paper are the same as those in [11]. We shall briefly review all choices in this section and for a more detailed description of all choices, we refer the reader to [11].

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	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.

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 both American and the corresponding European put simultaneously. If P_A and P_E are the tree price 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.** 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

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

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.

As in [11], we also consider the extra *matching* choice which is relevant when doing both smoothing and extrapolation. In order to match the smoothing time for the tree with n steps, We smooth the tree with $2n$ steps at $2(n-1)$ steps rather than at $2n-1$ steps

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

4. Numerical results

Our numerical tests follow those in [11] which were inspired by Broadie and Dutemple in [3]. We pick the option parameters from a random distribution and assess the pricing error by using Leisen-Reimer binomial tree with 14001 steps and Richardson extrapolation as the true value.

The option parameters were selected based on the following distribution: The current spot price, $S(0)$, has a uniform distribution between 70 and 130. 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. Volatility, σ, is distributed uniformly between 0.1 and 0.6. 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 doesn't 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 given in [11]

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

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 [11].

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 [1] (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.

Tables 4.1, 4.2 and 4.3 only show the number of option evaluations per second required to obtain a given level of accuracy. We observe that smoothing combined with Richardson extrapolation and truncation are very effective methods to improve speed regardless of the choice of tree parameters or other combinations of acceleration techniques used. Overall, the fastest results are obtained, when these methods are used together (that is the key of 13). 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. In particular, the key of 15 generally produces significantly slower results (for a given level of accuracy) than the key of 13. Also, we observed that the matching acceleration technique has negligible effect on pricing regardless of the choice of parameters, acceleration combinations, or error definition used.

Depending upon our error methodology, the most effective trinomial trees are Tian4 with key 13 (absolute), EqualProb with key 13 (Broadie–Detemple) and Boyle1.2 with key 9 (modified relative). However, when we compare these trees against the best binomial tree in [11], the Tian3 binomial tree with key 13, the Tian3 outperforms all the trinomial trees in all cases.

Without acceleration techniques, Boyle1.1 and KR trinomial always outperforms the Tian3 binomial tree under all 3 error measures while the reverse is true when the Tian3 binomial tree is used with suitable

[1]The specification of the model parameters of this tree is given in [11]

key	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Boyle1.2	7	52	83	301	13	92	65	212	7	54	82	307	103	371	63	213	102	370	63	212
LogSpace	16	90	15	77	3	23	11	69	28	149	16	83	35	182	12	72	34	181	12	71
EqualProb	3	29	64	255	11	81	51	181	3	30	68	270	126	425	52	189	124	422	52	188
Adjusted	16	91	15	80	3	24	12	71	28	150	16	84	29	162	12	72	29	162	12	72
Tian4	20	109	22	110	3	26	17	90	44	227	25	125	188	553	19	98	186	551	19	97
Boyle1.1	5	40	97	333	20	124	76	235	5	42	103	357	173	525	80	248	169	516	79	247
Boyle1.3	10	67	61	245	10	70	47	173	11	74	60	248	77	307	46	174	76	305	46	173
KR	47	234	71	271	12	87	56	192	75	361	67	268	108	382	52	187	106	381	52	186
Tian3	3	39	10	57	3	31	8	41	3	41	9	57	1039	1435	8	41	1014	1432	7	42

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
Boyle1.2	84	385	822	1291	62	260	614	864	97	429	662	1201	636	1210	482	796	626	1199	479	790
LogSpace	104	441	177	493	55	241	138	343	286	885	161	479	389	883	122	328	384	876	122	327
EqualProb	69	334	216	559	55	242	167	386	76	363	178	512	1051	1665	135	350	1036	1662	135	349
Adjusted	98	424	126	394	14	98	99	278	207	714	117	388	147	470	91	270	144	466	90	268
Tian4	108	452	209	547	13	93	163	380	360	1029	202	556	825	1428	153	380	815	1423	153	378
Boyle1.1	62	304	741	1211	80	309	566	822	67	332	732	1281	832	1436	512	828	811	1418	518	831
Boyle1.3	124	495	709	1178	42	203	537	795	164	610	570	1090	543	1094	413	720	522	1066	412	718
KR	55	273	109	359	14	100	86	255	86	393	98	344	148	472	76	242	144	466	76	239
Tian3	49	206	97	266	16	85	74	190	63	242	93	262	3166	3172	71	187	3145	3144	69	187

TABLE 4.2. 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	16	17	18	19
Boyle1.2	570	1327	116	374	106	370	90	261	583	1410	103	357	106	378	79	247	105	377	79	246
LogSpace	158	581	63	252	33	175	50	179	166	614	93	333	36	188	72	232	36	187	72	231
EqualProb	395	1051	143	429	132	426	112	301	368	1043	132	420	135	444	101	290	133	442	101	289
Adjusted	103	439	54	226	26	150	42	162	119	490	67	267	29	160	52	189	28	160	52	187
Tian4	255	794	104	348	102	362	82	247	248	807	139	437	174	525	107	301	172	523	107	300
Boyle1.1	600	1371	123	389	198	556	96	272	567	1385	113	379	212	598	88	264	206	588	87	262
Boyle1.3	492	1209	119	381	85	320	92	265	494	1265	107	366	87	333	82	253	86	331	82	252
KR	420	1093	115	372	105	368	90	262	472	1228	107	365	106	380	82	253	105	379	82	252
Tian3	415	884	196	427	202	455	151	306	459	939	317	606	1016	1413	238	425	996	1415	234	426

TABLE 4.3. Number of option evaluations a second obtainable with a Joshi modified relative error of 1E-3 using 0.5 additional weighting.

acceleration techniques. Similar to the findings in [11], this demonstrates that the accuracy without acceleration technique is a poor predictor of accuracy after acceleration.

Tables 4.4, 4.5, and 4.6 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 and Figures 1, 2 and 3 graphs these results. Agreeing with the findings from Tables 4.1, 4.2, and 4.3, we find that the keys of 13 and 17 offer the best results regardless of the choice of tree parameters.

The Tian3 binomial tree clearly outperforms any of the trinomial trees in both accuracy and speed of pricing regardless of the error definition; in particular, when 1600 steps are used in pricing, the Tian3 binomial tree prices options almost twice as fast for a given level of accuracy than the most accurate trinomial tree. Note also 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. When comparing between the trinomial trees only, the most accurate tree depends on the error definition used; in the absolute measure it is the Boyle1.2 tree (with key of 13) that is most accurate whereas under both the Broadie-Detemple relative measure and Joshi modified relative measure it is the EqualProb tree.

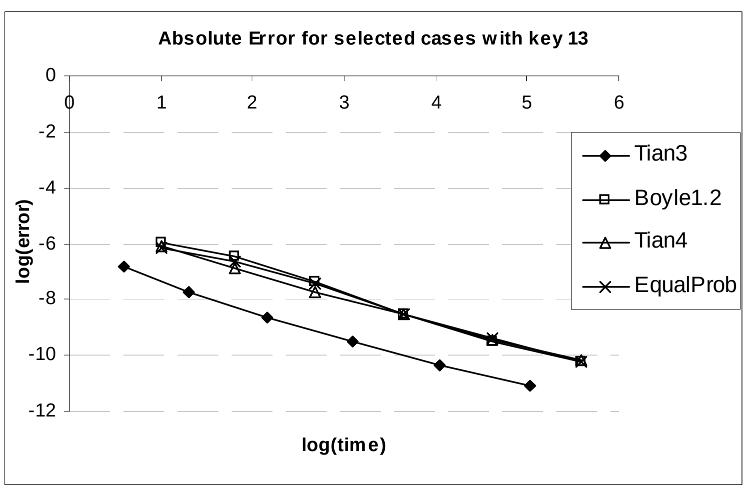

FIGURE 1. Graphs of log root-mean-square absolute error against log time for selected cases with smoothing, truncation and extrapolation.

		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
Tian3	17	3.23E-05	48	7.25E-05	124	1.78E-04	302	4.38E-04	734	1.08E-03	1523
Tian3	16	3.23E-05	6	7.25E-05	30	1.78E-04	111	4.38E-04	378	1.08E-03	1111
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
Boyle1.2	17	7.52E-05	26	1.95E-04	96	6.35E-04	240	1.53E-03	556	2.56E-03	1176
Boyle1.2	16	7.52E-05	3	1.95E-04	13	6.35E-04	53	1.53E-03	188	2.56E-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
Tian4	16	7.97E-05	3	1.97E-04	13	4.39E-04	53	1.01E-03	188	2.28E-03	608
Tian4	17	7.97E-05	26	1.97E-04	96	4.39E-04	240	1.01E-03	556	2.28E-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
EqualProb	17	8.45E-05	26	1.92E-04	96	5.93E-04	240	1.29E-03	556	2.15E-03	1176
EqualProb	16	8.45E-05	3	1.92E-04	13	5.93E-04	53	1.29E-03	188	2.15E-03	608

TABLE 4.4. rms error in absolute terms and number of option evaluations per second for 16 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
Tian3	17	3.11E-06	48	6.95E-06	124	1.65E-05	302	3.86E-05	734	8.83E-05	1523
Tian3	16	3.11E-06	6	6.95E-06	30	1.65E-05	111	3.86E-05	378	8.83E-05	1111
EqualProb	17	5.77E-06	26	1.29E-05	96	3.48E-05	240	7.49E-05	556	1.44E-04	1176
EqualProb	16	5.77E-06	3	1.29E-05	13	3.48E-05	53	7.49E-05	188	1.44E-04	608
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	16	6.29E-06	3	1.63E-05	13	3.71E-05	53	8.68E-05	188	1.90E-04	608
Tian4	17	6.29E-06	26	1.63E-05	96	3.71E-05	240	8.68E-05	556	1.90E-04	1176
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	17	6.48E-06	26	1.54E-05	96	4.40E-05	240	1.45E-04	556	2.43E-04	1176
Boyle1.2	16	6.48E-06	3	1.54E-05	13	4.40E-05	53	1.45E-04	188	2.43E-04	608
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 4.5. rms error in Broadie–Detemple relative terms with cut-off of 0.5 and number of option evaluations per second for 16 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	3.35E-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
Tian3	17	3.36E-05	48	6.70E-05	124	1.70E-04	302	4.35E-04	734	1.10E-03	1523
Tian3	16	3.36E-05	6	6.70E-05	30	1.70E-04	111	4.35E-04	378	1.10E-03	1111
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
EqualProb	17	6.23E-05	26	1.28E-04	96	5.89E-04	240	1.22E-03	556	1.61E-03	1176
EqualProb	16	6.23E-05	3	1.28E-04	13	5.89E-04	53	1.22E-03	188	1.61E-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
Boyle1.2	17	7.33E-05	26	1.58E-04	96	6.49E-04	240	1.45E-03	556	1.75E-03	1176
Boyle1.2	16	7.33E-05	3	1.58E-04	13	6.49E-04	53	1.45E-03	188	1.75E-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
Tian4	17	1.15E-04	26	2.34E-04	96	5.01E-04	240	1.05E-03	556	2.23E-03	1176
Tian4	16	1.15E-04	3	2.34E-04	13	5.01E-04	53	1.05E-03	188	2.23E-03	608

TABLE 4.6. rms error in Joshi modified relative terms with additional weight of 0.5 and number of option evaluations per second for 16 good cases using 3,000 evaluations.

	absolute slope	constant	B&D slope	constant	Joshi slope	constant
Tian3 key 13	-0.796	-7.092	-0.822	-9.364	-0.734	-7.340
Tian3 key 17	-0.801	-7.073	-0.823	-9.344	-0.735	-7.338
Tian4 key 17	-0.824	-5.569	-0.860	-7.936	-0.741	-5.662
Tian4 key 13	-0.831	-5.529	-0.868	-7.895	-0.747	-5.627
EqualProb key 17	-0.853	-5.452	-0.875	-8.060	-0.776	-6.128
EqualProb key 13	-0.863	-5.402	-0.886	-8.002	-0.784	-6.088
Boyle1.2 key 17	-0.867	-5.422	-0.888	-7.850	-0.702	-6.225
Boyle1.2 key 13	-0.876	-5.377	-0.899	-7.796	-0.709	-6.192

Table 4.7. Order of convergence as expressed as a power of time for a selected few interesting cases.

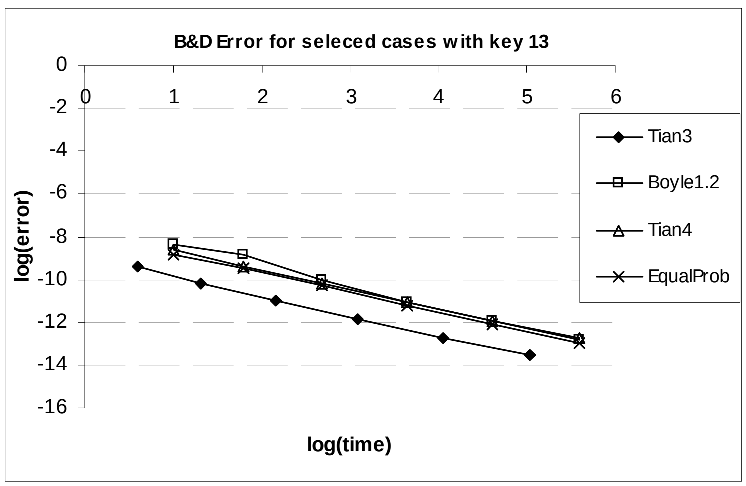

Figure 2. Graphs of log root-mean-square Broadie–Detemple relative error against log time for selected cases with smoothing, truncation and extrapolation.

To compute orders of convergence, we regress the log RMS error against log time taken and fitting the best straight line through the cases with 800, 1600 and 3200 steps; see Table 4.7 for selected cases. We display results for absolute errors, relative errors with modification, and the Broadie–Detemple relative errors.

Although our previous test suggests that the Tian3 binomial tree with key 13 is the best tree, table 4.7 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

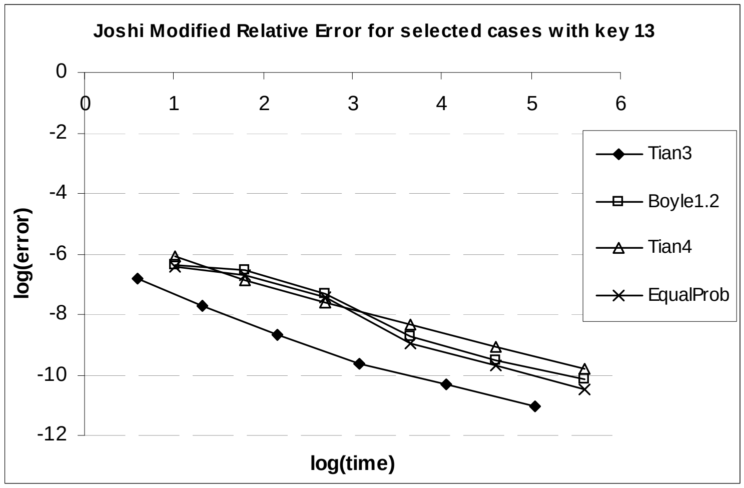

Figure 3. Graphs of log root-mean-square modified relative error against log time for selected cases with smoothing, truncation and extrapolation.

the Tian3 binomial tree. However, for "reasonable" numbers of time steps they do worse, see figures 1, 2 and 3. A naive extrapolation for the case of rms absolute error suggests that it would take 225 years for the trinomial trees to win.

5. 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, 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, Curtailing the range for lattice and grid methods, *Journal of Derivatives,* 55–61 (2004)

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

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

[4] N. Chriss, *Black–Scholes and beyond: Option Pricing Models,* McGraw–Hill, New York (1996)
[5] J. Hull, A. White, The use of the control variate technique in option pricing, *Journal of Financial and Quantitative Analysis,* 23: 237–251 (1988)
[6] R. Jarrow, A. Rudd, *Option pricing,* Homewood, IL: Richard D. Irwin, (1993)
[7] R. Jarrow, Stuart Turnbull, *Derivative Securities,* 2nd ed. Cincinnati: South-Western College Publishing (2000)
[8] M. Joshi, *The Concepts and Practice of Mathematical Finance*, Cambridge University Press (2003)
[9] M. Joshi, Achieving Smooth Convergence for The Prices of European Options In Binomial Trees, preprint (2006)
[10] M. Joshi, Achieving Higher Order Convergence for The Prices of European Options In Binomial Trees, to appear in *Mathematical Finance.*
[11] M. Joshi, The convergence of binomial trees for pricing the american put, preprint (2007)
[12] B. Kamrad, P. Ritchken, Multinomial approximating models for options with k state variables, *Management Science.* Volume 37, Number 12: 1640 – 1652 (1991)
[13] D.P. Leisen, M. Reimer, Binomial models for option valuation-examining and improving convergence, *Applied Mathematical Finance,* 3: 319–346 (1996)
[14] M. Staunton, Efficient estimates for valuing American options, *the Best of Wilmott 2,* John Wiley and Sons Ltd (2005)
[15] Y. Tian, A modified lattice approach to option pricing, *Journal of Futures Markets,* 13(5): 563–577 (1993)

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